

**PREDIKSI RATING PADA *REVIEW* PRODUK KECANTIKAN  
MENGUNAKAN METODE *CONTEXTUAL VALENCE SHIFTERS*  
DAN REGRESI LINEAR**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Nanda Firizki Ananta

NIM: 145150200111068



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019**

# PENGESAHAN

PREDIKSI RATING PADA REVIEW PRODUK KECANTIKAN  
MENGUNAKAN METODE *CONTEXTUAL VALENCE SHIFTERS*  
DAN REGRESI LINEAR

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Nanda Firzki Ananta  
NIM: 145150200111068

Skripsi ini telah diuji dan dinyatakan lulus pada  
24 April 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

M. Ali Fauzi, S.Kom., M.Kom  
NIK. 201502 890101 1 001

Dosen Pembimbing II

Sutrisno, Ir., M.T  
NIP. 19570325 198701 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kumiawan, S.T., M.T., Ph.D  
NIP. 19710518 200312 1 001

## PERNYATAAN ORISINILITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 01 April 2019



Nanda Firizki Ananta

NIM: 145150200111068

## KATA PENGANTAR

Syukur alhamdulillah dihaturkan kepada Allah Subhanahu Wa Ta'ala atas segala berkah, rahmat, dan hidayah yang senantiasa diberikan sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "Prediksi *Rating* Pada *Review* Produk Kecantikan Menggunakan Metode *Contextual Valence Shifters* Dan Regresi Linear". Shalawat dan salam semoga dicurahkan kepada junjungan dan suri tauladan kita, Nabi Muhammad Shalallahu 'Alaihi Wasallam yang telah memberikan tuntunan dan petunjuk kepada umat manusia.

Penulisan skripsi ini merupakan salah satu syarat bagi penulis dalam memperoleh gelar sarjana komputer pada fakultas ilmu komputer universitas brawijaya. Dalam penulisan skripsi ini penulis banyak belajar berbagai hal yang tidak diajarkan pada materi perkuliahan. Belajar menghadapi permasalahan dan mengatasinya serta rasa semangat yang penulis miliki sehingga dapat menyelesaikan skripsi ini, pada kesempatan kali ini penulis ingin menyampaikan rasa syukur dan terima kasih kepada:

1. Bapak M. Ali Fauzi, S.Kom, M.Kom selaku Dosen Pembimbing I yang telah sabar dalam memberikan bimbingan, ilmu, arahan, bantuan, dan saran terhadap pelaksanaan dan penulisan skripsi penulis.
2. Bapak Sutrisno, Ir., M.T selaku Dosen Pembimbing II yang telah memberikan ilmu, bantuan, arahan, dan saran terhadap pelaksanaan dan penulisan skripsi penulis.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T., Ph.D, selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan , S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
5. Keluarga penulis Bapak Marwoto dan Ibu Sumarni selaku kedua orang tua penulis, Addin Aulia Nanda selaku adik penulis serta seluruh keluarga besar atas segala do'a, nasihat, dukungan baik moril maupun materil yang begitu besar terhadap kelancaran dalam menyelesaikan skripsi ini.
6. Seluruh dosen dan karyawan Fakultas Ilmu Komputer Universitas Brawijaya yang telah mendidik dan memberikan ilmu kepada penulis selama menempuh masa studi di Fakultas Ilmu Komputer.
7. Teman-teman seperjuangan yang telah banyak membantu penulis baik secara langsung maupun tidak langsung yaitu Wahyu, Shobah, Adiana, Alfin, Agung, Udin, Irwan dan lainnya.
8. Seluruh pihak yang membantu kelancaran pengerjaan skripsi yang tidak dapat penulis sebutkan satu per satu.

Semoga Allah subhanahu wa Ta'ala senantiasa membalas segala kebaikan dan jasa yang telah diberikan kepada penulis. Penulis menyadari dalam penulisan skripsi ini terdapat banyak kekurangan baik dalam penulisan serta isinya, maka dari itu penulis menerima kritik dan saran untuk penyempurnaan skripsi ini.

Malang, 01 April 2019

Penulis  
nandafirizkiananta@gmail.com



## ABSTRAK

**Nanda Firizki Ananta, Prediksi *Rating* Pada *Review* Produk Kecantikan Menggunakan Metode *Contextual Valence Shifters* dan Regresi Linear**

**Pembimbing: M. Ali Fauzi, S.Kom., M.Kom dan Sutrisno, Ir., M.T.**

Pada saat ini terdapat berbagai macam produk kecantikan. Dengan adanya berbagai macam produk tersebut pemilihan produk kecantikan yang sesuai kebutuhan perlu dilakukan untuk mendapatkan hasil terbaik. Salah satu cara untuk pemilihan produk kecantikan untuk konsumen adalah dengan melihat *review* beserta rating dari produk yang akan dibeli. Dengan adanya berbagai sumber *review*, tidak jarang *review* tersebut tidak disertai dengan rating sehingga mempersulit konsumen untuk melihat apakah produk yang akan dibelinya merupakan produk yang bagus atau tidak. Oleh karena pada penelitian ini bertujuan untuk mengategorikan *review* tersebut ke dalam bentuk rating sehingga mempermudah konsumen untuk menentukan produk yang dipilih. Sistem yang dibangun pada penelitian ini menggunakan metode *Contextual Valence Shifters* dan Regresi Linear serta penggunaan *n-gram* meliputi *bigram*, *trigram*, dan kalimat *review*. Pada pengujian sistem, hasil tertinggi untuk model pengujian toleransi 0 adalah 21,6% untuk *bigram* dan *trigram* sedangkan model pengujian toleransi 1 akurasi tertinggi adalah 66,5% untuk *bigram* dan untuk sentimen *review* adalah 62,4% untuk *bigram*. Dari hasil pengujian yang dilakukan, penggunaan *n-gram* khususnya *bigram* berdampak positif pada hasil akurasi sistem.

**Kata kunci:** Prediksi rating, *review*, *Contextual Valence Shifters*, Regresi Linear, *n-gram*.

## ABSTRACT

**Nanda Firizki Ananta, *Rating Prediction on Beauty Product Reviews Using Contextual Valence Shifters and Linear Regression Method***

**Pembimbing: M. Ali Fauzi, S.Kom., M.Kom dan Sutrisno, Ir., M.T.**

*At present, there are various kinds of beauty products. With a variety of products, the selection of beauty products in accordance with the needs must be done to get the best results. One way to choose beauty products for consumers is to look at reviews along with ratings of the products to be purchased. But with the existence of various review sources, it is not uncommon for the review not to be accompanied by a rating, making it difficult for the consumers to see whether the product to buy is a good product or not. Therefore, this research aims to categorize the review into a rating so that it is easier for consumers to determine the selected product. The system built in this research uses the Contextual Valence Shifters and Linear Regression methods and the use of n-grams includes taking the word bigram, trigram, and review sentences. In system testing, the highest results for the tolerance 0 testing model are 21.6% for bigram and trigram, the tolerance 1 test model the highest accuracy is 66.5% for bigram and for sentiment review is 62.4% for bigram. From the results of the tests, the use of n-gram especially bigram had a positive impact on the results of system accuracy.*

**Keywords:** *Rating prediction, review, Contextual Valence Shifters, Linear Regression, n-gram.*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINILITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN .....	xiii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	2
1.6 Sistematika pembahasan.....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>4</b>
2.1 Tinjauan Penelitian .....	4
2.2 <i>Text Mining</i> .....	5
2.3 <i>Pre-Processing</i> .....	5
2.3.1 <i>Case Folding</i> .....	5
2.3.2 <i>Tokenizing</i> .....	5
2.4 <i>Lexicon Based Features</i> .....	6
2.5 <i>N-gram</i> .....	6
2.6 <i>Contextual Valence Shifters</i> .....	6
2.7 Regresi Linear.....	7
2.8 Pengujian .....	8
2.8.1 Akurasi .....	8
2.8.2 <i>K-Fold Cross Validation</i> .....	8



BAB 3 METODOLOGI PENELITIAN .....	9
3.1 Tipe Penelitian.....	9
3.2 Strategi dan Rancangan Penelitian .....	9
3.2.1 Strategi/Metode .....	9
3.2.2 Subjek atau Partisipan Penelitian .....	9
3.2.3 Lokasi Penelitian .....	9
3.2.4 Metode/Teknik Pengumpulan Data .....	9
3.2.5 Metode/Teknik Analisis Data.....	10
3.2.6 Peralatan Pendukung.....	10
BAB 4 ANALISIS DAN PERANCANGAN .....	11
4.1 Deskripsi Permasalahan .....	11
4.2 Deskripsi Umum Sistem .....	11
4.3 Proses <i>Training</i> .....	12
4.4 Proses <i>Testing</i> .....	13
4.5 <i>Pre-Processing</i> .....	14
4.5.1 <i>Case Folding</i> .....	15
4.5.2 <i>Tokenizing</i> .....	16
4.6 <i>Lexicon Based Features</i> .....	17
4.7 <i>N-gram dan Contextual Valence Shifters</i> .....	20
4.8 Regresi Linear .....	25
4.9 Perhitungan Manual.....	29
4.10 Perancangan Pengujian .....	40
4.11 Perancangan Antarmuka .....	41
BAB 5 IMPLEMENTASI .....	44
5.1 Batasan Implementasi.....	44
5.2 Implementasi Sistem.....	44
5.3 <i>Pre-processing</i> .....	46
5.3.1 <i>Case Folding</i> .....	46
5.3.2 <i>Tokenizing</i> .....	47
5.4 Pembobotan dengan <i>Lexicon Based features</i> .....	47
5.5 <i>N-gram dan Contextual Valence Shifters</i> .....	49
5.4.1 <i>Bigram</i> .....	49

5.4.2	<i>Trigram</i> .....	50
5.4.3	Kalimat <i>Review</i> .....	53
5.6	Regresi Linear .....	54
5.6.1	Regresi Linear <i>Data Training</i> .....	54
5.6.2	Regresi Linear <i>Data Testing</i> .....	55
5.7	Akurasi.....	56
5.8	Implementasi Antarmuka .....	58
BAB 6 PENGUJIAN DAN ANALISIS.....		61
6.1	Skenario Pengujian.....	61
6.2	Evaluasi Hasil Pengujian .....	62
6.2.1	Analisis Hasil Pengujian dengan <i>Bigram</i> .....	63
6.2.2	Analisis Hasil Pengujian dengan <i>Trigram</i> .....	64
6.2.3	Analisis Hasil Pengujian dengan Kalimat <i>review</i> .....	64
BAB 7 PENUTUP .....		65
7.1	Kesimpulan.....	65
7.2	Saran.....	65
DAFTAR REFERENSI .....		66
LAMPIRAN .....		67



## DAFTAR TABEL

Tabel 2.1 Ilustrasi <i>case folding</i> .....	5
Tabel 2.2 Ilustrasi <i>tokenizing</i> .....	5
Tabel 2.3 Ilustrasi <i>N-gram</i> .....	6
Tabel 2.4 Ilustrasi <i>Contextual Valence Shifters</i> .....	7
Tabel 4.1 <i>Review Produk</i> .....	29
Tabel 4.2 Hasil proses <i>Pre-processing</i> .....	30
Tabel 4.3 Pembobotan perkata.....	35
Tabel 4.4 Pembobotan Bigram.....	31
Tabel 4.5 Pembobotan Trigram .....	33
Tabel 4.6 Proses Regresi Bigram .....	37
Tabel 4.7 Proses Regresi Trigram .....	38
Tabel 4.8 Proses Regresi Kalimat .....	38
Tabel 4.9 Perancangan Pengujian .....	40
Tabel 5.1 Daftar fungsi yang ada pada sistem .....	44
Tabel 6.1 Hasil pengujian Contextual Valence Shifter dan N-gram .....	61

## DAFTAR GAMBAR

Gambar 4.1 Deskripsi Umum Sistem .....	12
Gambar 4.2 Alur Data Training .....	13
Gambar 4.3 Alur Data Testing.....	14
Gambar 4.4 Proses <i>Pre-Processing</i> .....	15
Gambar 4.5 Proses <i>Case Folding</i> .....	16
Gambar 4.6 Proses <i>Tokenizing</i> .....	17
Gambar 4.7 Proses pembobotan <i>Lexicon Sentistrength_id</i> .....	19
Gambar 4.8 Bigram dan <i>Contextual Valence Shifters</i> .....	21
Gambar 4.9 Trigram dan <i>Contextual Valence Shifters</i> .....	24
Gambar 4.10 Kalimat review dan <i>Contextual Valence Shifters</i> .....	25
Gambar 4.11 Regresi Linear Data Training bigram .....	26
Gambar 4.12 Regresi Linear Data Training Trigram.....	27
Gambar 4.13 Regresi Linear Data Training Kalimat .....	28
Gambar 4.14 Regresi Linear Data Testing.....	29
Gambar 4.15 Rancangan User Interface Halaman Input .....	41
Gambar 4.16 Rancangan User Interface Halaman Pembobotan.....	42
Gambar 4.17 Rancangan User Interface Halaman Prediksi .....	43
Gambar 5.1 Antarmuka Sistem Prediksi Rating (Menu Data).....	59
Gambar 5.2 Antarmuka Sistem Prediksi Rating (menu Pembobotan) .....	59
Gambar 5.3 Antarmuka Sistem Prediksi Rating (menu Prediksi).....	60
Gambar 6.1 Grafik Hasil Pengujian Prediksi Rating dengan Metode <i>Contextual Valence Shifters</i> dan Regresi Linear .....	62

## DAFTAR LAMPIRAN

Lampiran 1 Data Training dan Data Testing ..... 67



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Saat ini jual beli *online* sangat sering dilakukan oleh masyarakat, salah satunya adalah jual beli produk kecantikan. Pemilihan produk kecantikan yang sesuai dengan kebutuhan perlu dilakukan untuk mendapatkan hasil terbaik sesuai yang diinginkan. Pada saat ini terdapat beberapa *website* yang menyediakan *review* terhadap produk-produk kecantikan.

*Website* yang menyediakan *review* khususnya produk kecantikan adalah femaledialy.com, *website* ini berisi kurang lebih 10.000 produk kecantikan beserta *review* dan rating. Adanya sistem rating pada produk di *website* ini merupakan suatu cerminan dari kualitas produk di mata pembeli. Dikarenakan banyaknya data dari berbagai sumber-sumber lain yang berupa *review* namun tidak disertai rating akan sangat menyulitkan bagi pembeli yang ingin membeli produk tersebut. Oleh karena itu dibutuhkan suatu sistem yang dapat membantu untuk menentukan rating dari suatu *review* yang ada sehingga tidak menyulitkan pembeli untuk menilai apakah produk yang ingin dibeli kualitasnya baik atau tidak.

Banyak metode yang dapat digunakan untuk prediksi rating, salah satunya adalah Regresi Linear. Regresi Linear adalah metode yang menghasilkan persamaan linear, dari persamaan yang dihasilkan, prediksi dapat dilakukan dengan cara memasukkan variabel prediktor pada persamaan linear tersebut. Proses tersebut menghasilkan suatu nilai berupa nilai prediksi variabel respon (Andini, Witanti, & Renaldi, 2016).

Metode Regresi Linear membutuhkan dua variabel untuk dihubungkan. Pada data yang digunakan, *review* sudah memiliki satu variabel yaitu rating. Oleh karena itu dibutuhkan satu lagi variabel yang akan digunakan untuk memodelkan hubungan antara 2 variabel. Salah satu metode yang dapat digunakan adalah metode *Contextual Valence Shifters* untuk memperoleh suatu bobot dari *review* berdasarkan *lexicon*.

Pada penelitian yang dilakukan oleh Prakash Kumar Singh, peneliti menggunakan pendekatan metode *Contextual Valence Shifters* yang berorientasi kata kerja. Peneliti menghitung skor sentimen dengan *SentiWordNet* dan *WordNet*. Pada penelitian ini mendapatkan *precision* sebesar 91,428%, *recall* 82,051% dan akurasi sebesar 79,166% (Singh, Singh, & Paul, 2015).

Metode *Contextual Valence Shifters* merupakan metode yang berbasis pada *lexicon* atau biasa disebut *lexicon based feature*. Pada penelitian yang dilakukan oleh Hailong Zhang, peneliti menggunakan metode *Naïve bayes*, *Support Vector Machine (SVM)* dan *Lexicon based method* dengan *sentiWordNet* sebagai *lexicon* yang digunakan untuk penelitiannya. Hasil yang diperoleh dari penelitian ini adalah *lexicon based* mendapatkan akurasi sebesar 64,25%, *Naïve bayes* mendapatkan akurasi 68,75% dan *Support Vector Machine* mendapatkan akurasi sebesar 71,00% (Zhang, Gan, & Jiang, 2014).

Mengacu dari permasalahan dan uraian penjelasan dari penelitian yang ada sebelumnya, mendorong peneliti untuk mengusulkan penelitian yang berjudul Prediksi Rating pada *Review* Produk Kecantikan Menggunakan Metode *Contextual Valence Shifters* dan Regresi Linear. Pada penelitian ini diharapkan dapat mempermudah konsumen untuk menentukan produk yang ingin dibeli kualitasnya baik atau tidak, berdasarkan rating.

## 1.2 Rumusan masalah

Terdapat masalah yang ada pada penelitian ini, yaitu :

1. Bagaimana menerapkan metode *Contextual Valence Shifters* dan Regresi Linear untuk prediksi rating produk kecantikan ?
2. Bagaimana pengaruh *n-gram* terhadap hasil akurasi dari prediksi rating produk kecantikan?

## 1.3 Tujuan

Terdapat Tujuan yang ingin di capai pada penelitian ini, yaitu :

1. Menerapkan metode *Contextual Valence Shifters* dan regresi linear untuk prediksi rating produk kecantikan.
2. Mengetahui pengaruh *n-gram* terhadap hasil akurasi dari prediksi rating produk kecantikan.

## 1.4 Manfaat

Adapun manfaat yang diperoleh dari penelitian ini, yaitu:

1. Mempermudah orang-orang yang ingin mengetahui rating dari produk kecantikan yang ingin dibeli jika belum terdapat rating pada *review* tersebut.
2. Untuk memperluas wawasan peneliti dan pembaca terkait dengan ilmu *text mining*, terutama pada metode *contextual valence shifters*, *lexicon based feature* dan regresi linear yang digunakan pada penelitian ini.

## 1.5 Batasan masalah

Adapun batasan-batasan yang terdapat pada penelitian ini untuk menghindari permasalahan yang tidak diinginkan.

1. Data yang digunakan pada penelitian ini diperoleh dari *website femaledialy.com* pada tanggal 9 agustus 2018.
2. Jumlah data yaitu 1000 dokumen, berupa 200 data pada setiap rating dan 100 data berupa data *testing*, dan 900 data merupakan data *training*.
3. *Review* yang digunakan menggunakan bahasa indonesia.
4. Klasifikasi rating menggunakan skala 1-5.

## 1.6 Sistematika pembahasan

Sistematika pembahasan bertujuan sebagai untuk memberi gambaran dan uraian singkat mengenai struktur penulisan secara garis besar pada setiap bab dijelaskan sebagai berikut.

### **BAB I Pendahuluan**

Pada bab ini memuat latar belakang dari permasalahan, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

### **BAB II Tinjauan Pustaka**

Pada bab ini memuat dasar teori dari berbagai sumber pustaka yang mendukung, referensi yang menyangkut penelitian-penelitian sebelumnya, dan menjelaskan perbedaan dari penelitian-penelitian sebelumnya.

### **BAB III Metodologi Penelitian**

Pada bab ini menjelaskan metode yang digunakan untuk penelitian yang dilakukan, berupa studi literature, analisis kebutuhan, pengumpulan data, perancangan simulasi dan pengambilan kesimpulan.

### **BAB IV Analisis dan Perancangan**

Bab ini berisi dengan proses analisis dan perancangan, termasuk di dalamnya terdapat perancangan sistem, uji coba, dan evaluasi.

### **BAB V Implementasi**

Bab ini berisi pembahasan proses implementasi, termasuk didalamnya terdapat batasan, dan algoritme yang digunakan.

### **BAB VI Pengujian dan Analisis**

Bab ini berisi dengan pengujian sistem yang dibuat, dan analisis dari pengujian yang dilaksanakan.

### **BAB VII Penutup**

Bab ini berisi kesimpulan dari hasil penelitian yang dilakukan peneliti dan saran untuk penelitian selanjutnya

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Tinjauan Penelitian

Dalam penulisan penelitian ini, dilakukan kajian terhadap penelitian-penelitian sebelumnya. Penelitian tersebut meliputi prediksi rating, *Lexicon-Based Method*, *Contextual Valence Shifters* dan Regresi Linear.

Pada penelitian sebelumnya yang berkaitan dengan prediksi rating yang dilakukan Young-Chan Lee, peneliti melakukan penelitian untuk memprediksi rating kredit perusahaan, data yang digunakan diambil dari the Korea Information Service. Metode yang digunakan adalah dengan *Support Vector Machines* (SVM) dan dibandingkan dengan *Multiple Discriminant Analysis* (MDA), *Case-Based Reasoning* (CBR) dan *Back-Propagation Neural Networks* (BPN). Hasil yang diperoleh oleh peneliti untuk *training set* metode SVM sebesar 77,62%, BPN sebesar 62,95%, MDA sebesar 58,77% dan CBR tidak ada. Sedangkan untuk *Holdout set* metode SVM sebesar 67,22%, BPN sebesar 59,93%, MDA sebesar 58,72% dan CBR sebesar 63,41% (Lee, 2007).

Penelitian selanjutnya yang berkaitan dengan *lexicon-Based Method* yang dilakukan oleh Phan Trong Ngoc, peneliti melakukan penelitian untuk analisis sentimen untuk *fan page ranking* pada Facebook. Metode yang digunakan adalah menggunakan lexicon AFINN yaitu suatu kamus yang berisi kata-kata bahasa Inggris yang memiliki berat antara -5 sampai 5. Hasil yang diperoleh menunjukkan bahwa metode yang digunakan peneliti lebih mengarah pada opini pengguna (*user*) oleh karena itu hasilnya lebih bermanfaat untuk *social network marketing* (Ngoc & Yoo, 2014).

Penelitian selanjutnya yang berkaitan dengan *Contextual Valence Shifters* yang dilakukan oleh Michal Ptaszynski, peneliti melakukan penelitian untuk analisis pengaruh kontekstual. Metode yang digunakan adalah contextual valence shifter yang di aplikasikan pada *Emotive Elements and Emotive Expressions Analysis System* (ML-Ask). Hasil yang diperoleh pada penelitian ini adalah akurasi sebesar 80,0% (Ptaszynski et al., 2010).

Penelitian selanjutnya yang berkaitan dengan Regresi Linear yang dilakukan oleh M. Syafruddin, peneliti melakukan penelitian untuk memprediksi kebutuhan energi listrik jangka panjang dengan studi kasus Provinsi Lampung. Metode yang digunakan adalah Regresi Linear sederhana dan Regresi Linear berganda. Hasil yang diperoleh pada penelitian ini menunjukkan bahwa terjadi peningkatan terhadap daya listrik tersambung untuk pelanggan rumah tangga dengan rata-rata pertumbuhan 2,46%, sedangkan untuk rata-rata peningkatan konsumsi energi listrik di Provinsi Lampung sebesar 3,83% (Syafruddin, Hakim, & Despa, 2014).

## 2.2 Text Mining

*Text mining* adalah suatu proses menganalisis koleksi material tekstual untuk menangkap konsep utama, tema, hubungan tersembunyi, dan tren tanpa mengharuskan kita untuk mengetahui kata-kata atau istilah yang tepat yang digunakan penulis untuk mengekspresikan konsep-konsep itu (IBM, 2015). *Text mining* sendiri hampir sama dengan *data mining*, namun *text mining* lebih berfokus pada *text*.

## 2.3 Pre-Processing

*Pre-processing* merupakan tahap pertama yang dilakukan pada penelitian ini. *Preprocessing* adalah suatu proses yang dilakukan pada data mentah (*raw data*) guna untuk mempersiapkan prosedur selanjutnya. *Pre-Processing* mengubah data mentah (*raw data*) menjadi format yang akan lebih mudah dan efektif diproses, sehingga data tersebut dapat diolah dengan berbagai metode seperti pada metode yang digunakan pada penelitian ini.

### 2.3.1 Case Folding

Proses *case folding* adalah proses yang mengubah semua data *text* menjadi huruf kecil atau bapat juga sebaliknya. Untuk kasus pada penelitian ini, peneliti mengubah semua huruf besar menjadi huruf kecil, untuk lebih detailnya terdapat pada Tabel 2.1.

Tabel 2.1 Ilustrasi *case folding*

Masukan	Keluaran
Aku tidak suka banget dengan Pensil Alis ini karena harganya mahal dan pas diaplikasiin warnanya tidak keluar	aku tidak suka banget dengan pensil alis ini karena harganya mahal dan pas diaplikasiin warnanya tidak keluar

### 2.3.2 Tokenizing

Proses *tokenizing* adalah suatu proses untuk memecah rangkaian *string* menjadi beberapa bagian seperti kata, kata kunci, frasa, simbol, dan elemen lain yang disebut token. Untuk lebih detailnya terdapat pada Tabel 2.2.

Tabel 2.2 Ilustrasi *tokenizing*

Masukan	Keluaran
Aku tidak suka banget dengan Pensil Alis ini karena harganya mahal dan pas diaplikasiin warnanya tidak keluar	aku tidak suka banget dengan pensil alis ini

	karena harganya mahal dan pas diaplikasiin warnanya tidak keluar
--	--

### 2.4 Lexicon Based Features

*Lexicon Based features* merupakan suatu pembobotan kata berdasarkan kamus atau *lexicon* dengan hasil dari pembobotan tersebut memiliki fitur positif dan negatif (Rofiqoh, et al., 2017). Untuk kamus atau *lexicon* yang digunakan pada penelitian ini adalah kamus yang bernama *sentistrength\_id* (Wahid & Azhari, 2016).

### 2.5 N-gram

Dalam tata bahasa, terkadang suatu bahasa tidak hanya terbentuk dari kata individu, namun terkadang terdiri dari urutan kata individu dan frasa, baik frasa 2, 3 atau bahkan lebih yang biasa dikenal dengan *n-gram*. “n” pada *n-gram* sendiri memiliki arti banyaknya kata yang ada pada susunan kata tersebut, untuk penerapan *n-gram*, khususnya *bigram* pada kalimat “suka banget pensil alis” menjadi “suka banget”, “banget pensil”, dan “pensil alis”. Untuk lebih detailnya terdapat pada Tabel 2.5.

**Tabel 2.3 Ilustrasi N-gram**

<i>Review</i>	<i>Unigram</i>	<i>Bigram</i>	<i>Trigram</i>
suka banget pensil alis harga mahal pas aplikasi warna keluar	suka banget pensil alis harga mahal pas aplikasi warna keluar	Suka banget Banget pensil Pensil alis Alis harga Harga mahal Mahal pas Pas aplikasi Aplikasi warna Warna keluar	Suka banget pensil Banget pensil alis Pensil alis harga Alis harga mahal Harga mahal pas Mahal pas aplikasi Pas aplikasi warna Aplikasi warna keluar

### 2.6 Contextual Valence Shifters

*Contextual Valence Shifters* adalah sebuah metode pembobotan berbasis *lexicon*. Metode ini pertama kali di usulkan oleh Livia Polanyi dan Annie Zaenen pada jurnal yang berjudul *Contextual Valence Shifters*. Metode ini membalik suatu

nilai sentimen dari positif ke negatif jika pada kalimat tersebut terdapat kata negasi contohnya “tidak”, “bukan” dan lainnya. Tidak hanya membalik jika ada kata negasi, metode ini juga menghitung nilai dari *modifier* atau *intensifiers*, jika terdapat kata *modifier* maka tergantung kata yang dikuatkan oleh *modifier* tersebut, jika kata tersebut bernilai positif, lalu *modifier* juga positif maka nilai dari kata tersebut itu akan ditambah satu, namun jika *modifier* negatif maka nilai dari kata tersebut akan dikurangi satu (Polanyi & Zaenen, 2006). Untuk lebih detailnya dapat dilihat pada Tabel 2.6.

**Tabel 2.4 Ilustrasi Contextual Valence Shifters**

Tipe	Kata Awal	Bobot	Hasil	Bobot Hasil
Kata Negasi	Bagus	4	Tidak Bagus	-4
	Cantik	5	Tidak Cantik	-5
Kata Modifier	Jujur	4	Kurang Jujur	3
	Kotor	-3	Kotor Sekali	-4

### 2.7 Regresi Linear

Regresi Linear merupakan salah satu metode yang digunakan untuk prediksi. Tujuan dari Regresi Linear adalah untuk memodelkan hubungan antara dua variabel dengan mencari persamaan Regresi Linear yang akan diamati. Dari persamaan yang telah didapat, dapat dihitung prediksi dengan cara memasukkan nilai variabel prediktor pada persamaan regresi tersebut. Dari proses perhitungan dengan menggunakan persamaan regresi didapatkan hasil berupa nilai prediksi variabel respon. Persamaan umum dari Regresi Linear adalah:

$$Y = a + b \cdot X$$

Dengan:

$$b = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2} \tag{2.1}$$

$$a = \frac{\sum y}{n} - b \frac{\sum x}{n} \tag{2.2}$$

Dimana:

- X merupakan variabel prediktor (*independent variable* atau *explanatory variables*).
- Y merupakan variabel respon (*dependen variable*).
- a adalah konstanta.
- b adalah koefisien regresi.
- n adalah jumlah data



## 2.8 Pengujian

Pengujian adalah mencari tahu seberapa baik sesuatu bekerja. Dalam penelitian ini Pengujian adalah proses untuk membandingkan antara hasil implementasi dengan kriteria standar yang telah ditetapkan untuk melihat keberhasilannya.

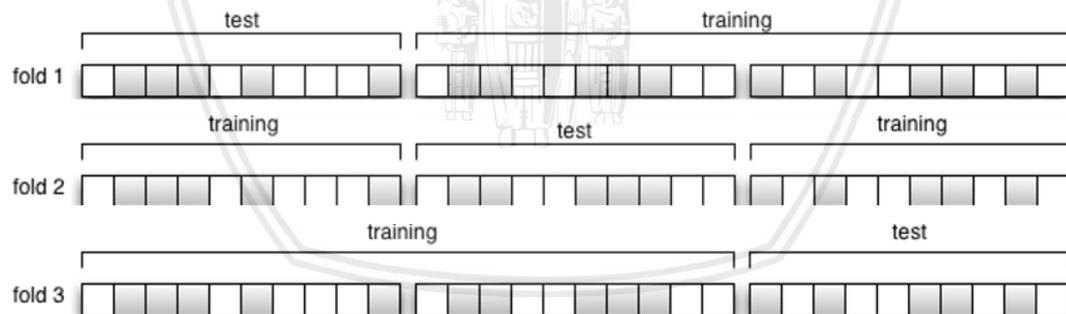
### 2.8.1 Akurasi

Pengujian Akurasi berguna sebagai bahan evaluasi sebgas apa metode yang digunakan peneliti. Pengujian sendiri dilakukan dengan cara membandingkan hasil dari perhitungan data uji benar dan banyaknya seluruh data uji untuk menghitung akurasinya. Pengujian nilai akurasi dapat menggunakan rumus (Mahmudy & Widodo, 2014).

$$\text{Akurasi} = \frac{\text{jumlah data uji benar}}{\text{jumlah seluruh data uji}} \times 100\% \tag{2.3}$$

### 2.8.2 K-Fold Cross Validation

Dalam *data mining* dan *machine learning*. *K-fold cross validation* merupakan suatu pola abstrak untuk mengestimasi akurasi dari klasifikasi dari suatu algoritma. caranya adalah dengan membagi data menjadi data *training* dan data *testing* pembagiannya sebanyak K, untuk lebih memahaminya dapat dilihat pada Gambar 2.1



**Gambar 2.1 K-Fold Cross Validation dengan 30 data dan 3 fold**

Sumber: Dispel Tutorial 0.8 documentation (Martin, 2012)

## BAB 3 METODOLOGI PENELITIAN

Pada bab ini menjelaskan beberapa hal yaitu tipe penelitian, strategi dan rancangan penelitian, serta jadwal penelitian.

### 3.1 Tipe Penelitian

Pada penelitian ini termasuk nonimplementatif dikarenakan lebih mengarah pada investigasi terhadap suatu fenomena atau suatu analisis terhadap hubungan antar fenomena yang sedang dikaji yang tujuannya untuk mendapatkan suatu hasil analisis ilmiah sebagai produknya. Ditinjau dari kegiatan penelitiannya, penelitian ini bertipe analitik (*analytical/explanatory*) yaitu suatu penelitian nonimplementatif yang dilakukan untuk menjelaskan hubungan antar objek dengan suatu fenomena yang sedang diteliti.

### 3.2 Strategi dan Rancangan Penelitian

Strategi dan rancangan penelitian dibagi menjadi enam bagian, yaitu strategi/metode, subjek atau partisipan penelitian, lokasi penelitian, metode/teknik pengumpulan data, metode/teknik analisis data dan Peralatan pendukung.

#### 3.2.1 Strategi/Metode

Strategi/metode yang digunakan pada penelitian ini adalah eksperimen dengan menciptakan suatu hipotesis, metode pengumpulan data, analisis, metode eksperimen dan menghasilkan produk utama. Metode yang digunakan pada eksperimen penelitian ini yaitu metode *Contextual Valence Shifter* dan Regresi Linear.

#### 3.2.2 Subjek atau Partisipan Penelitian

Subjek atau partisipan penelitian yang terlibat adalah *reviewer* dari *website* <https://femaledaily.com>. Alasan pemilihan partisipan karena web tersebut merupakan web yang menggunakan bahasa Indonesia sehingga tepat dijadikan partisipan.

#### 3.2.3 Lokasi Penelitian

Lokasi penelitian yang dipilih adalah Laboratorium Komputasi Cerdas milik Fakultas Ilmu Komputer Brawijaya Malang. Alasan pemilihan lokasi tersebut karena peralatan yang lengkap terkait penelitian Prediksi Rating pada *Review Produk Kecantikan Menggunakan Metode Contextual Valence Shifters* dan Regresi Linear.

#### 3.2.4 Metode/Teknik Pengumpulan Data

Pengumpulan data merupakan metode untuk mendapatkan suatu data untuk mengembangkan perangkat lunak. Data tersebut berupa data-data *review* singkat suatu produk kecantikan yang didapat dari *website* <https://femaledaily.com> yang sudah dikategorikan berdasarkan *rating* 1, 2, 3, 4 dan 5 yang ada pada web tersebut.

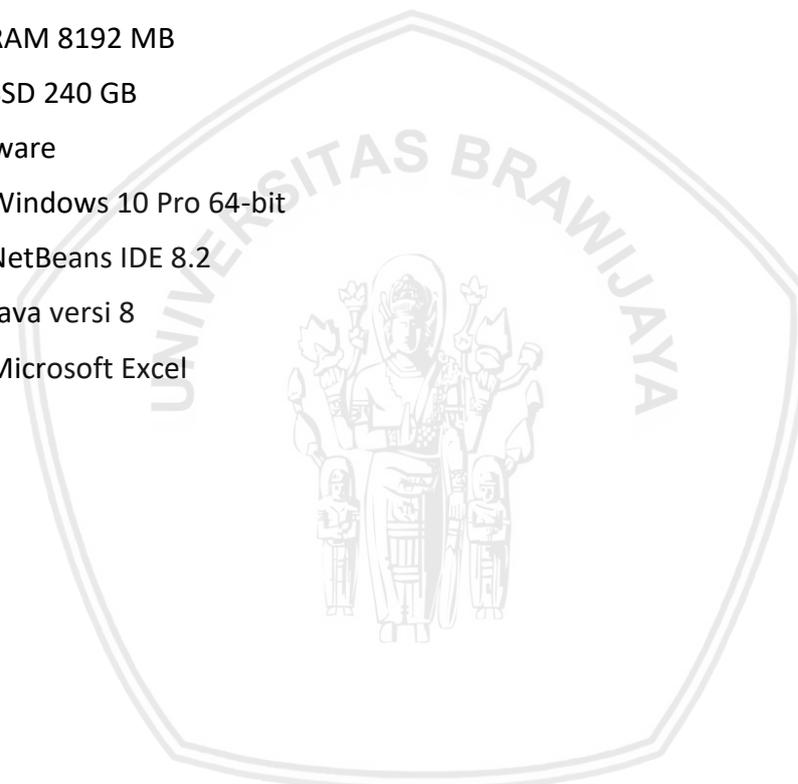
### 3.2.5 Metode/Teknik Analisis Data

Analisis data dengan mengkategorikan *review* produk ke dalam *rating* 1, 2, 3, 4 dan 5. Lalu data tersebut diproses dengan metode *Contextual Valence Shifters* dan Regresi Linear untuk implementasi sistem.

### 3.2.6 Peralatan Pendukung

Peralatan pendukung yang dibutuhkan pada penelitian ini dapat mempengaruhi performa sistem. Peralatan pendukung pada sistem dijelaskan sebagai berikut:

1. Hardware
  - a. Notebook Asus dengan tipe X450C dengan Processor Intel(R) core(TM) i3-3217U CPU @ 1.80GHz (4CPUs)
  - b. RAM 8192 MB
  - c. SSD 240 GB
2. Software
  - a. Windows 10 Pro 64-bit
  - b. NetBeans IDE 8.2
  - c. Java versi 8
  - d. Microsoft Excel



## BAB 4 ANALISIS DAN PERANCANGAN

Analisis dan perancangan akan membahas mengenai deskripsi permasalahan & umum sistem, perhitungan manual, perancangan algoritme untuk menyelesaikan masalah, perancangan antarmuka, perancangan pengujian, dan penarikan kesimpulan.

### 4.1 Deskripsi Permasalahan

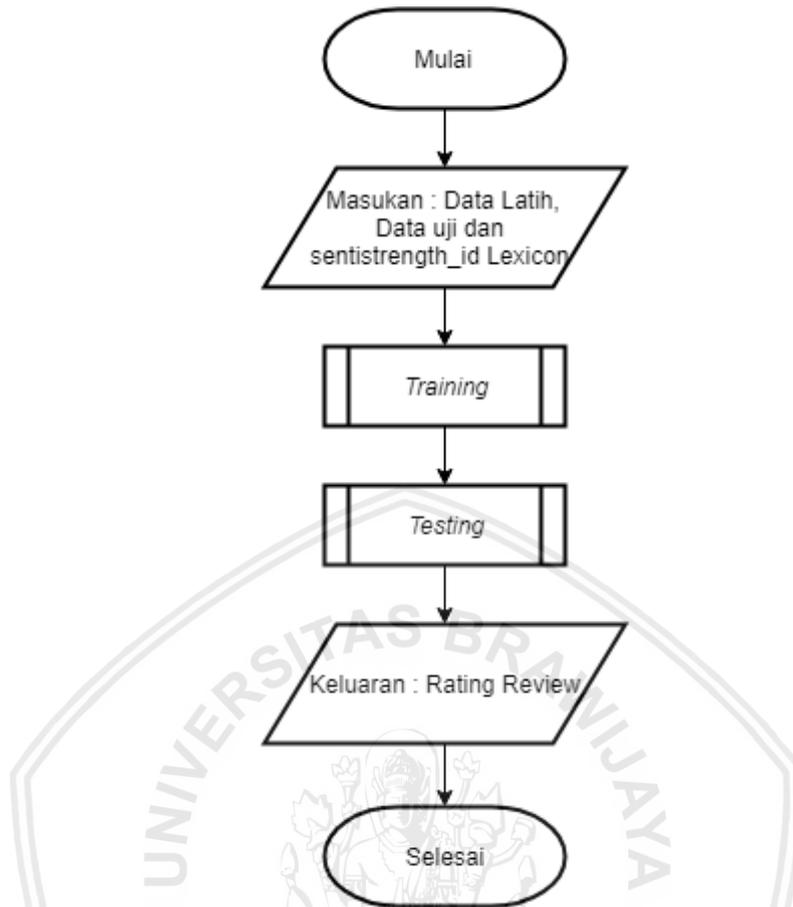
*Review* produk merupakan suatu sarana untuk menyampaikan pendapat pada suatu barang yang telah dibeli dan digunakan oleh pengguna. *Review* tersebut menjadi suatu tolak ukur yang sangat penting bagi pembeli lain yang ingin membeli suatu produk, bahkan bisa suatu *review* dapat digunakan untuk pertimbangan bagi pemilik produk untuk meningkatkan kualitas dari produk tersebut. *Review* biasanya memiliki suatu skala *rating* pada interval tertentu. Untuk yang ada pada penelitian ini, peneliti menggunakan interval dari 1 sampai 5. *Rating* tersebut merupakan suatu tolak ukur dari suatu produk. *Review* tersebut dapat ditemukan di banyak sekali media online baik seperti Facebook, Twitter, YouTube dan lainnya. Namun terkadang suatu *review* produk tidak memiliki *rating* sehingga terkadang menyulitkan untuk pembeli untuk melihat apakah produk yang ingin dibelinya merupakan produk yang bagus atau tidak. Bahkan tidak hanya bagi pembeli, namun untuk pemilik produk terkadang kesulitan untuk mengembangkan produknya. Oleh karena itu dibutuhkan suatu sistem yang dapat digunakan untuk memprediksi suatu *rating* dari *review* suatu produk.

Terdapat banyak metode yang dapat digunakan untuk melakukan suatu prediksi. Salah satu metode yang dapat digunakan adalah metode *Contextual Valence Shifters* untuk pembobotannya yang dikombinasikan dengan *n-gram* yang diharapkan dapat meningkatkan akurasi dan regresi linear sebagai metode prediksinya.

### 4.2 Deskripsi Umum Sistem

Sistem yang dibuat berguna untuk memprediksi *rating* dari *review* konsumen dalam skala *rating* yang ditentukan menggunakan metode *Contextual Valence Shifters* sebagai pembobotannya. Metode *Contextual Valence Shifters* sendiri adalah metode pembobotan yang menggunakan kamus *lexicon* untuk bobot dari setiap katanya, pembobotannya sendiri dikombinasikan dengan metode *n-gram* dengan pengambilan suatu kata dalam kalimat sebanyak  $n=2$  adalah *bigram*,  $n=3$  adalah *trigram*, dan juga kalimat *review* dari *review* konsumen.

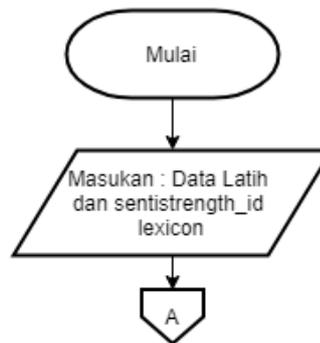
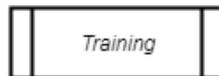
Pada tahap awal, sistem melakukan *pre-processing* dahulu untuk menghilangkan *noise* yang ada pada *text review*. Lalu setelah proses *pre-processing* selesai maka hasil dari *pre-processing* tersebut dilakukan pembobotan dengan metode *Contextual Valence Shifters*. Hasil dari pembobotan tersebut digunakan untuk prediksi dengan metode regresi linear. Untuk lebih jelasnya, alur dari sistem digambarkan secara detail pada Gambar 4.1.

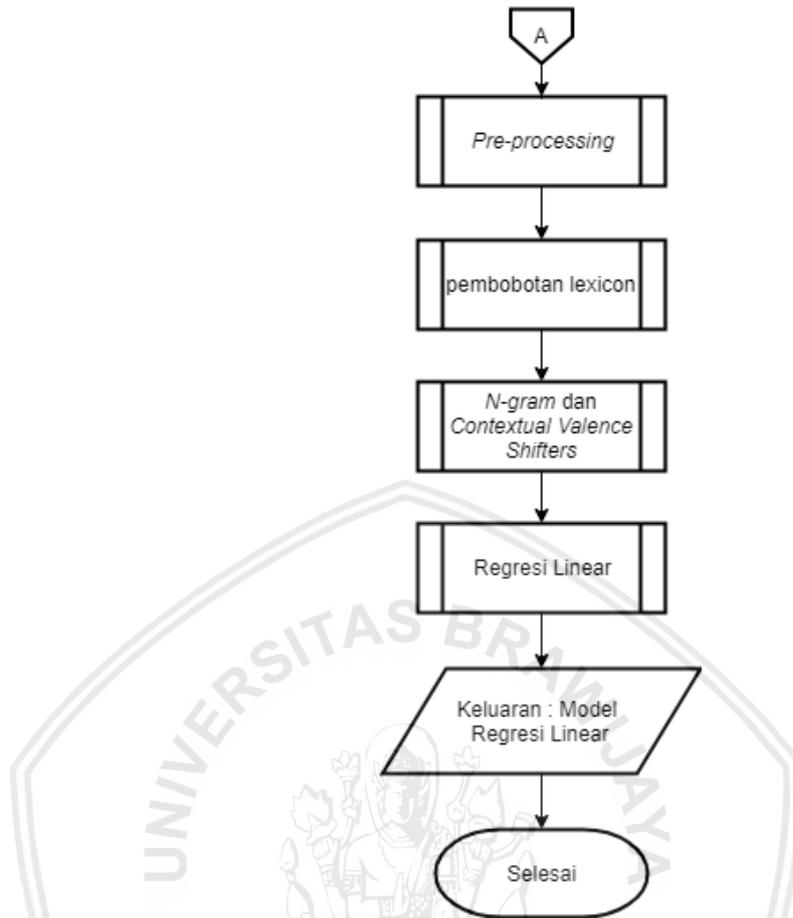


Gambar 4.1 Deskripsi Umum Sistem

### 4.3 Proses Training

*Training* atau *Data Training* adalah kumpulan data awal yang digunakan untuk membantu program memahami bagaimana menerapkan teknologi seperti *Text Mining*, *Neural Network* atau lainnya untuk belajar dan menghasilkan hasil yang baik. *Data Training* adalah bahan yang digunakan komputer untuk mempelajari cara memproses informasi. Untuk proses data *training* sendiri dapat dilihat pada gambar 4.2.

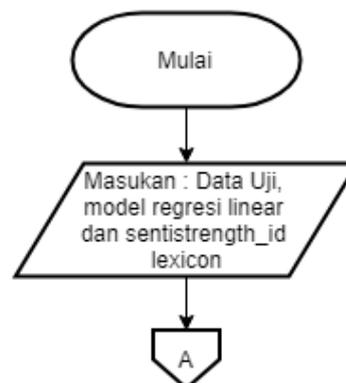
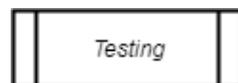


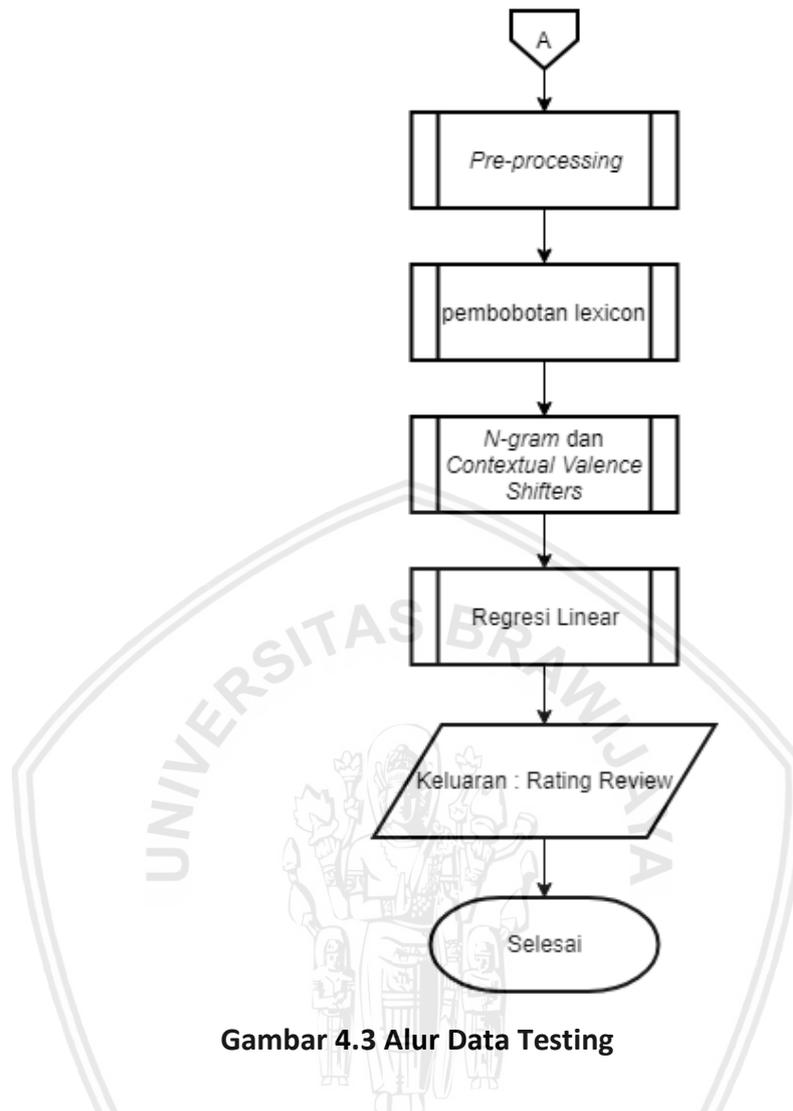


Gambar 4.2 Alur Data Training

#### 4.4 Proses Testing

Testing atau *data testing* adalah proses setelah *training*, dimana hasil dari *data training* di uji pada proses ini, dengan masukan berupa suatu *dataset* yang mirip seperti masukan yang ada pada proses *data training* prosesnya mirip seperti *data training*, namun berbeda keluarannya. Untuk lebih jelasnya dapat di lihat pada gambar 4.3.

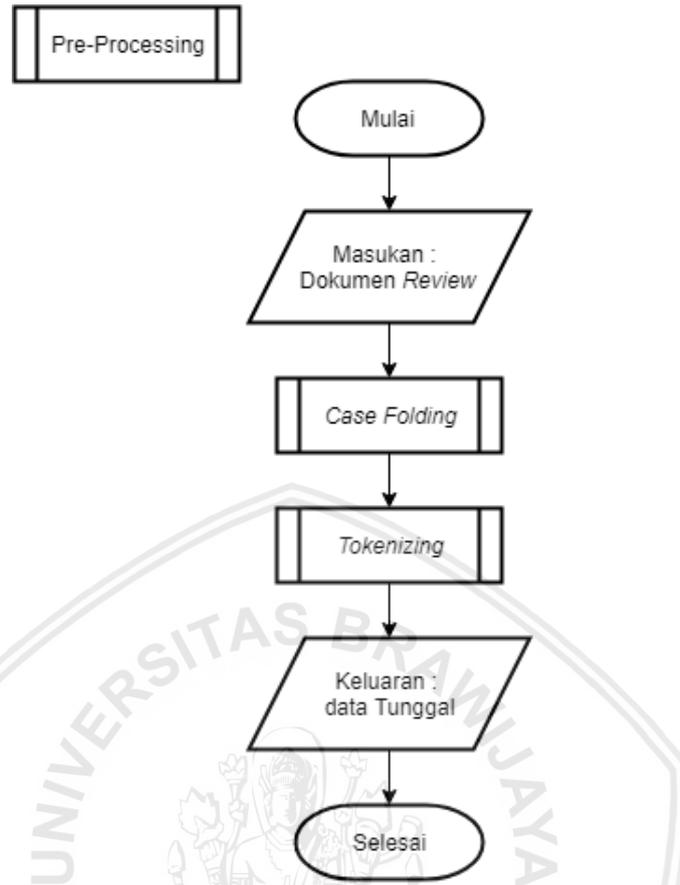




Gambar 4.3 Alur Data Testing

#### 4.5 Pre-Processing

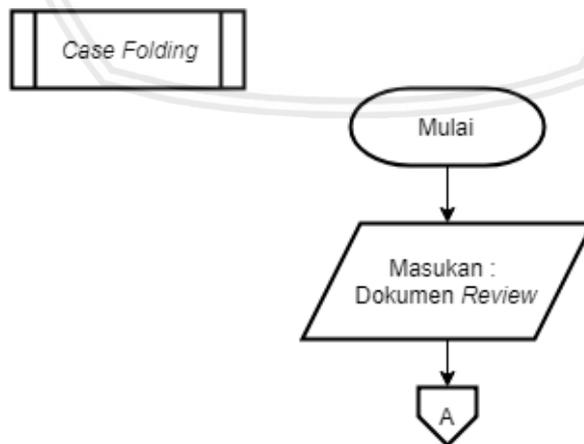
Tahap *pre-processing* adalah tahap awal dari sistem yang peneliti buat, *pre-processing* meliputi proses *case folding* dan *tokenizing*. Sebenarnya pada tahap *pre-processing* terdapat tahap *filtering* dan *stemming*, *filtering* sendiri berguna untuk menghilangkan kata-kata *noise* yang sering muncul di banyak dokumen, sedangkan *Stemming* untuk mendapatkan kata dasar, namun peneliti menghilangkan proses *filtering* dikarenakan proses ini banyak menghilangkan kata-kata penting yang digunakan untuk pembobotan *Contextual Valence Shifters*, seperti kata “sekali”, “kurang” dan juga menghilangkan proses *stemming* dikarenakan kamus lexicon bernama *sentistrength\_id* yang digunakan pada penelitian ini mewadahi kata imbuhan. Hasil dari *pre-processing* digunakan untuk perhitungan bobot tiap dokumen. Untuk lebih jelasnya, alur dari tiap tahap proses ini digambarkan pada Gambar 4.4

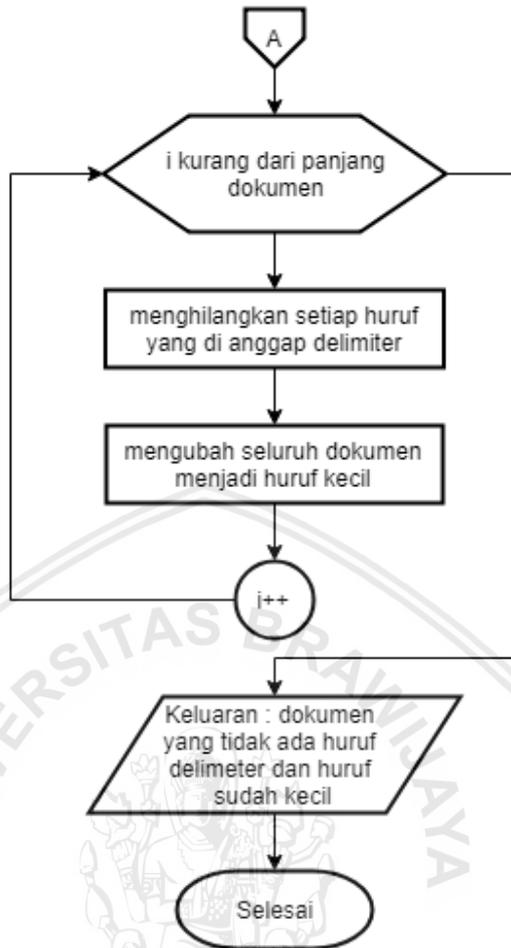
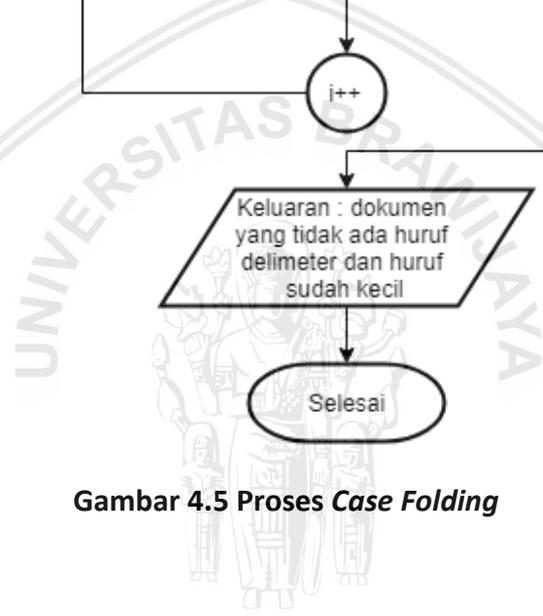


Gambar 4.4 Proses *Pre-Processing*

#### 4.5.1 Case Folding

Tahap pertama dari *pre-processing* adalah proses *case folding*. Proses ini bertujuan untuk mengubah semua huruf yang ada pada dokumen menjadi huruf kecil (*lowercase*). Untuk proses *case folding* dapat dilihat pada Gambar 4.5.

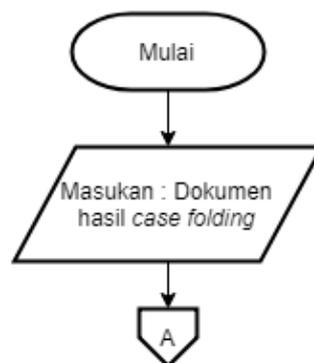




Gambar 4.5 Proses *Case Folding*

**4.5.2 Tokenizing**

*Tokenizing* adalah tahap selanjutnya dari *Pre-Processing*. Pada tahap ini dokumen dipecah berdasarkan spasi yang ada pada dokumen tersebut. Untuk lebih jelasnya dapat di lihat pada Gambar 4.6

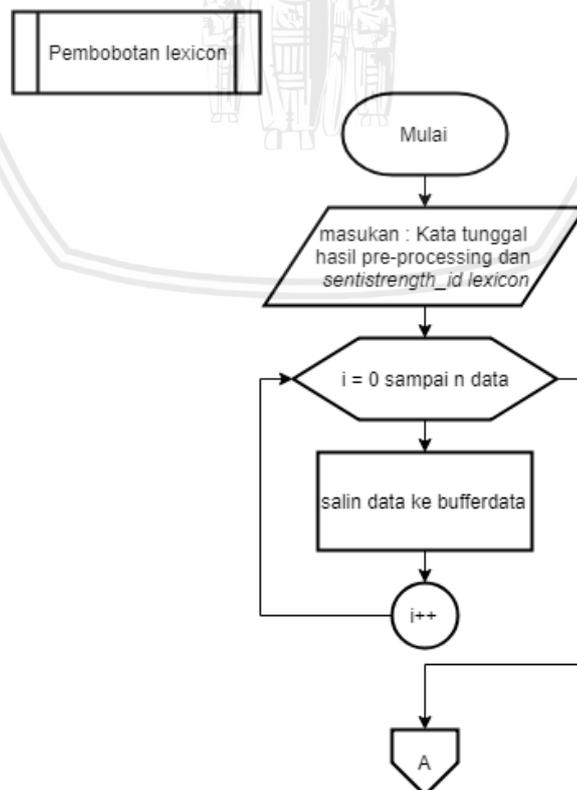


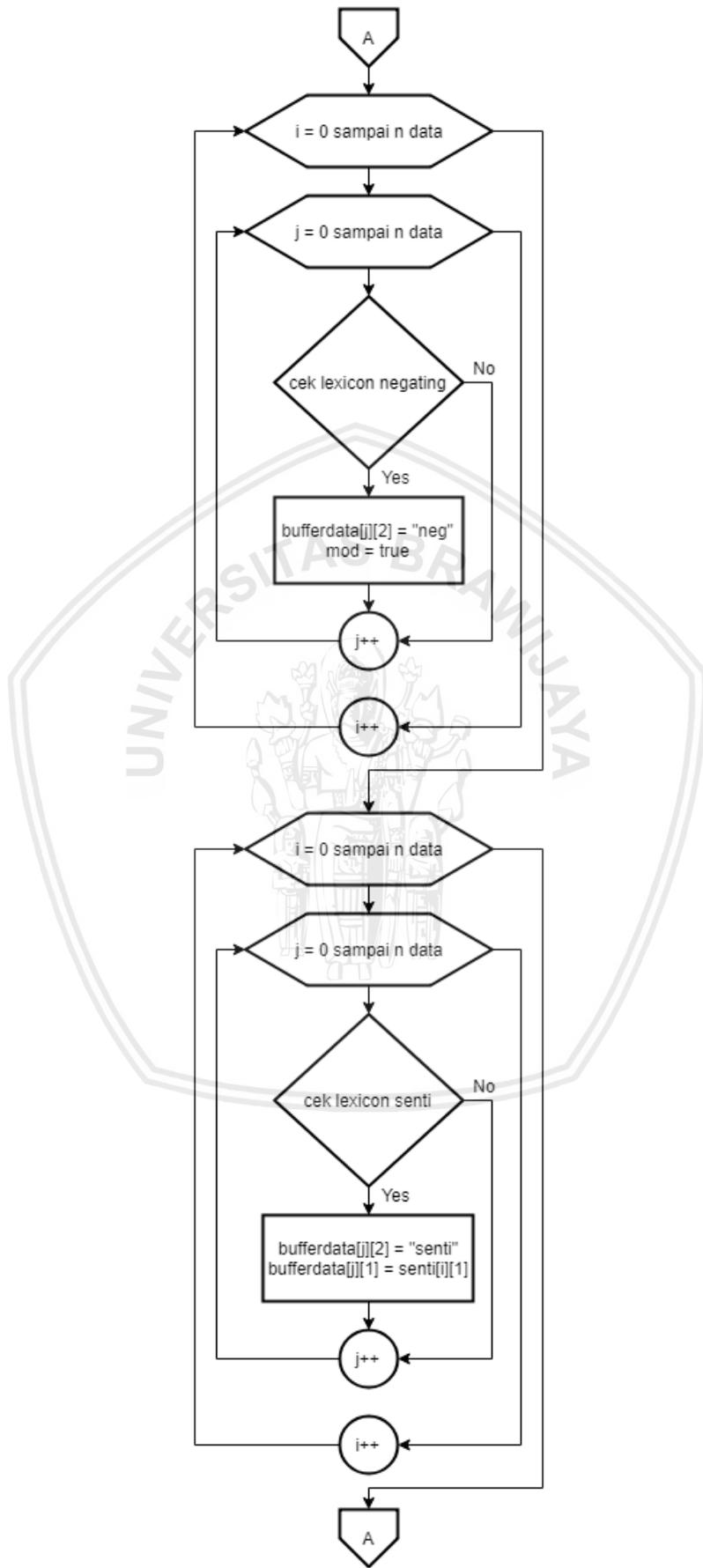


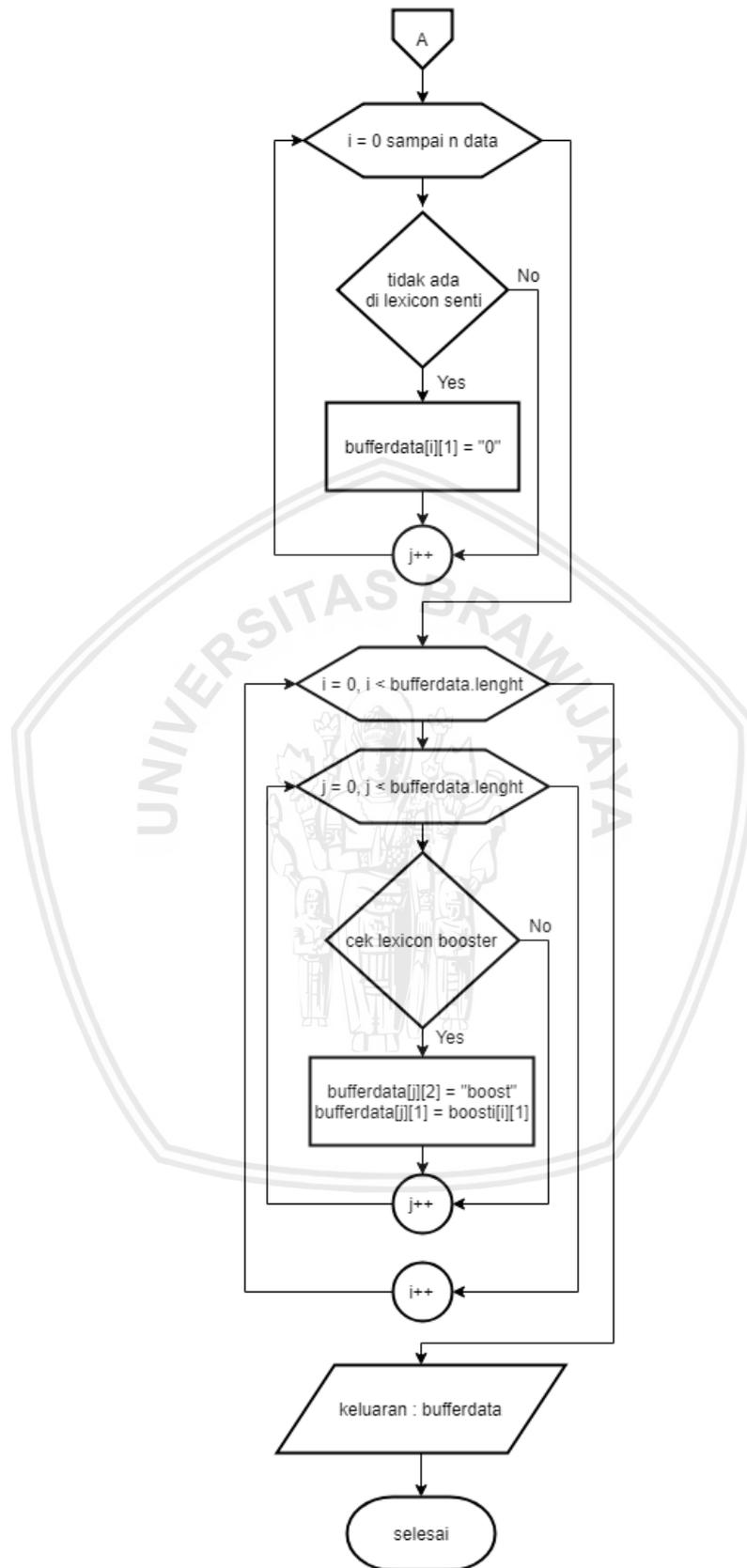
Gambar 4.6 Proses *Tokenizing*

#### 4.6 *Lexicon Based Features*

Pembobotan *Lexicon Based features* merupakan suatu pembobotan berdasarkan kamus. Kamus yang digunakan dalam penelitian ini bernama *sentistrength\_id*. Dalam kamus tersebut terdapat berbagai macam tipe kata, seperti *boosterwords*, *emoticon\_id*, *idioms\_id*, *negatingword*, *questionword* dan *sentiword\_id*. Namun yang akan digunakan dalam penelitian ini hanya *boosterwords*, *negatingwords* dan *sentiword\_id*. Untuk lebih jelasnya dapat dilihat pada Gambar 4.7.



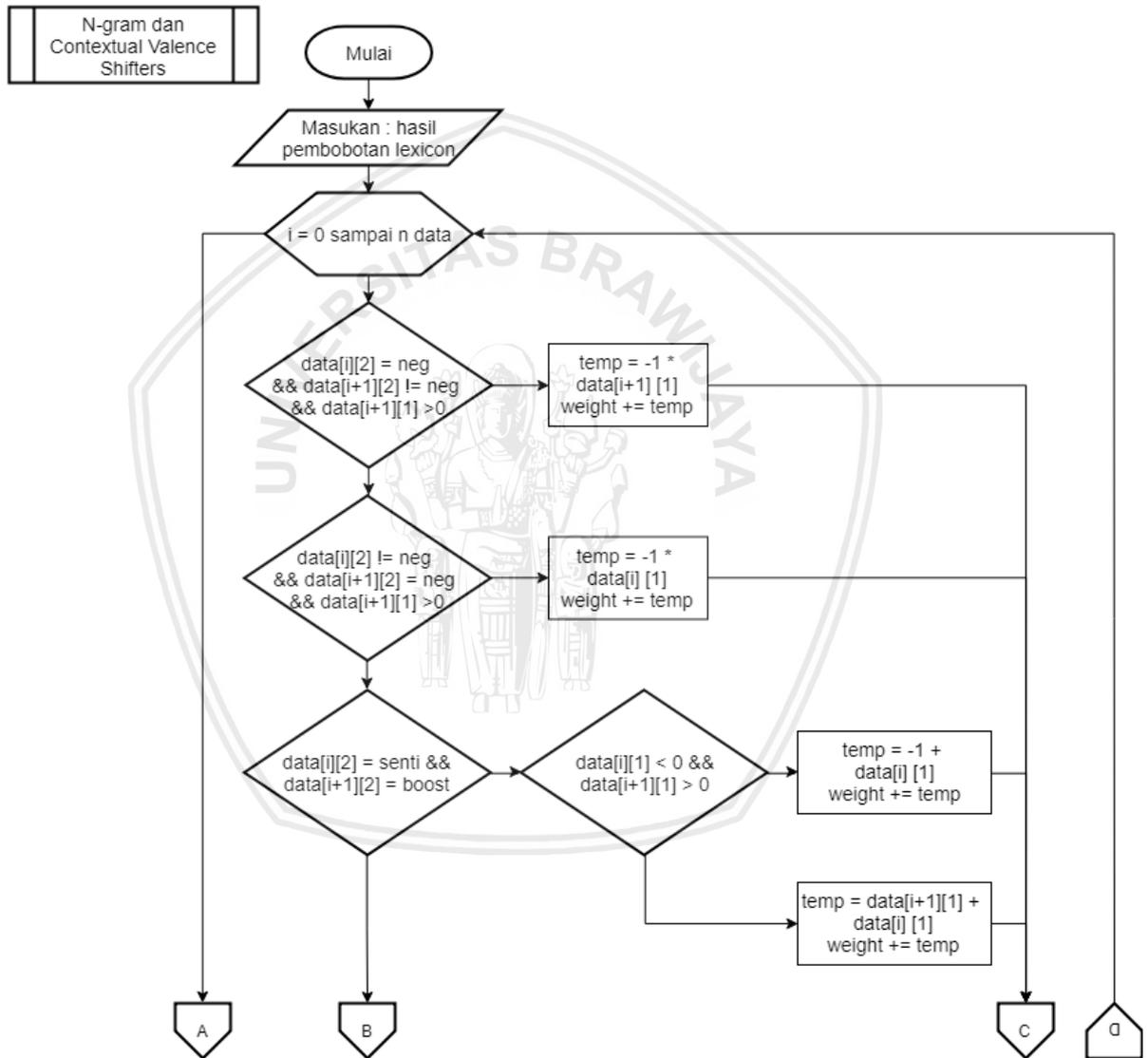


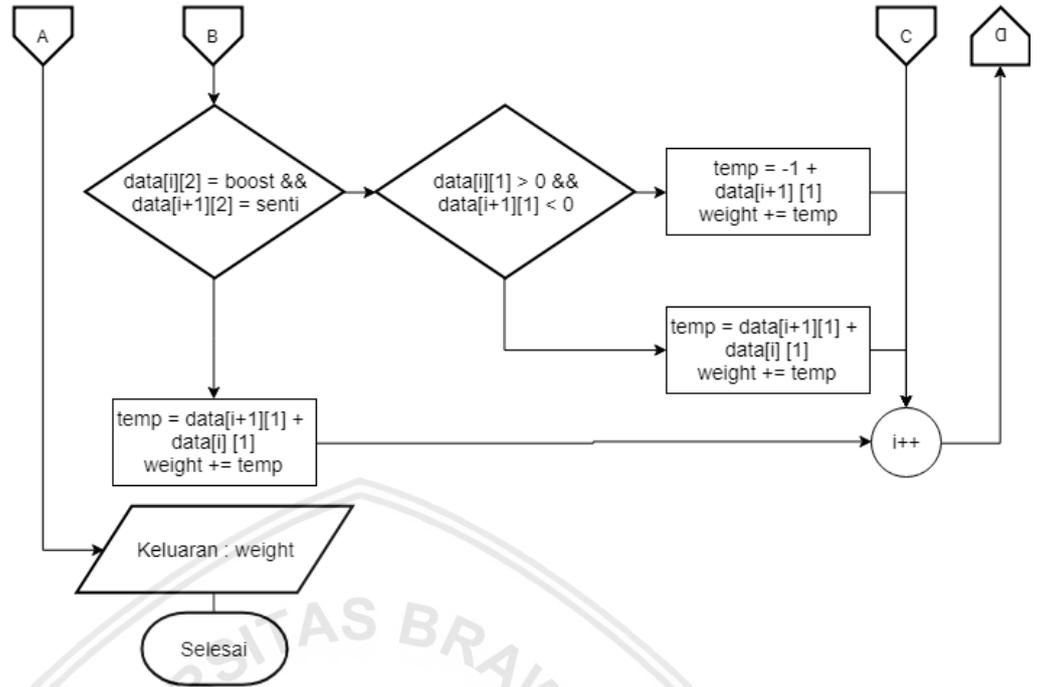


Gambar 4.7 Proses pembobotan *Lexicon Sentistrength\_id*

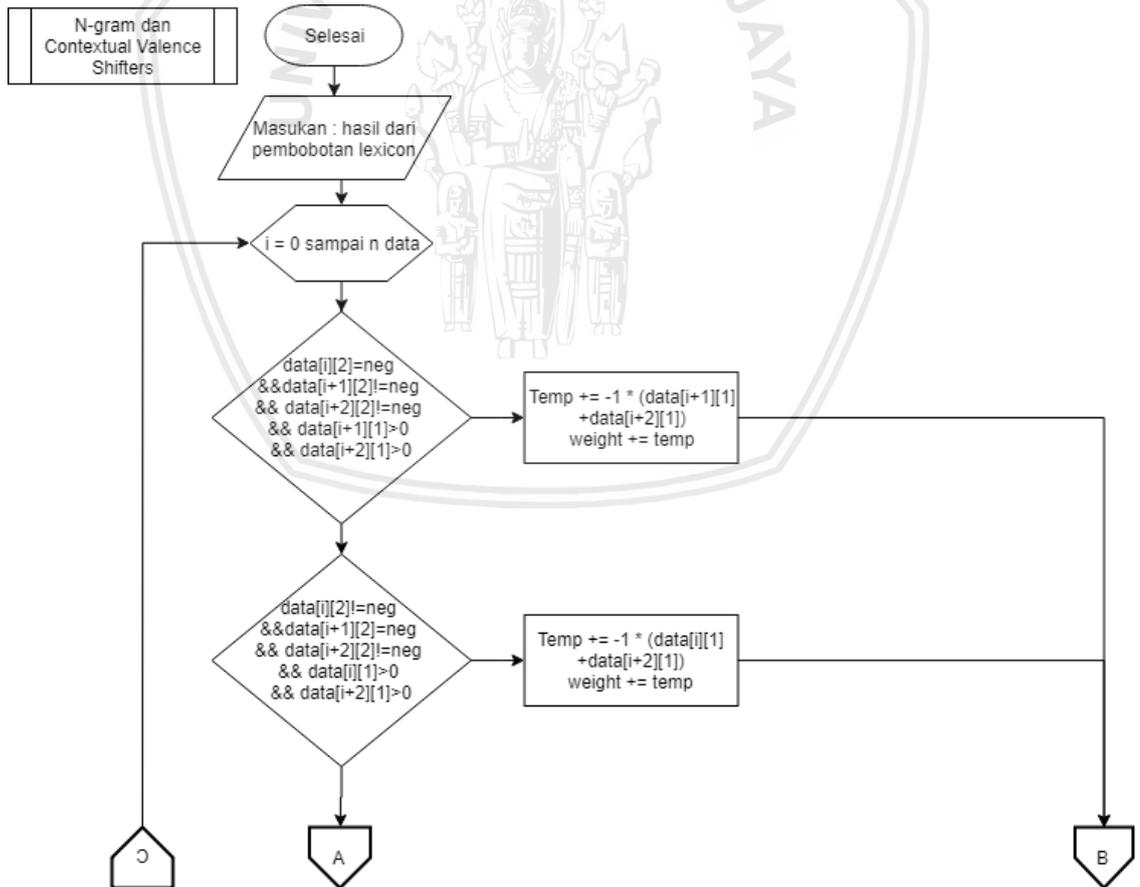
### 4.7 N-gram dan Contextual Valence Shifters

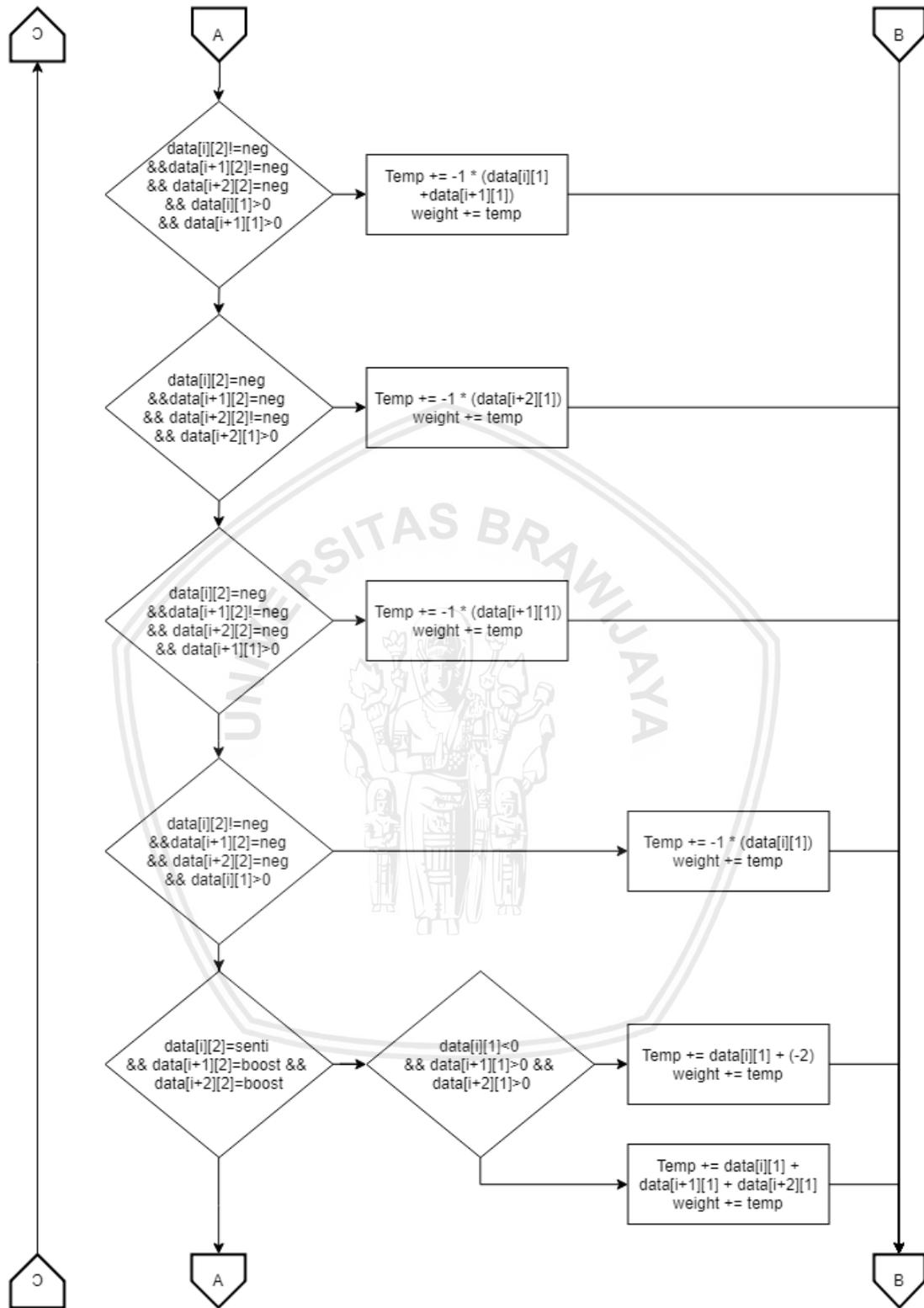
Proses ini adalah proses setelah *pre-processing*, *n-gram* merupakan proses pengambilan kata sebanyak *n* pada suatu dokumen, dengan *n* = 2 adalah bigram, *n* = 3 adalah trigram, dan kalimat yang ada pada *review*. Sedangkan *Contextual Valence Shifters* sendiri untuk pembobotannya berdasarkan *Lexicon Based features*. Untuk lebih jelasnya dapat di lihat pada Gambar 4.8, Gambar 4.9 dan Gambar 4.10

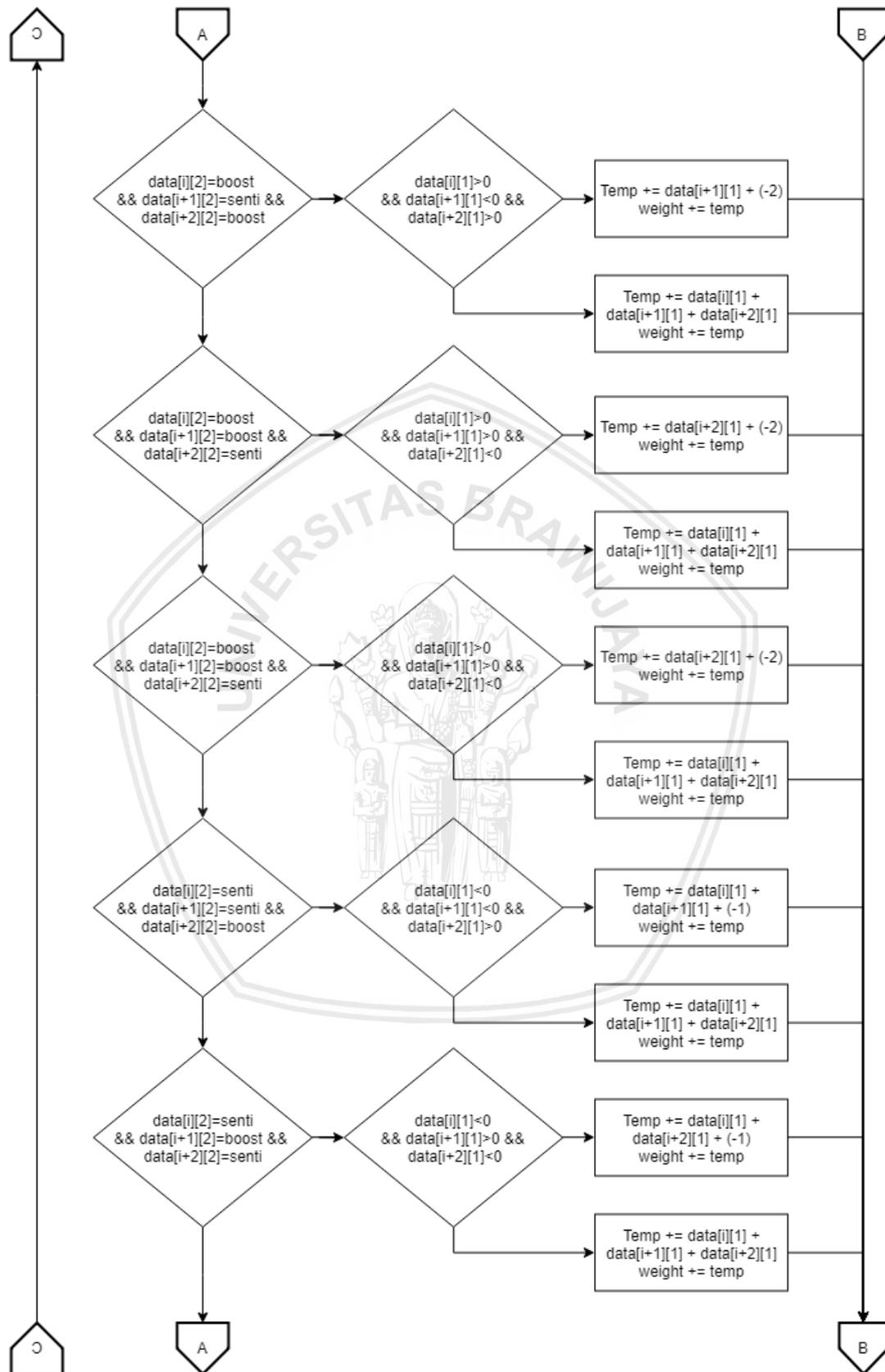


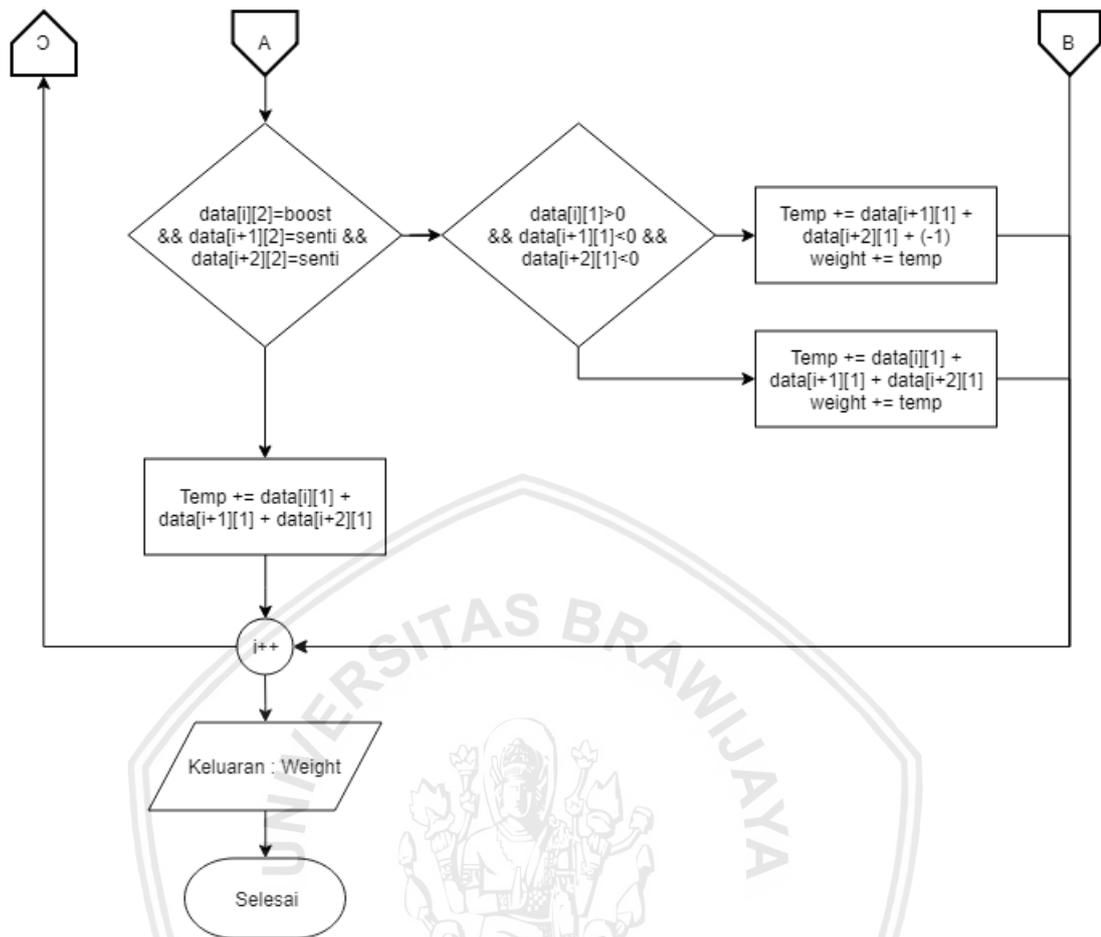


Gambar 4.8 Bigram dan Contextual Valence Shifters

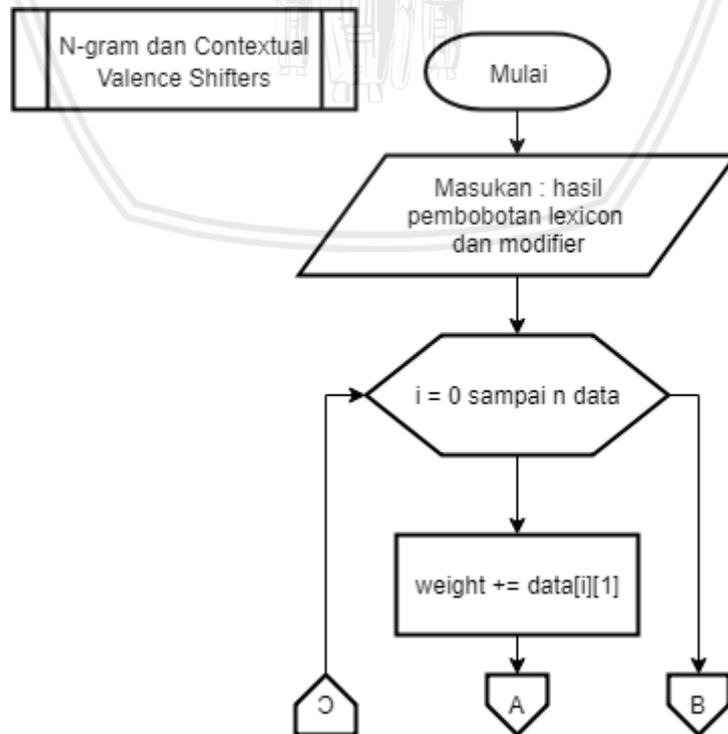


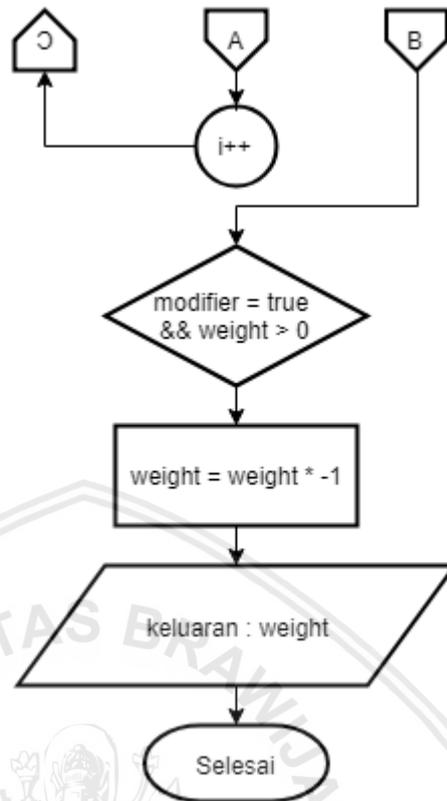






Gambar 4.9 Trigram dan Contextual Valence Shifters

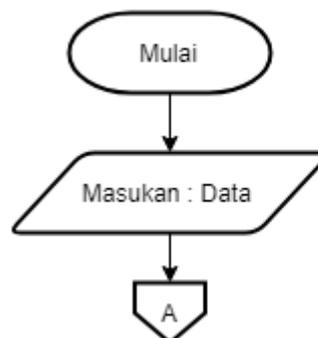
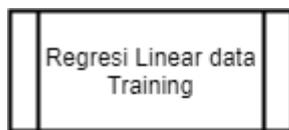


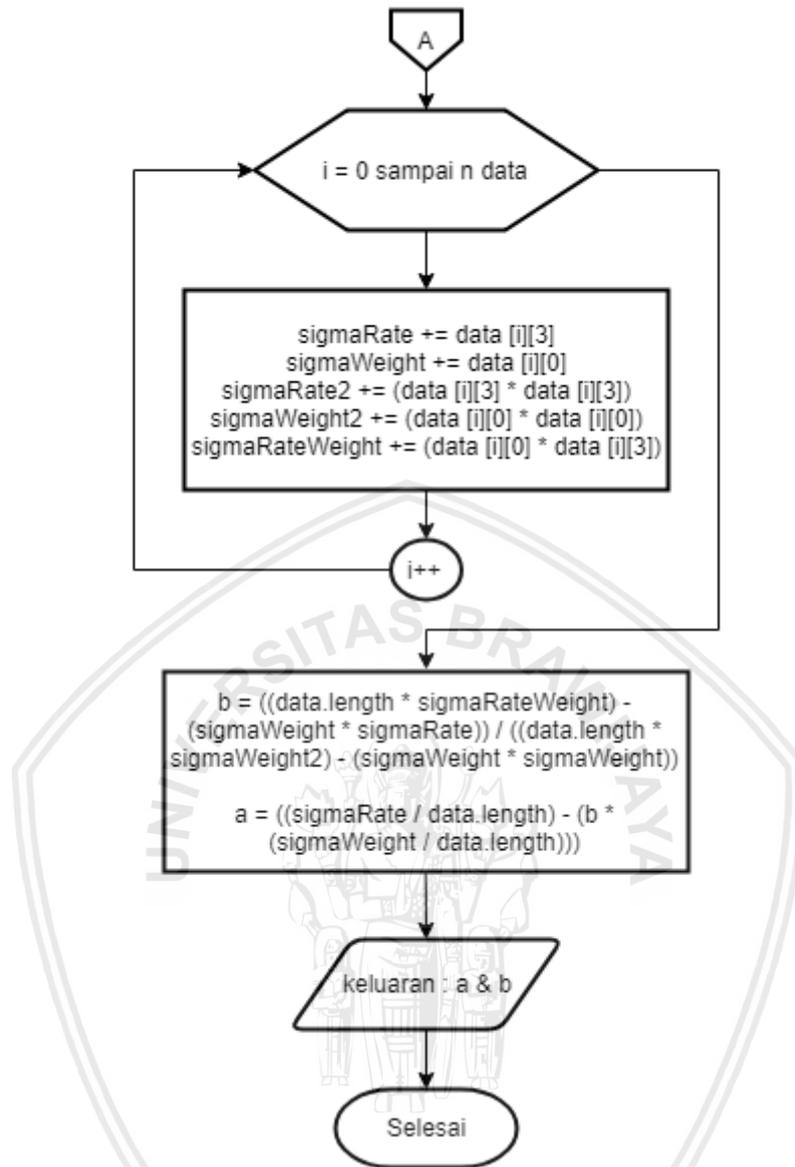


Gambar 4.10 Kalimat review dan *Contextual Valence Shifters*

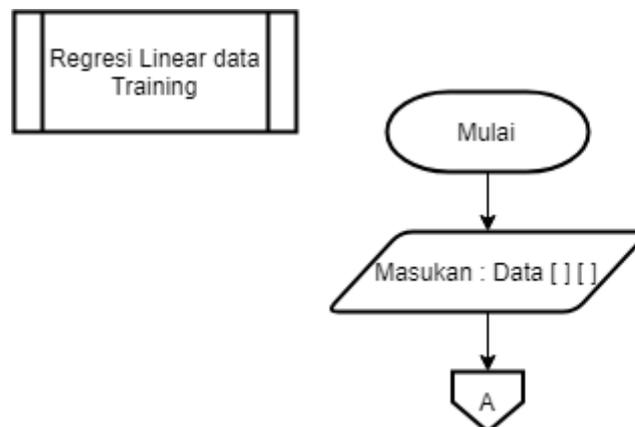
### 4.8 Regresi Linear

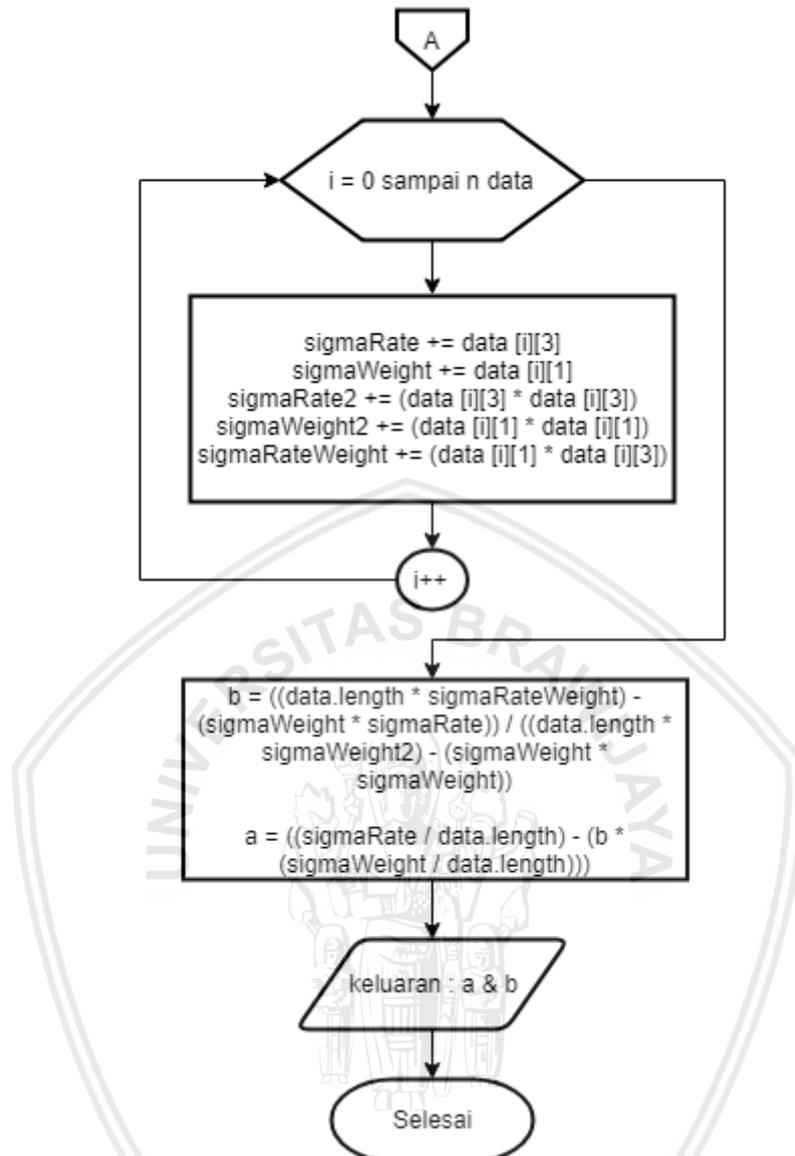
Regresi linear adalah metode yang umum digunakan untuk melakukan prediksi, regresi linear mencoba untuk memodelkan hubungan antara dua variabel dengan menggunakan persamaan linear untuk suatu data yang di amati. Satu variabel dianggap sebagai variabel penjelas (*explanatory variable*), dan yang lain sebagai variabel dependen. Terdapat dua regresi linear yang ada di program, yaitu regresi linear yang ada pada *data training*, dan regresi linear yang ada pada *data testing*. Untuk lebih jelasnya dapat di lihat pada gambar 4.11, 4.12, 4.13 dan 4.14.



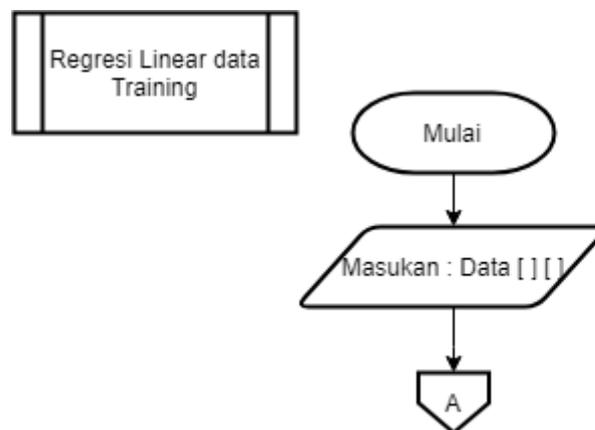


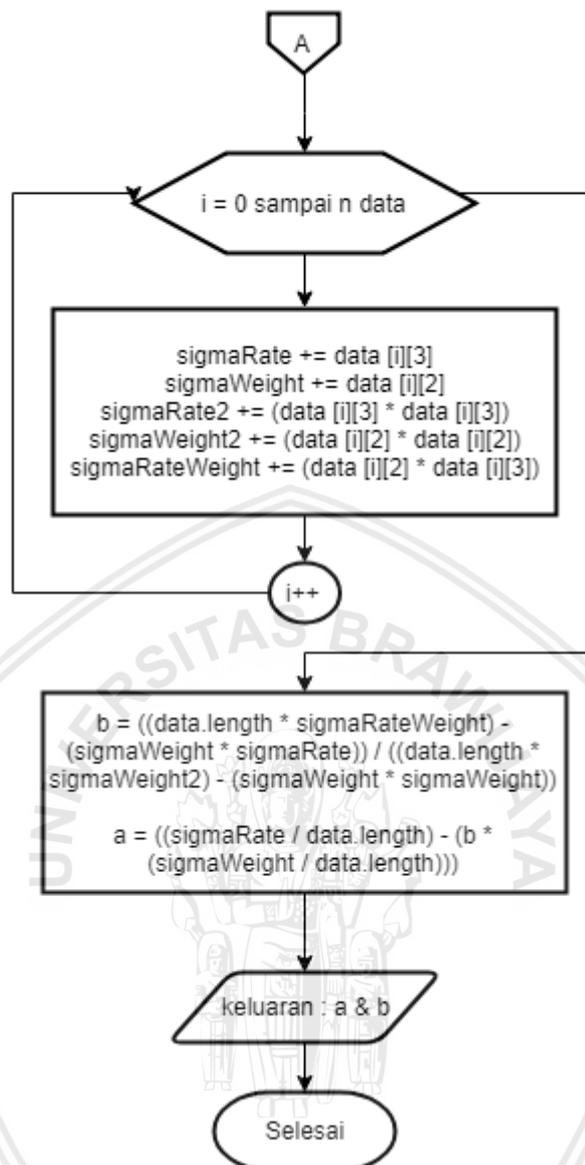
Gambar 4.11 Regresi Linear Data Training bigram



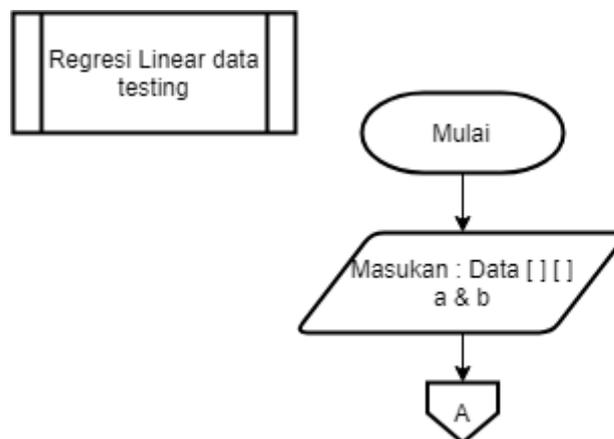


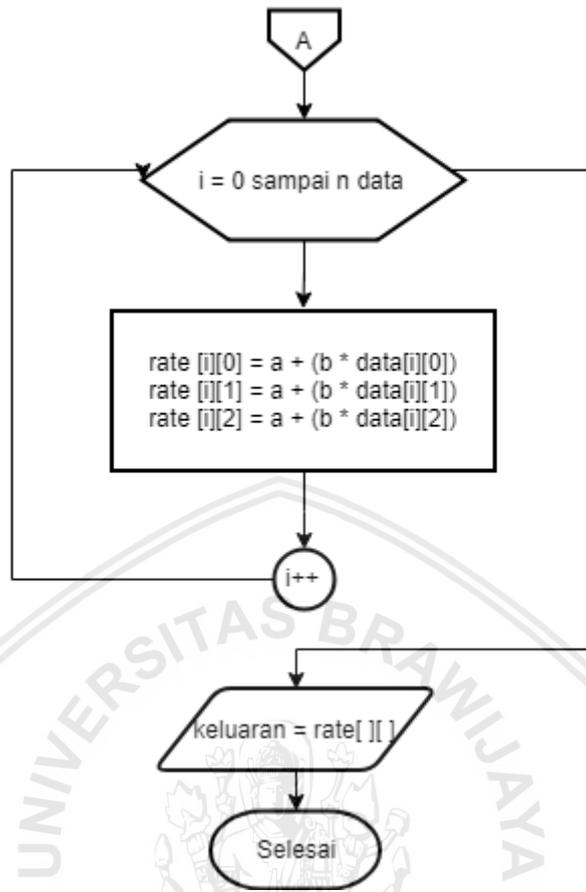
Gambar 4.12 Regresi Linear Data Training Trigram





Gambar 4.13 Regresi Linear Data Training Kalimat





Gambar 4.14 Regresi Linear Data Testing

#### 4.9 Perhitungan Manual

Pada sub bab ini akan dilakukan contoh manualisasi, manualisasi menggunakan 8 data *training* dan 1 data *testing* pada Tabel 4.1, proses awal yang dilakukan adalah dengan *pre-processing*, hasil dari *pre-processing* digunakan untuk memprediksi *rating* menggunakan regresi linear.

Tabel 4.1 Review Produk

No	Review	Rating	
1	kemasan oke, produk di pakai bagus, sip deh	4	Data training
2	warna nya lumayan bagus	3	
3	Warna lipstick bagus tapi kurang cocok buat aku	2	
4	Aku tidak suka banget dengan pensil alis ini karena harganya mahal dan pas dipake warnanya tidak keluar	1	
5	Suka banget sama conditioner ini, wanginya enak banget di rambut aku. Rekomendasi deh!	5	
6	Ternyata sama sekali tidak cocok, rambut jadi lepek lalu berketombe. Beli lagi ? Tidak	2	

7	Aku suka banget pakai produk ini pokoknya bagus dah	4	
8	berharap rambut rontok berkurang, eh malah tambah rontok :( udah pakai Kondisioner juga, tidak cocok kadang buat ketombe gatel ya ampun	1	
9	cocok banget sama aku, pokoknya bagus deh	?	Data testing

Data pada Tabel 4.1 diproses pada *pre-processing* yaitu proses *case folding* dan *tokenizing*. Hasil yang didapat setelah proses *pre-processing* dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Hasil proses *Pre-processing***

No	Review	Rating	Keterangan
1	kemasan   oke   produk   di   pakai   bagus   sip   deh	4	Data training
2	warna   nya   lumayan   bagus	3	
3	warna   lipstick   bagus   tapi   kurang   cocok   buat   aku	2	
4	aku   tidak   suka   banget   dengan   pensil   alis   ini   karena   harganya   mahal   dan   pas   dipake   warnanya   tidak   keluar	1	
5	suka   banget   sama   conditioner   ini   wanginya   enak   banget   di   rambut   aku   rekomendasi   deh	5	
6	ternyata   sama   sekali   tidak   cocok   rambut   jadi   lepek   lalu   berketombe   beli   lagi   tidak	2	
7	aku   suka   banget   pakai   produk   ini   pokoknya   bagus   dah	5	
8	berharap   rambut   rontok   berkurang   eh   malah   tambah   rontok   udah   pakai   kondisioner   juga   tidak   cocok   kadang   buat   ketombe   gatel   ya   ampun	1	
9	cocok   banget   sama   aku   pokoknya   bagus   deh	?	Data testing

Setelah proses *pre-processing* selesai, selanjutnya setiap kata yang ada pada dokumen diberi bobot berdasarkan *Lexicon Based features* dengan catatan apabila dalam kamus tidak ada kata yang sama, maka nilai dari kata tersebut adalah nol. Pembobotan sendiri menggunakan metode *Contextual Valence Shifters*. Untuk lebih jelasnya, pembobotan perkata dapat dilihat pada tabel 4.3.



Tabel 4.3 Pembobotan Bigram

No	Kata (Bigram)	bobot	bobot total	rating
1	kemasan oke	3	14	4
2	oke produk	3		
3	produk di	0		
4	di pakai	0		
5	pakai bagus	4		
6	bagus sip	4		
7	sip deh	0		
8	warna nya	0	8	3
9	nya lumayan	2		
10	lumayan bagus	6		
11	warna lipstik	0	12	2
12	lipstik bagus	4		
13	bagus tapi	4		
14	tapi kurang	-1		
15	kurang cocok	2		
16	cocok buat	3		
17	buat aku	0		
18	aku tidak	0	-7	1
19	tidak suka	-4		
20	suka banget	5		
21	banget dengan	1		
22	dengan pensil	0		
23	pensil alis	0		
24	alis ini	0		
25	ini karena	0		
26	karena harganya	0		
27	harganya mahal	-3		
28	mahal dan	-3		
29	dan pas	0		
30	pas dipake	0		
31	dipake warnanya	0		
32	warnanya tidak	0		
33	tidak keluar	-3		
34	suka banget	5		
35	banget sama	1		
36	sama conditioner	0		
37	conditioner ini	0		
38	ini wanginya	0		
39	wanginya enak	4		
40	enak banget	5		



41	banget di	1		
42	di rambut	0		
43	rambut aku	0		
44	aku rekomendasi	0		
45	rekomendasi deh	0		
46	ternyata sama	0		
47	sama sekali	0		
48	sekali tidak	-1		
49	tidak cocok	-3		
50	cocok rambut	0		
51	rambut jadi	0	-4	2
52	jadi lepek	0		
53	lepek lalu	0		
54	lalu berketombe	0		
55	berketombe beli	0		
56	beli lagi	0		
57	lagi tidak	0		
58	aku suka	4		
59	suka banget	5		
60	banget pakai	1	18	5
61	pakai produk	0		
62	produk ini	0		
63	ini pokoknya	0		
64	pokoknya bagus	4		
65	bagus dah	4		
66	berharap rambut	-1		
67	rambut rontok	0		
68	rontok berkurang	-2		
69	berkurang eh	-2		
70	eh mahal	0		
71	malah tambah	1		
72	tambah rontok	1		
73	rontok udah	0		
74	udah pakai	0	-3	1
75	pakai kondisioner	0		
76	kondisioner juga	0		
77	juga tidak	0		
78	tidak cocok	-3		
79	cocok kadang	3		
80	kadang buat	0		
81	buat ketombe	0		
82	ketombe gatel	0		
83	gatel ya	0		

84	ya ampun	0		
Data Uji				
1	cocok banget	4	13	?
2	banget sama	1		
3	sama aku	0		
4	aku pokoknya	0		
5	pokoknya bagus	4		
6	bagus deh	4		

**Tabel 4.4 Pembobotan Trigram**

No	Kata (Trigram)	bobot	bobot total	Rating
1	kemasan oke produk	3	18	4
2	oke produk di	3		
3	produk di pakai	0		
4	di pakai bagus	4		
5	pakai bagus sip	4		
6	bagus sip deh	4		
7	warna nya lumayan	2	8	3
8	nya lumayan bagus	6		
9	warna lipstik bagus	4	18	2
10	lipstik bagus tapi	4		
11	bagus tapi kurang	3		
12	tapi kurang cocok	2		
13	kurang cocok buat	2		
14	cocok buat aku	3		
15	aku tidak suka	0	-11	1
16	tidak suka banget	-5		
17	suka banget dengan	5		
18	banget dengan pensil	1		
19	dengan pensil alis	0		
20	pensil alis ini	0		
21	alis ini karena	0		
22	ini karena harganya	0		
23	karena harganya mahal	-3		
24	harganya mahal dan	-3		
25	mahal dan pas	-3		
26	dan pas dipake	0		
27	pas dipake warnanya	0		
28	dipake warnanya tidak	0		
29	warnanya tidak keluar	-3		
30	suka banget sama	5	21	5
31	banget sama conditioner	1		

32	sama conditioner ini	0		
33	conditioner ini wanginya	0		
34	ini wanginya enak	4		
35	wanginya enak banget	5		
36	enak banget di	5		
37	banget di rambut	1		
38	di rambut aku	0		
39	rambut aku rekomendasi	0		
40	aku rekomendasi deh	0		
41	ternyata sama sekali	1		
42	sama sekali tidak	-1		
43	sekali tidak cocok	-3		
44	tidak cocok rambut	-3		
45	cocok rambut jadi	3		
46	rambut jadi lepek	0	-3	2
47	jadi lepek lalu	0		
48	lepek lalu berketombe	0		
49	lalu berketombe beli	0		
50	berketombe beli lagi	0		
51	beli lagi tidak	0		
52	aku suka banget	5		
53	suka banget pakai	4		
54	banget pakai produk	1		
55	pakai produk ini	0	18	5
56	produk ini pokoknya	0		
57	ini pokoknya bagus	4		
58	pokoknya bagus dah	4		
59	berharap rambut rontok	-1		
60	rambut rontok berkurang	-2		
61	rontok berkurang eh	-2		
62	berkurang eh malah	-2		
63	eh malah tambah	1		
64	malah tambah rontok	1		
65	tambah rontok udah	1		
66	rontok udah pakai	0		
67	udah pakai kondisioner	0	-7	1
68	pakai kondisioner juga	0		
69	kondisioner juga tidak	0		
70	juga tidak cocok	-3		
71	tidak cocok kadang	-3		
72	cocok kadang buat	3		
73	kadang buat ketombe	0		
74	buat ketombe gatal	0		

75	ketombe gatel ya	0		
76	gatel ya ampun	0		
Data Uji				
1	cocok banget sama	4	13	?
2	banget sama aku	1		
3	sama aku pokoknya	0		
4	aku pokoknya bagus	4		
5	pokoknya bagus dah	4		

**Tabel 4.5 Pembobotan kalimat review**

No	Kata	bobot	bobot total	rating
1	kemasan	0	7	4
2	oke	3		
3	produk	0		
4	di	0		
5	pakai	0		
6	bagus	4		
7	sip	0		
8	deh	0	6	3
9	warna	0		
10	nya	0		
11	lumayan	2		
12	bagus	4		
13	warna	0	6	2
14	lipstick	0		
15	bagus	4		
16	tapi	0		
17	kurang	-1		
18	cocok	3		
19	buat	0	-1	1
20	aku	0		
21	aku	0		
22	tidak	negasi		
23	suka	4		
24	banget	1		
25	dengan	0		
26	pensil	0		
27	alis	0		
28	ini	0		
29	karena	0		
30	harganya	0		

31	mahal	-3		
32	dan	0		
33	pas	0		
34	dipake	0		
35	warnya	0		
36	tidak	negasi		
37	keluar	-3		
38	suka	4		
39	banget	1		
40	sama	0		
41	conditioner	0		
42	ini	0		
43	wanginya	0		
44	enak	4	10	5
45	banget	1		
46	di	0		
47	rambut	0		
48	aku	0		
49	rekomendasi	0		
50	deh	0		
51	ternyata	0		
52	sama	0		
53	sekali	1		
54	tidak	negasi		
55	cocok	3		
56	rambut	0		
57	jadi	0	-4	2
58	lepek	0		
59	lalu	0		
60	berketombe	0		
61	beli	0		
62	lagi	0		
63	tidak	negasi		
64	aku	0		
65	suka	4		
66	banget	1		
67	pakai	0		
68	produk	0	9	5
69	ini	0		
70	pokoknya	0		
71	bagus	4		
72	dah	0		
73	berharap	-1	-1	1

74	rambut	0		
75	rontok	0		
76	berkurang	-2		
77	eh	0		
78	malah	0		
79	tambah	1		
80	rontok	0		
81	udah	0		
82	pakai	0		
83	kondisioner	0		
84	juga	0		
85	tidak	negasi		
86	cocok	3		
87	kadang	0		
88	buat	0		
89	ketombe	0		
90	gatel	0		
91	ya	0		
92	ampun	0		
Data Uji				
1	cocok	3	8	?
2	banget	1		
3	sama	0		
4	aku	0		
5	pokoknya	0		
6	bagus	4		
7	deh	0		

Setelah kata diberi bobot, maka data *training* selanjutnya masuk proses awal regresi, yaitu untuk mencari beberapa nilai yang digunakan untuk regresi linear, untuk lebih jelasnya dapat dilihat pada Tabel 4.4.

**Tabel 4.6 Proses Regresi Bigram**

data uji					
no	bobot	rating	X <sup>2</sup>	Y <sup>2</sup>	X*Y
1	14	4	196	16	56
2	8	3	64	9	24
3	12	2	144	4	24
4	-7	1	49	1	-7
5	16	5	256	25	80

6	-4	2	16	4	-8
7	18	5	324	25	90
8	-3	1	9	1	-3
n	ΣX	ΣY	ΣX <sup>2</sup>	ΣY <sup>2</sup>	ΣX*Y
8	54	23	1058	85	256

**Tabel 4.7 Proses Regresi Trigram**

data uji					
no	bobot	rating	X <sup>2</sup>	Y <sup>2</sup>	X*Y
1	18	4	324	16	72
2	8	3	64	9	24
3	18	2	324	4	36
4	-11	1	121	1	-11
5	21	5	441	25	105
6	-3	2	9	4	-6
7	18	5	324	25	90
8	-7	1	49	1	-7
n	ΣX	ΣY	ΣX <sup>2</sup>	ΣY <sup>2</sup>	ΣX*Y
8	62	23	1656	85	303

**Tabel 4.8 Proses Regresi Kalimat review**

data uji					
no	bobot	rating	X <sup>2</sup>	Y <sup>2</sup>	X*Y
1	7	4	49	16	28
2	6	3	36	9	18
3	6	2	36	4	12
4	-1	1	1	1	-1
5	10	5	100	25	50
6	-4	2	16	4	-8
7	9	5	81	25	45
8	-1	1	1	1	-1
n	ΣX	ΣY	ΣX <sup>2</sup>	ΣY <sup>2</sup>	ΣX*Y
8	32	23	320	85	143

Setelah proses sebelumnya dilakukan, maka selanjutnya adalah mencari nilai "a" dan nilai "b" untuk setiap tipe *n-gram* yang ada pada persamaan umum dari regresi linear, untuk lebih jelasnya dapat dilihat sebagai berikut.

Untuk Bigram:

$$Y = a + b.X$$

$$b = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}$$



$$b = \frac{(8 * 256) - (54 * 23)}{(8 * 1058) - (54 * 54)}$$

$$b = 0,15$$

$$a = \frac{\sum y}{n} - b \frac{\sum x}{n}$$

$$a = \frac{23}{8} - 0,15 \frac{54}{8}$$

$$a = 1.894$$

$$Y = 1.894 + 0,15 X$$

Untuk Trigram:

$$Y = a + b.X$$

$$b = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}$$

$$b = \frac{(8 * 303) - (62 * 23)}{(8 * 1656) - (62 * 62)}$$

$$b = 0,11$$

$$a = \frac{\sum y}{n} - b \frac{\sum x}{n}$$

$$a = \frac{23}{8} - 0,11 \frac{64}{8}$$

$$a = 2.053$$

$$Y = 2.053 + 0,11 X$$

Untuk kalimat *review*:

$$Y = a + b.X$$

$$b = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2}$$

$$b = \frac{(8 * 143) - (32 * 23)}{(8 * 320) - (32 * 32)}$$

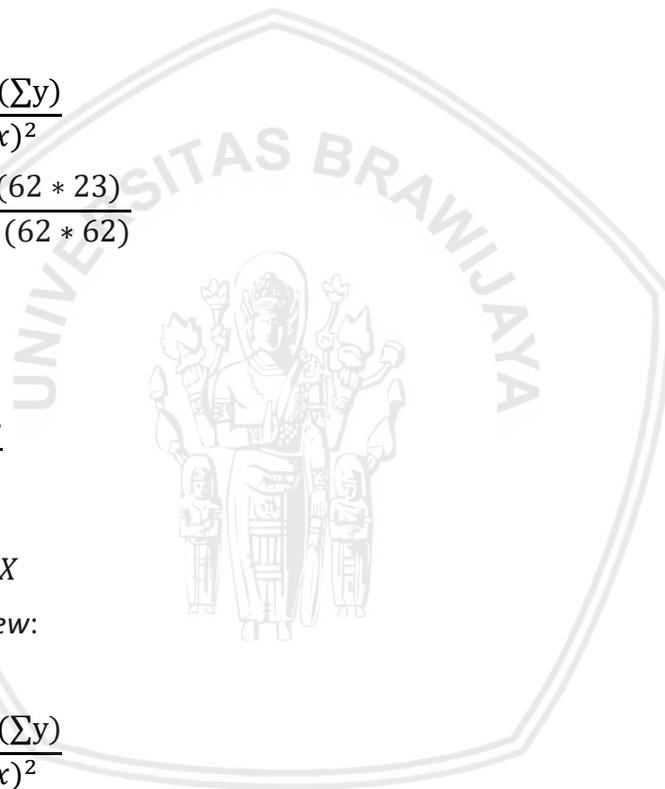
$$b = 0,266$$

$$a = \frac{\sum y}{n} - b \frac{\sum x}{n}$$

$$a = \frac{23}{8} - 0,266 \frac{32}{8}$$

$$a = 1.813$$

$$Y = 1.813 + 0,266 X$$



Setelah persamaan umum dan nilai dari “a” dan “b” didapat, maka selanjutnya dapat menghitung *rating* yaitu dengan memasukkan bobot dari data uji. Dapat dilihat sebagai berikut.

Untuk Bigram:

$$Y = 1.894 + 0,15 (13)$$

$$Y = 3,78$$

$$Y \approx 4$$

Untuk Trigram:

$$Y = 2.053 + 0,11 (13)$$

$$Y = 3,43$$

$$Y \approx 3$$

Untuk Kalimat *review*:

$$Y = 1.813 + 0,266 (8)$$

$$Y = 3,9$$

$$Y \approx 4$$

Dari hasil perhitungan yang telah dilakukan di atas, maka didapatkan hasil dari prediksi *rating* data uji yaitu 4 untuk kalimat dan *bigram*, dan 3 untuk *trigram*.

#### 4.10 Perancangan Pengujian

Tabel 4.9 Perancangan Pengujian

Fold	N-gram	Model Pengujian		
		Toleransi (0)	Toleransi (1)	Sentimen Review
1	Bigram			
	Trigram			
	Kalimat review			
2	Bigram			
	Trigram			
	Kalimat review			
3	Bigram			
	Trigram			
	Kalimat review			
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
10	Bigram			
	Trigram			
	Kalimat review			

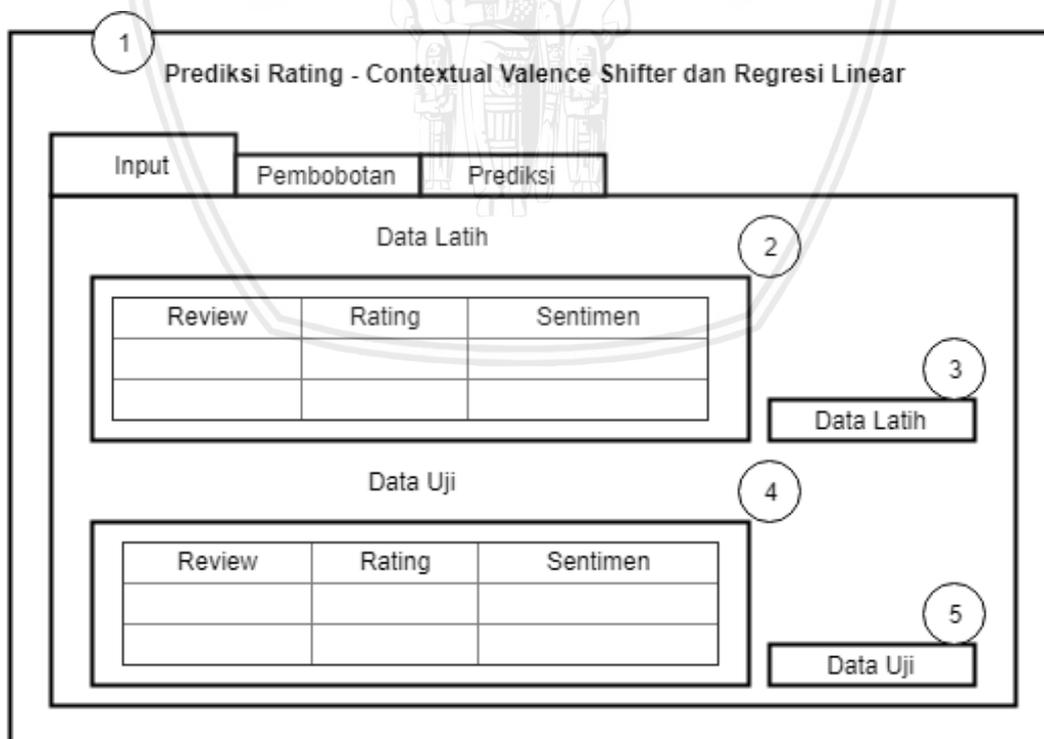


Proses pengujian dilakukan untuk mengetahui akurasi *rating* dengan metode *Contextual Valence Shifters* dan Regresi Linear. Pengujian menggunakan metode *K-Fold Cross Validation* dengan jumlah *K* adalah 10. Terdapat tiga model pengujian yang digunakan yaitu:

1. Model pengujian dengan toleransi 0, yang berarti seluruh hasil dari proses *rating* harus sama dengan *rating* pada data uji
2. Model pengujian dengan toleransi 1, yang berarti hasil dari prediksi *rating* boleh berbeda dengan data uji dengan maksimal 1 tingkat *rating*.
3. Model pengujian dengan sentimen *review*, pada pengujian ini dibagi menjadi dua, yaitu sentimen positif dan negatif. Karena dalam proses *rating* di bagi menjadi 5, maka harus ada ambang batas (*threshold*). *Threshold* yang digunakan adalah 3, *rating* 3 ke bawah merupakan sentimen negatif dan di atas 3 adalah sentimen positif.

#### 4.11 Perancangan Antarmuka

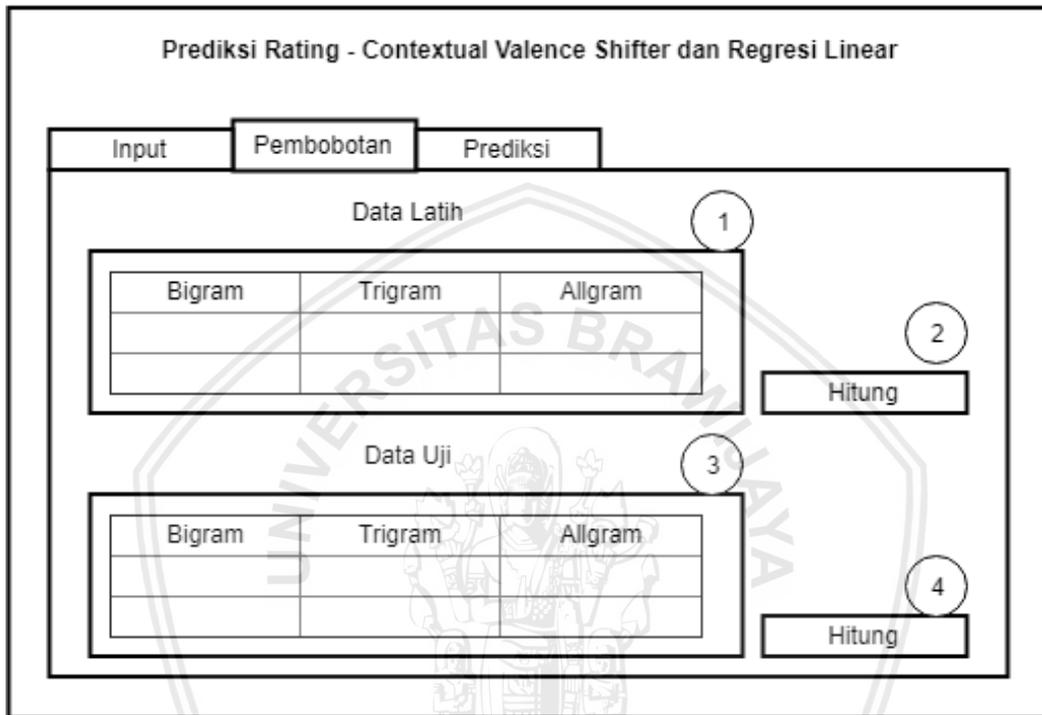
Antarmuka adalah suatu mekanisme dimana sistem dan pengguna saling berhubungan. Antarmuka yang dibuat berupa satu *window*, dengan terdiri dari tiga panel (*tab*), panel utama berisi tabel data *training* dan data *testing*. Panel kedua berisi hasil dari pembobotan *Contextual Valence Shifters*. Panel ketiga berisi hasil dari prediksi berupa *rating* dari proses regresi linear. Adapaun rancangan dari antarmuka yang dibuat ditunjukkan pada Gambar 4.15



Gambar 4.15 Rancangan halaman Input

Keterangan:

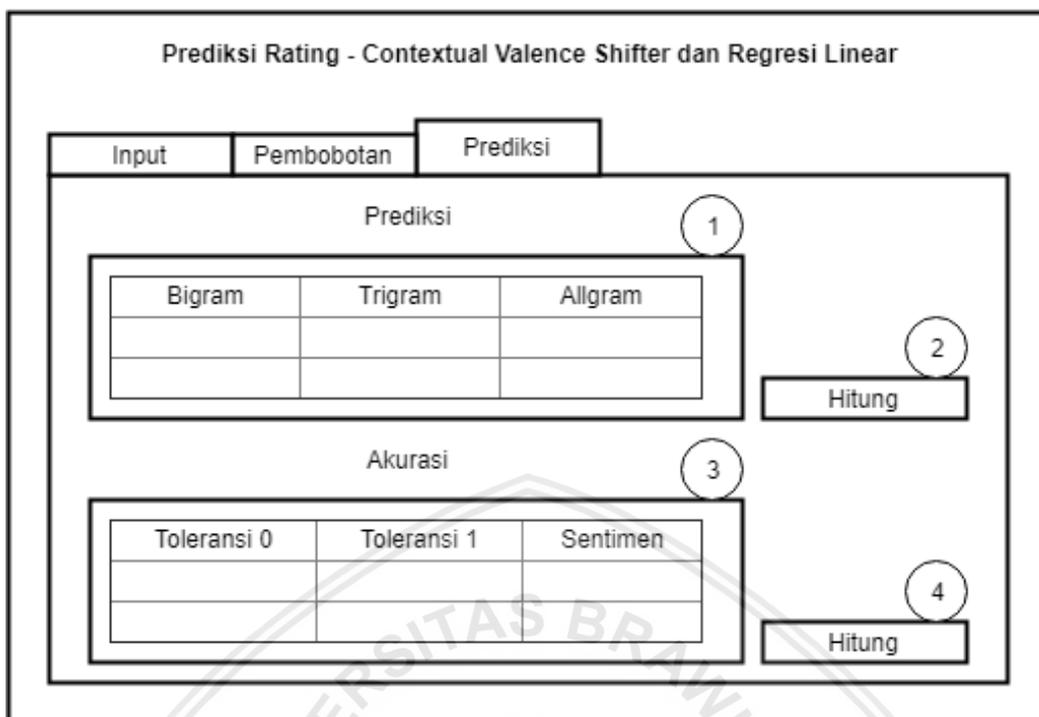
1. Judul
2. Tabel data *training*
3. Tombol input data *training*
4. Tabel data *testing*
5. Tombol input data *testing*



**Gambar 4.16 Rancangan halaman Pembobotan**

Keterangan:

1. Tabel Pembobotan data *training*
2. Tombol hitung pembobotan data *training*
3. Tabel Pembobotan data *testing*
4. Tombol Hitung pembobotan data *testing*



**Gambar 4.17 Rancangan halaman Prediksi**

Keterangan:

1. Tabel hasil Prediksi
2. Tombol hitung Prediksi
3. Table Akurasi
4. Tombol Hitung Akurasi

## BAB 5 IMPLEMENTASI

Bab ini berisi implementasi dari sistem yang telah dirancang meliputi fungsi, batasan-batasan yang digunakan untuk membangun sistem dan kode fungsi serta penjelasannya.

### 5.1 Batasan Implementasi

Sub bab ini berisi batasan-batasan pada sistem yang dibangun dan harus dipenuhi pada proses implementasi sistem. Tujuan dari batasan implementasi adalah agar sistem yang dibangun tidak melebar dari tujuan sistem, terdapat beberapa batasan yang ada pada sistem yaitu.

1. Implementasi sistem menggunakan bahasa pemrograman Java.
2. Metode yang digunakan untuk menyelesaikan masalah adalah *Contextual Valence Shifters* dan Regresi Linear.
3. *N-gram* yang digunakan yaitu *bigram*, *trigram* dan keseluruhan kalimat *review*.
4. Data yang digunakan sebagai *data training* dan *data testing* merupakan data berupa *review* dan *rating* yang di peroleh dari femaledialy.com
5. Keluaran pada sistem prediksi *rating review* adalah *Rating 1*, *Rating 2*, *Rating 3*, *Rating 4* dan *Rating 5*.

### 5.2 Implementasi Sistem

Pada sub bab ini berisi implementasi dari sistem yang dibuat serta mengacu pada perancangan sistem yang ada pada bab sebelumnya dan juga tahapan dari sistem yaitu *pre-processing* meliputi *casefolding* dan *tokenizing*, *n-gram*, *Lexicon Based features*, proses *Contextual Valence Shifters* dan metode prediksi regresi linear. Proses tersebut diurakan pada fungsi-fungsi kode program. Fungsi-fungsi tersebut lebih detailnya terdapat pada tabel 5.1

**Tabel 5.1 Daftar fungsi yang ada pada sistem**

NO	Proses	Nama Fungsi	Keterangan
1	Import kamus sentistrength_id	readBoost()	Fungsi ini digunakan untuk mengimport kata-kata booster (pendorong) untuk pembobotan kata
		readNegating()	Fungsi ini digunakan untuk mengimport kata-kata negative untuk pembobotan kata
		readSenti()	Fungsi ini digunakan untuk mengimport kata-kata yang mengandung berat.

2	Proses Training dan Testing	dataTraining()	Fungsi ini digunakan untuk masuk ke dalam proses <i>training</i> data.
		dataTesting()	Fungsi ini digunakan untuk masuk ke dalam proses <i>testing</i> data.
		inputData()	Fungsi ini digunakan untuk mengimport <i>review</i> dari excel ke dalam bentuk array baik data <i>training</i> maupun data <i>testing</i>
		Casefolding()	Fungsi ini digunakan untuk mengubah yang awalnya huruf besar ke huruf kecil, menghilangkan karakter selain alphabet dan angka.
		Tokenisasi()	Berfungsi untuk memecah kata berdasarkan spasi
3	Proses Pembobotan	lexicon()	Fungsi ini digunakan untuk pembobotan kata berdasarkan <i>lexicon</i> <i>sentistrength_id</i>
		countBigram()	Fungsi ini digunakan untuk pembobotan <i>Contextual Valence Shifters</i> berdasarkan menggunakan pengambilan kata <i>bigram</i>
		countTrigram()	Fungsi ini digunakan untuk pembobotan <i>Contextual Valence Shifters</i> berdasarkan menggunakan pengambilan kata <i>trigram</i>
		countAllgram()	Fungsi ini digunakan untuk pembobotan <i>Contextual Valence Shifters</i> berdasarkan menggunakan pengambilan kata kalimat <i>review</i>
4	Proses Prediksi <i>rating</i>	regresiLinear()	Fungsi ini digunakan untuk masuk ke dalam proses regresi linear.
		RegresiBigram()	Fungsi ini digunakan untuk mencari persamaan regresi dari Bigram
		RegresiTrigram()	Fungsi ini digunakan untuk mencari persamaan regresi dari Trigram

		RegresiAllgram()	Fungsi ini digunakan untuk mencari persamaan regresi dari kalimat <i>review</i>
		<i>ratingPrediction()</i>	Fungsi ini digunakan untuk memprediksi <i>rating</i> dari <i>review</i> data <i>testing</i> baik bigram, trigram maupun kalimat <i>review</i>
5	Penghitungan Akurasi	akurasi()	Fungsi ini digunakan untuk menghitung akurasi dari sistem yang dibuat, akurasinya sendiri menggunakan toleransi 0 dan toleransi 1

### 5.3 Pre-processing

*Pre-processing* merupakan tahap awal yang ada pada sistem ini. Pada tahap ini terdapat sub proses yaitu *Case Folding* dan *Tokenizing*. Kedua proses tersebut dibutuhkan untuk proses selanjutnya dari sistem yang dibuat.

#### 5.3.1 Case Folding

Tahap *Case Folding* adalah tahap yang digunakan untuk menghilangkan huruf yang dianggap *delimiter* dan mengubah huruf besar menjadi huruf kecil (*lowercase*), untuk kode program ditunjukkan pada Kode Program 5.1

##### Kode Program 5.1 Case Folding

```

1 public static String[][] Casefolding(String[][] data) throws
2 Exception {
3 String kalimathasil;
4 //proses casefolding
5 for (int i = 0; i < row; i++) {
6 String kalimatAwal = data[i][1];
7 //menghilangkan angka
8 kalimathasil = kalimatAwal.replaceAll("[0-9]", "");
9 kalimathasil = kalimathasil.replaceAll("[^a-zA-Z]", " ");
10 //mengubah ke huruf kecil
11 kalimathasil = kalimathasil.toLowerCase();
12 //menyimpan hasil
13 data[i][1] = kalimathasil;
14 }
15 //mengembalikan data
16 return data;
17 }

```

Penjelasan kode program dari implementasi *case folding* diuraikan sebagai berikut:

1. Baris 1 berupa nama method dengan parameter array 2 dimensi bertipe data *string*

2. Baris 3 deklarasi variable *string*
3. Baris 5 berupa perulangan for dengan maksimal perulangan kurang dari *row*, *row* sendiri merupakan jumlah baris dari array 2 dimensi data
4. Baris 6 adalah deklarasi dan inialisasi variabel string. Baris 8-9 dalah untuk mengilangkan kata yang dianggap *delimiter*. Baris 11 untuk mengubah huruf menjadi *lowercase*. Baris 11 adalah untuk menyimpan data kembali dalam array 2 dimensi
5. Baris 16 adalah untuk mengembalikan nilai dari hasil proses *method Case Folding*.

### 5.3.2 Tokenizing

Proses *tokenizing* adalah proses yang digunakan untuk mengubah suatu kalimat menjadi potongan kata. Untuk prosesnya dapat dilihat pada Kode Program 5.2

**Kode Program 5.2 Tokenizing**

```

1 public static String[] Tokenisasi(String data) throws
2 Exception {
3     //melakukan split untuk tokenisasi
4     String token[] = data.split(" ");
5     return token;
6 }

```

Penjelasan kode program dari implementasi *Tokenizing* di uraikan sebagai berikut:

1. Baris 1 merupakan deklarasi *method* tokenisasi dengan parameter *String* data
2. Baris 4 merupakan deklarasi *array* bertipe data *string* dan langsung di inialisasi dengan hasil dari *data.split()*
3. Baris 5 adalah untuk mengembalikan nilai dari hasil proses *method* tokenisasi.

### 5.4 Pembobotan dengan *Lexicon Based features*

Pembobotan *Lexicon Based features* merupakan salah satu metode yang digunakan untuk pembobotan suatu kata. Pembobotan ini memerlukan suatu kamus *lexicon*, *lexicon* yang digunakan dalam penelitian ini adalah *sentistrength\_id*. Berikut adalah kode programnya.

**Kode Program 5.3 Pembobotan Lexicon**

```

1 public static int Lexicon(String[] data, int tipe) throws
2 Exception {
3     //melakukan n-gram dan langsung diberi bobot
4     int weight = 0;
5     String[][] bufferdata = new String[data.length][3];
6     boolean mod = false;
7     for (int i = 0; i < data.length; i++) {
8         bufferdata[i][0] = data[i];
9     }
10    for (int i = 0; i < rowNegating; i++) {

```

```

11         for (int j = 0; j < bufferdata.length; j++) {
12             String kata = bufferdata[j][0];
13             String kataNegating = negating[i][0];
14             if (kata.equalsIgnoreCase(kataNegating)) {
15                 bufferdata[j][2] = "negative";
16                 mod = true;
17             }
18         }
19     }
20     for (int i = 0; i < rowSenti; i++) {
21         for (int j = 0; j < bufferdata.length; j++) {
22             String kata = bufferdata[j][0];
23             String kataSenti = senti[i][0];
24             if (kata.equalsIgnoreCase(kataSenti)) {
25                 bufferdata[j][1] = senti[i][1];
26                 bufferdata[j][2] = "senti";
27             }
28         }
29     }
30     for (int i = 0; i < bufferdata.length; i++) {
31         if (bufferdata[i][1] == null) {
32             bufferdata[i][1] = "0";
33         }
34     }
35     for (int i = 0; i < rowBoost; i++) {
36         for (int j = 0; j < bufferdata.length; j++) {
37             String kata = bufferdata[j][0];
38             String kataBoost = boost[i][0];
39             if (kata.equalsIgnoreCase(kataBoost)) {
40                 bufferdata[j][1] = boost[i][1];
41                 bufferdata[j][2] = "boost";
42             }
43         }
44     }
45     if (tipe == 1) {
46         //melakukan pembobotan secara bigram
47         weight = countBigram(bufferdata);
48     } else if (tipe == 2) {
49         //melakukan pembobotan secara trigram
50         weight = countTrigram(bufferdata);
51     } else if (tipe == 3) {
52         //melakukan pembobotan secara trigram
53         weight = allgram(bufferdata, mod);
54     }
55     return weight;
56 }

```

Penjelasan kode program dari implementasi Pembobotan *lexicon* *Sentistrength\_id* di uraikan sebagai berikut:

1. Baris 1 deklarasi method *Lexicon()* dengan parameter *array string* data dan integer tipe
2. Baris 4-6 deklarasi dan inisialisasi variabel *integer weight*, array 2 dimensi *bufferdata* dan *boolean* mod.
3. Baris 7-9 menyalin data yang ada pada *array* data ke dalam *bufferdata*

4. Baris 10-19 memberikan pembobotan pada *bufferdata* apakah didalam nya terdapat kata yang ada pada *lexicon negatingwords*.
5. Baris 20-29 memberikan pembobotan perkata pada *bufferdata* jika di dalamnya terdapat kata yang sama dengan *lexicon sentiwords\_id*
6. Baris 30-34 memeberikan nilai 0 apabila didalam *bufferdata* bernilai *null*
7. Baris 35-44 memberikan pembobotan *bufferdata* jika didalamnya terdapat kata yang sama dengan *lexicon boosterwords\_id*
8. Baris 45-54 melakukan pembobotan berdasarkan *n-gram* dan *Contextual Valence Shifters*
9. Baris 55 mengembalikan nilai weight.

### 5.5 N-gram dan Contextual Valence Shifters

Pada bagian ini dokumen hasil dari *pre-processing* di proses dengan menggunakan *n-gram* dan *Contextual Valence Shifters*. *N-gram* adalah suatu pengambilan atau kombinasi kata dengan jumlah n untuk setiap perhitungannya dengan n yang bernilai 2 bernama *bigram*, n yang bernilai 3 bernama *trigram* dan juga kalimat *review*. Setelah itu kata kombinasi tersebut dihitung dengan metode *Contextual Valence Shifters* menggunakan *lexicon* berbahasa Indonesia yang bernama *sentistrength\_id*.

#### 5.4.1 Bigram

*Bigram* adalah pengambilan kata atau kombinasi kata dengan jumlah kata yang digunakan adalah 2. Untuk kode program ditunjukkan pada Kode Program 5.4.

Kode Program 5.4 Bigram

```

1 public static int countBigram(String[][] bufferdata) throws
2 Exception {
3     int weight = 0;
4     //proses Contextual Valence Shifters
5     for (int i = 0; i < (bufferdata.length - 1); i++) {
6         //negative shifter context
7         if (bufferdata[i][2] == "negative" &&
8     bufferdata[i + 1][2] != "negative" &&
9     Integer.parseInt(bufferdata[i + 1][1]) > 0) {
10            int tempweight = -1 *
11    Integer.parseInt(bufferdata[i + 1][1]);
12            weight += tempweight;
13        } else if (bufferdata[i][2] != "negative" &&
14    bufferdata[i + 1][2] == "negative" &&
15    Integer.parseInt(bufferdata[i][1]) > 0) {
16            int tempweight = -1 *
17    Integer.parseInt(bufferdata[i + 1][1]);
18            weight += tempweight;
19        } else if (bufferdata[i][2] == "senti" &&
20    bufferdata[i + 1][2] == "boost") {
21            //jika weight (senti) lebih kecil dari 0 dan
22    booster lebih besar dari 0
23

```

```

24         int          array1          =
25 Integer.parseInt(bufferdata[i][1]);
26         int array2 = Integer.parseInt(bufferdata[i +
27 1][1]);
28         if (array1 < 0 && array2 > 0) {
29             int          berat          =
30 Integer.parseInt(bufferdata[i][1]) + (-1);
31             weight += berat;
32         } else {
33             int          berat          =
34 Integer.parseInt(bufferdata[i][1])
35 Integer.parseInt(bufferdata[i + 1][1]);
36             weight += berat;
37         }
38     } else if (bufferdata[i][2] == "boost" &&
39 bufferdata[i + 1][2] == "senti") {
40         int          array1          =
41 Integer.parseInt(bufferdata[i][1]);
42         int array2 = Integer.parseInt(bufferdata[i +
43 1][1]);
44         if (array1 > 0 && array2 < 0) {
45             int          berat          =
46 Integer.parseInt(bufferdata[i + 1][1]) + (-1);
47             weight += berat;
48         } else {
49             int          berat          =
50 Integer.parseInt(bufferdata[i][1])
51 Integer.parseInt(bufferdata[i + 1][1]);
52             weight += berat;
53         }
54     } else {
55         int          tempweight      =
56 Integer.parseInt(bufferdata[i][1])
57 Integer.parseInt(bufferdata[i + 1][1]);
58         weight += tempweight;
59     }
60 }
61 return weight;
62 }

```

Penjelasan Kode Program 5.4 dari implementasi *bigram* dan *Contextual Valence Shifters* diuraikan sebagai berikut:

1. Baris 1 deklarasi *method* *countBigram* dengan parameter *array* 2 dimensi.
2. Baris 3 deklarasi variabel *weight*
3. Baris 5-59 merupakan perhitungan *Contextual Valence Shifters* dengan pengambilan 2 kata (*bigram*).
4. Baris 60 pengembalian nilai *weight*.

#### 5.4.2 Trigram

*Trigram* adalah pengambilan kata atau kombinasi kata dengan jumlah kata yang digunakan adalah 3. Untuk kode program ditunjukkan pada Kode Program 5.5.

## Kode Program 5.5 Trigram

```

1 public static int countTrigram(String[][] bufferdata) throws
2 Exception {
3     int weight = 0;
4     for (int i = 0; i < (bufferdata.length - 2); i++) {
5         int array1 = Integer.parseInt(bufferdata[i][1]);
6         int array2 = Integer.parseInt(bufferdata[i +
7 1][1]);
8         int array3 = Integer.parseInt(bufferdata[i +
9 2][1]);
10        //negative shifter context trigram
11        if (bufferdata[i][2] == "negative" &&
12 bufferdata[i + 1][2] != "negative" && bufferdata[i + 2][2]
13 != "negative" && Integer.parseInt(bufferdata[i + 1][1]) > 0
14 && Integer.parseInt(bufferdata[i + 2][1]) > 0) {
15            int tempweight = -1 *
16 (Integer.parseInt(bufferdata[i + 1][1]) +
17 Integer.parseInt(bufferdata[i + 2][1]));
18            weight += tempweight;
19        } else if (bufferdata[i][2] != "negative" &&
20 bufferdata[i + 1][2] == "negative" && bufferdata[i + 2][2]
21 != "negative" && Integer.parseInt(bufferdata[i][1]) > 0 &&
22 Integer.parseInt(bufferdata[i + 2][1]) > 0) {
23            int tempweight = -1 *
24 (Integer.parseInt(bufferdata[i][1]) +
25 Integer.parseInt(bufferdata[i + 2][1]));
26            weight += tempweight;
27        } else if (bufferdata[i][2] != "negative" &&
28 bufferdata[i + 1][2] != "negative" && bufferdata[i + 2][2]
29 == "negative" && Integer.parseInt(bufferdata[i][1]) > 0 &&
30 Integer.parseInt(bufferdata[i + 1][1]) > 0) {
31            int tempweight = -1 *
32 (Integer.parseInt(bufferdata[i][1]) +
33 Integer.parseInt(bufferdata[i + 1][1]));
34            weight += tempweight;
35        } else if (bufferdata[i][2] == "negative" &&
36 bufferdata[i + 1][2] == "negative" && bufferdata[i + 2][2]
37 != "negative" && Integer.parseInt(bufferdata[i + 2][1]) > 0)
38 {
39            int tempweight = -1 *
40 Integer.parseInt(bufferdata[i + 2][1]);
41            weight += tempweight;
42        } else if (bufferdata[i][2] == "negative" &&
43 bufferdata[i + 1][2] != "negative" && bufferdata[i + 2][2]
44 == "negative" && Integer.parseInt(bufferdata[i + 1][1]) > 0)
45 {
46            int tempweight = -1 *
47 Integer.parseInt(bufferdata[i + 1][1]);
48            weight += tempweight;
49        } else if (bufferdata[i][2] != "negative" &&
50 bufferdata[i + 1][2] == "negative" && bufferdata[i + 2][2]
51 == "negative" && Integer.parseInt(bufferdata[i][1]) > 0) {
52            int tempweight = -1 *
53 Integer.parseInt(bufferdata[i][1]);
54            weight += tempweight;
55        } else if (bufferdata[i][2] == "senti" &&
56 bufferdata[i + 1][2] == "boost" && bufferdata[i + 2][2] ==
57 "boost") {

```

```
58         if (array1 < 0 && array2 > 0 && array3 > 0)
59     {
60         int berat = array1 + (-2);
61         weight += berat;
62     } else {
63         int berat = array1 + array2 + array3;
64         weight += berat;
65     }
66     } else if (bufferdata[i][2] == "boost" &&
67 bufferdata[i + 1][2] == "senti" && bufferdata[i + 2][2] ==
68 "boost") {
69         if (array1 > 0 && array2 < 0 && array3 > 0)
70     {
71         int berat = array2 + (-2);
72         weight += berat;
73     } else {
74         int berat = array1 + array2 + array3;
75         weight += berat;
76     }
77     } else if (bufferdata[i][2] == "boost" &&
78 bufferdata[i + 1][2] == "boost" && bufferdata[i + 2][2] ==
79 "senti") {
80         if (array1 > 0 && array2 > 0 && array3 < 0)
81     {
82         int berat = array3 + (-2);
83         weight += berat;
84     } else {
85         int berat = array1 + array2 + array3;
86         weight += berat;
87     }
88     } else if (bufferdata[i][2] == "senti" &&
89 bufferdata[i + 1][2] == "senti" && bufferdata[i + 2][2] ==
90 "boost") {
91         if (array1 < 0 && array2 < 0 && array3 > 0)
92     {
93         int berat = array1 + array2 + (-1);
94         weight += berat;
95     } else {
96         int berat = array1 + array2 + array3;
97         weight += berat;
98     }
99     } else if (bufferdata[i][2] == "senti" &&
100 bufferdata[i + 1][2] == "boost" && bufferdata[i + 2][2] ==
101 "senti") {
102         if (array1 < 0 && array2 > 0 && array3 < 0)
103     {
104         int berat = array1 + array3 + (-1);
105         weight += berat;
106     } else {
107         int berat = array1 + array2 + array3;
108         weight += berat;
109     }
110     } else if (bufferdata[i][2] == "boost" &&
111 bufferdata[i + 1][2] == "senti" && bufferdata[i + 2][2] ==
112 "senti") {
113         if (array1 > 0 && array2 < 0 && array3 < 0)
114     {
115         int berat = array2 + array3 + (-1);
116         weight += berat;
```

```

117         } else {
118             int berat = array1 + array2 + array3;
119             weight += berat;
120         }
121     } else {
122         int berat = array1 + array2 + array3;
123         weight += berat;
124     }
125 }
126 return weight;
127 }
128
129

```

Penjelasan Kode Program 5.5 dari implementasi *trigram* dan *Contextual Valence Shifters* diuraikan sebagai berikut:

1. Baris 1 deklarasi *method countTrigram* dengan parameter *array* 2 dimensi.
2. Baris 3 deklarasi variabel *weight*
3. Baris 4-127 merupakan perhitungan *Contextual Valence Shifters* dengan pengambilan 3 kata (*trigram*).
4. Baris 128 pengembalian nilai *weight*.

#### 5.4.3 Kalimat Review

Kalimat *review* yang dimaksud adalah semua kata yang ada pada suatu *review*. Untuk kode program ditunjukkan pada Kode Program 5.6.

#### Kode Program 5.6 Kalimat *review*

```

1 public static int allgram(String[][] bufferdata , boolean
2 modifier) throws Exception {
3     //proses unigram dan pembobotan CVS
4     int weight = 0;
5
6     for (int i = 0; i < bufferdata.length; i++) {
7         int bobot = Integer.parseInt(bufferdata[i][1]);
8         weight += bobot;
9     }
10
11     if (modifier == true && weight > 0) {
12         weight = weight * -1;
13     }
14     return weight;
15 }

```

Penjelasan kode program dari implementasi kalimat *review* dan *Contextual Valence Shifters* di uraikan sebagai berikut:

1. Baris 1 deklarasi *method allgram* dengan parameter *array string* 2 dimensi dan Boolean
2. Baris 4 deklarasi variabel *weight*

3. Baris 6-13 perhitungan *Contextual Valence Shifters* dengan kalimat *review*
4. Baris 14 pengembalian nilai

## 5.6 Regresi Linear

Regresi linear adalah pendekatan linear dengan memodelkan hubungan antara variabel dependen dengan variabel independen (*explanatory variables*). Dalam penelitian ini dibagi menjadi dua proses regresi linear, yaitu regresi linear *data training* dan regresi linear *data testing*.

### 5.6.1 Regresi Linear *Data Training*

Regresi linear *data training* bertujuan untuk mendapatkan suatu nilai (model regresi linear) yang akan digunakan oleh regresi linear *data testing*. Berikut adalah implementasinya.

**Kode Program 5.7 Regresi Linear *Data Training***

	RegresiBigram()
1	public static void RegresiBigram(double data[][]) {
2	int totalData = data.length;
3	double sigmaRate = 0;
4	double sigmaWeight = 0;
5	double sigmaRate2 = 0;
6	double sigmaWeight2 = 0;
7	double sigmaRateWeight = 0;
8	for (int i = 0; i < totalData; i++) {
9	sigmaRate += data[i][3];
10	sigmaWeight += data[i][0];
11	sigmaRate2 += (data[i][3] * data[i][3]);
12	sigmaWeight2 += (data[i][0] * data[i][0]);
13	sigmaRateWeight += (data[i][3] * data[i][0]);
14	}
15	b_bigram = (((totalData * sigmaRateWeight) -
16	(sigmaWeight * sigmaRate)) /
17	((totalData * sigmaWeight2) - (sigmaWeight *
18	sigmaWeight));
19	
20	a_bigram = ((sigmaRate / totalData) - (b_bigram *
21	(sigmaWeight / totalData));
22	}
	RegresiTrigram()
1	public static void RegresiTrigram(double data[][]) {
2	int totalData = data.length;
3	double sigmaRate = 0;
4	double sigmaWeight = 0;
5	double sigmaRate2 = 0;
6	double sigmaWeight2 = 0;
7	double sigmaRateWeight = 0;
8	for (int i = 0; i < totalData; i++) {
9	sigmaRate += data[i][3];
10	sigmaWeight += data[i][1];
11	sigmaRate2 += (data[i][3] * data[i][3]);
12	sigmaWeight2 += (data[i][1] * data[i][1]);
13	sigmaRateWeight += (data[i][3] * data[i][1]);
14	}
15	

16	<code>b_trigram = (((totalData * sigmaRateWeight) -</code>
17	<code>(sigmaWeight * sigmaRate)) /</code>
18	<code>((totalData * sigmaWeight2) - (sigmaWeight *</code>
19	<code>sigmaWeight));</code>
20	
21	<code>a_trigram = ((sigmaRate / totalData) - (b_trigram *</code>
22	<code>(sigmaWeight / totalData));</code>
	<code>}</code>
	<code>RegresiAllgram()</code>
1	<code>public static void RegresiAllgram(double data[][]) {</code>
2	<code>int totalData = data.length;</code>
3	<code>double sigmaRate = 0;</code>
4	<code>double sigmaWeight = 0;</code>
5	<code>double sigmaRate2 = 0;</code>
6	<code>double sigmaWeight2 = 0;</code>
7	<code>double sigmaRateWeight = 0;</code>
8	<code>for (int i = 0; i &lt; totalData; i++) {</code>
9	<code>sigmaRate += data[i][3];</code>
10	<code>sigmaWeight += data[i][2];</code>
11	<code>sigmaRate2 += (data[i][3] * data[i][3]);</code>
12	<code>sigmaWeight2 += (data[i][2] * data[i][2]);</code>
13	<code>sigmaRateWeight += (data[i][3] * data[i][2]);</code>
14	<code>}</code>
15	<code>b_allgram = (((totalData * sigmaRateWeight) -</code>
16	<code>(sigmaWeight * sigmaRate)) /</code>
17	<code>((totalData * sigmaWeight2) - (sigmaWeight *</code>
18	<code>sigmaWeight));</code>
19	
20	<code>a_allgram = ((sigmaRate / totalData) - (b_allgram *</code>
21	<code>(sigmaWeight / totalData));</code>
22	<code>}</code>

Pada dasarnya ketiga kode di atas yaitu yaitu *regresiBigram*, *regresiTrigram* dan *regresiAllgram* sama, yang membedakan hanya kolom dalam array data, dimana kolom nol berisi data pembobotan *bigram*, kolom satu berisi data pembobotan *trigram*, dan kolom dua data pembobotan kalimat *review*.

Penjelasan kode program dari implementasi Regresi Linear *data training* di uraikan sebagai berikut:

1. Baris 1 deklarasi method regresi dengan parameter *array double*
2. Baris 2-7 deklarasi variabel untuk penghitungan regresi linear
3. Baris 8-14 perulangan untuk memberi nilai dari variabel yang pada tahap sebelumnya sudah dideklarasikan
4. Baris 15-18 merupakan perhitungan nilai B
5. Baris 20-22 merupakan perhitungan nilai A

### 5.6.2 Regresi Linear *Data Testing*

Regresi linear *data testing* bertujuan untuk mendapatkan nilai prediksi *rating review*. Regresi linear ini menggunakan model regresi linear yang telah diperoleh

dari regresi linear *data training*. Berikut adalah implementasinya ditunjukkan pada Kode Program 5.8.

### Kode Program 5.8 Regresi Linear *Data Testing*

```

1 public static void ratingPrediction(String data[][], String
2 dataTest[][]) {
3     // y = a+bx
4     double rate[][] = new double[data.length][3];
5     int rateFinal[][] = new int[rate.length][3];
6     for (int i = 0; i < rate.length; i++) {
7         rate[i][0] = (a_bigram + (b_bigram *
8 (Double.parseDouble(data[i][0]))));
9         rate[i][1] = (a_trigram + (b_trigram *
10 (Double.parseDouble(data[i][1]))));
11         rate[i][2] = (a_allgram + (b_allgram *
12 (Double.parseDouble(data[i][2]))));
13     }
14     for (int i = 0; i < rate.length; i++) {
15         for (int j = 0; j < 3; j++) {
16             int a = (int) Math.round(rate[i][j]);
17             rateFinal[i][j] = a;
18         }
19     }
20 }

```

Penjelasan kode program dari implementasi Regresi Linear *data testing* diuraikan sebagai berikut:

1. Baris 1 deklarasi method *ratingPrediction* dengan parameter *array String*.
2. Baris 4-5 deklarasi array bertipe data *double* dan *integer*.
3. Baris 6-13 perhitungan rating, baik pada *bigram*, *trigram* dan kalimat *review*.
4. Baris 14-19 membulatkan hasil dari prediksi rating pada proses sebelumnya.

## 5.7 Akurasi

Akurasi merupakan suatu tolak ukur hasil dari suatu sistem yang dibuat. Berikut adalah implementasinya.

### Kode Program 5.9 Akurasi

```

1 public static void akurasi(int rateFinal[][], String
2 dataTest[][]) {
3     System.out.println("=====" + "Prediksi Rating Data
4 Test =====");
5     for (int i = 0; i < rateFinal.length; i++) {
6         System.out.print((i+1)+". ");
7         for (int j = 0; j < 3; j++) {
8             System.out.print(rateFinal[i][j] + "=");
9         }
10        System.out.println("");
11    }
12    int AccBigram = 0;
13    int AccTrigram = 0;

```

```

14     int AccAllgram = 0;
15     for (int i = 0; i < rateFinal.length; i++) {
16         int A = Integer.parseInt(dataTest[i][2]);
17         if (rateFinal[i][0] == A) {
18             AccBigram += 1;
19         }
20         if (rateFinal[i][1] == A) {
21             AccTrigram += 1;
22         }
23         if (rateFinal[i][2] == A) {
24             AccAllgram += 1;
25         }
26     }
27     System.out.println("=====" + "Akurasi Toleransi 0
28 =====");
29     System.out.println("Akurasi Bigram : " + AccBigram);
30     System.out.println("Akurasi Trigram : " +
31 AccTrigram);
32     System.out.println("Akurasi Allgram : " +
33 AccAllgram);
34
35     //Akurasi
36     int AccBigram1 = 0;
37     int AccTrigram1 = 0;
38     int AccAllgram1 = 0;
39     for (int i = 0; i < rateFinal.length; i++) {
40         int A = Integer.parseInt(dataTest[i][2]);
41         if (rateFinal[i][0] == A || (rateFinal[i][0] -
42 1) == A || (rateFinal[i][0] + 1) == A) {
43             AccBigram1 += 1;
44         }
45         if (rateFinal[i][1] == A || (rateFinal[i][1] -
46 1) == A || (rateFinal[i][1] + 1) == A) {
47             AccTrigram1 += 1;
48         }
49         if (rateFinal[i][2] == A || (rateFinal[i][2] -
50 1) == A || (rateFinal[i][2] + 1) == A) {
51             AccAllgram1 += 1;
52         }
53     }
54     System.out.println("=====" + "Akurasi Toleransi 1
55 =====");
56     System.out.println("Akurasi Bigram : " + AccBigram1);
57
58     System.out.println("Akurasi Trigram : " +
59 AccTrigram1);
60     System.out.println("Akurasi Allgram : " +
61 AccAllgram1);
62     int sentBi = 0;
63     int sentTri = 0;
64     int sentAll = 0;
65     for (int i = 0; i < rateFinal.length; i++) {
66         // A adalah rating
67         int A = Integer.parseInt(dataTest[i][2]);
68         if (A <= 3 && rateFinal[i][0] <= 3) {
69             sentBi += 1;
70         } else if (A > 3 && rateFinal[i][0] > 3) {
71             sentBi += 1;
72         }

```

```

73         if (A <= 3 && rateFinal[i][1] <= 3) {
74             sentTri += 1;
75         }else if(A > 3 && rateFinal[i][1] > 3){
76             sentTri += 1;
77         }
78         if (A <= 3 && rateFinal[i][2] <= 3) {
79             sentAll += 1;
80         }else if(A > 3 && rateFinal[i][2] > 3){
81             sentAll += 1;
82         }
83     }
84     System.out.println("====="           Sentimen  Ulasan
85     =====");
86     System.out.println("Sentimen Bigram : " + sentBi);
87     System.out.println("Sentimen Trigram : " + sentTri);
88     System.out.println("Sentimen Allgram : " + sentAll);
89 }

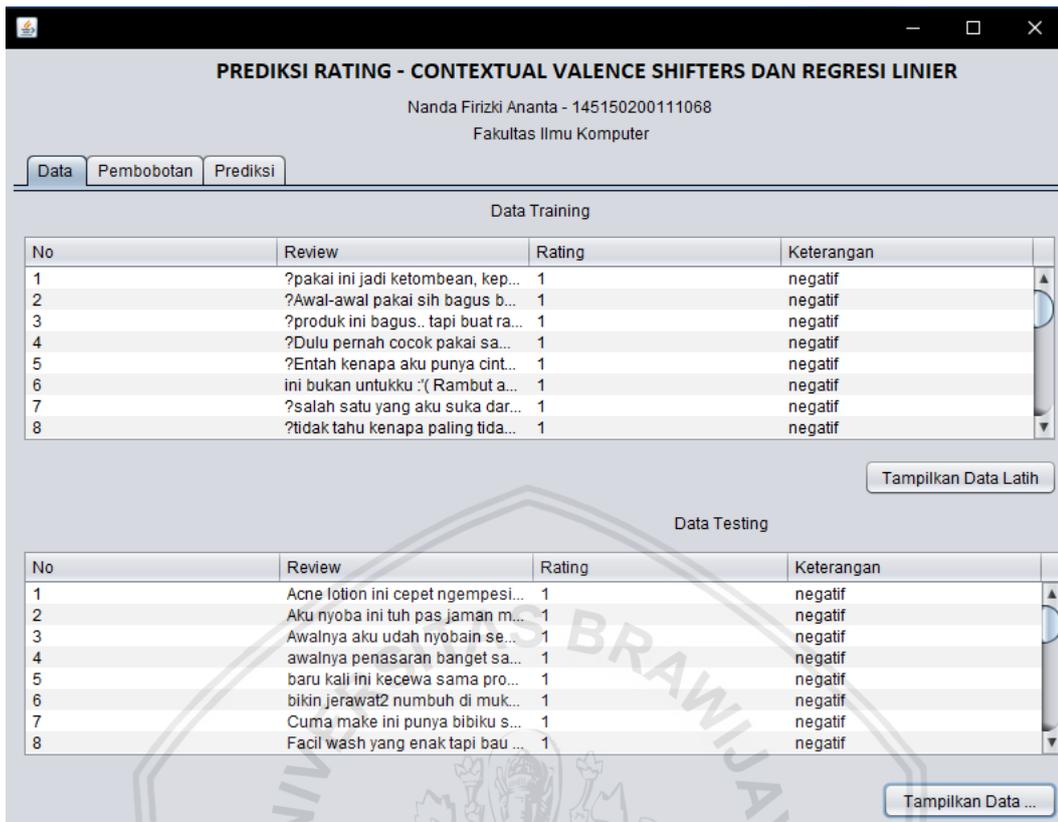
```

Penjelasan kode program dari implementasi akurasi diuraikan sebagai berikut:

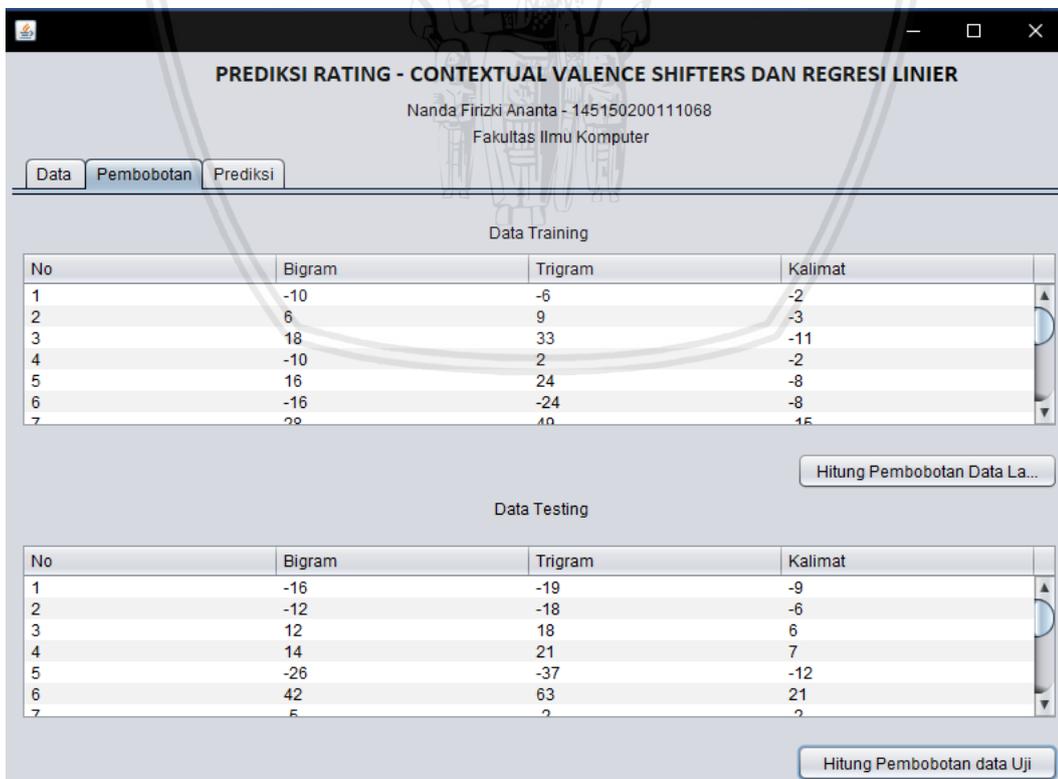
1. Baris 1 Deklarasi method akurasi dengan parameter array integer 2 dimensi dan array String 2 dimensi
2. Baris 5-11 menampilkan hasil dari prediksi *rating*
3. Baris 12-14 deklarasi dan inialisasi variabel *int*
4. Baris 15-26 perhitungan akurasi pada toleransi 0
5. Baris 27-33 menampilkan hasil dari akurasi pada proses sebelumnya
6. Baris 36-38 deklarasi dan inialisasi variabel *int*
7. Baris 39-53 perhitungan akurasi pada toleransi 1
8. Baris 54-61 menampilkan hasil dari akurasi pada proses sebelumnya
9. Baris 62-64 deklarasi dan inialisasi variabel *int*
10. Baris 65-83 perhitungan akurasi sentimen
11. Baris 84-88 menampilkan hasil dari akurasi pada proses sebelumnya

## 5.8 Implementasi Antarmuka

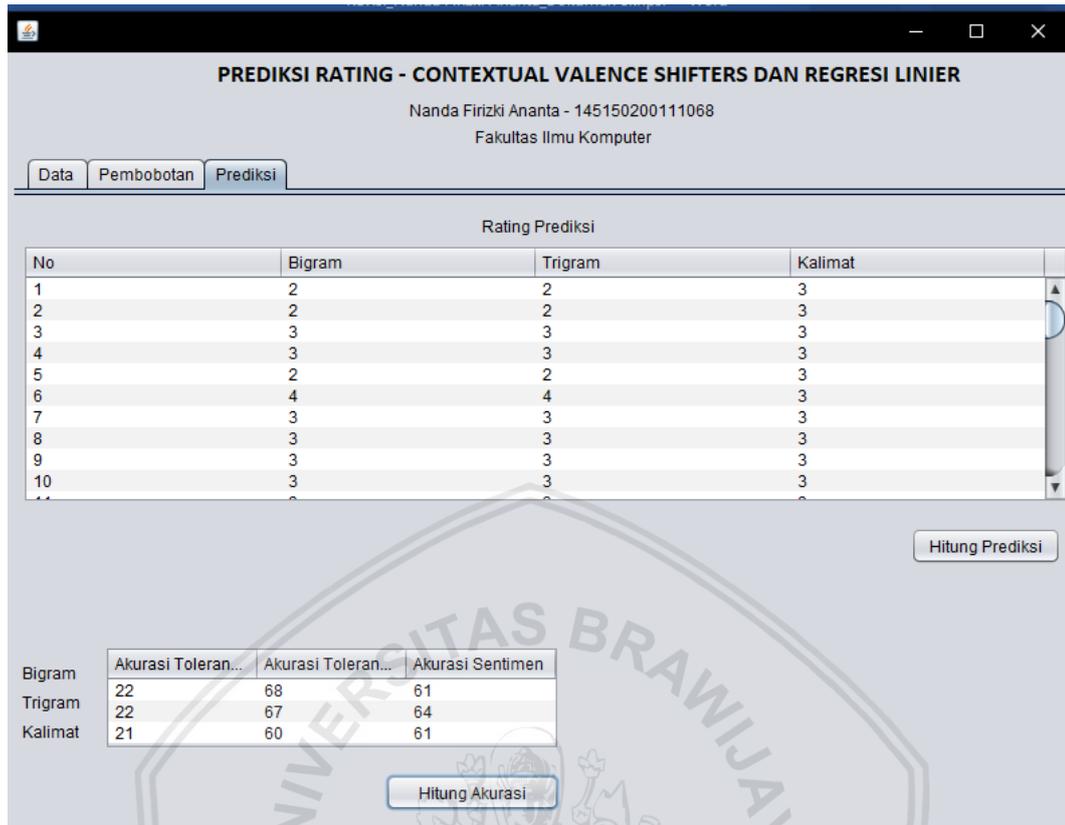
Antarmuka aplikasi digunakan oleh pengguna untuk berinteraksi dengan sistem. Aplikasi berbasis java dekstop dapat menangani operasi logika yang kompleks. Terdapat tiga menu yaitu yang pertama menu data yang digunakan untuk input data *training* dan data *testing*, antarmuka menu dapat dilihat pada Gambar 5.1. Kedua adalah menu pembobotan yang digunakan untuk pembobotan dengan metode *Contextual Valence Shifters*, antarmuka pembobotan dapat dilihat pada Gambar 5.2. Dan yang ketiga adalah prediksi yang digunakan untuk prediksi rating dengan metode Regresi Linear dan akurasi dari sistem, antarmuka menu prediksi dapat dilihat pada Gambar 5.3.



Gambar 5.1 Antarmuka Sistem Prediksi Rating (Menu Data)



Gambar 5.2 Antarmuka Sistem Prediksi Rating (menu Pembobotan)



Gambar 5.3 Antarmuka Sistem Prediksi Rating (menu Prediksi)

## BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang pengujian terhadap sistem yang dibuat. Selain itu, pada bab ini juga menjelaskan analisis terhadap hasil dari implementasi yang telah dilakukan.

### 6.1 Skenario Pengujian

Pengujian dilakukan dengan skenario metode *k-fold cross validation* dengan  $k$  sebanyak 10 terhadap pengambilan kata (*n-gram*) dengan model pengujian Toleransi 0 yaitu suatu hasil prediksi *rating* harus sama dengan *rating data testing*, pengujian Toleransi 1 merupakan pengujian dengan hasil prediksi boleh memiliki *rating* yang berbeda namun maksimal perbedaan dari hasil *rating* terhadap *rating* asli data uji hanya 1 dan yang terakhir adalah model pengujian *sentiment* yaitu pengujian dengan informasi sentimen yang ada pada *review* melalui *rating* yang ada.

Tujuan dari pengujian dengan skenario dan model pengujian di atas adalah untuk mengetahui pengaruh *n-gram* pada prediksi *rating*. Data yang digunakan dalam penelitian sebanyak 1000 data dengan pembagian 100 *data testing* dan 900 *data training* yang terdiri dari *review* beserta *rating*.

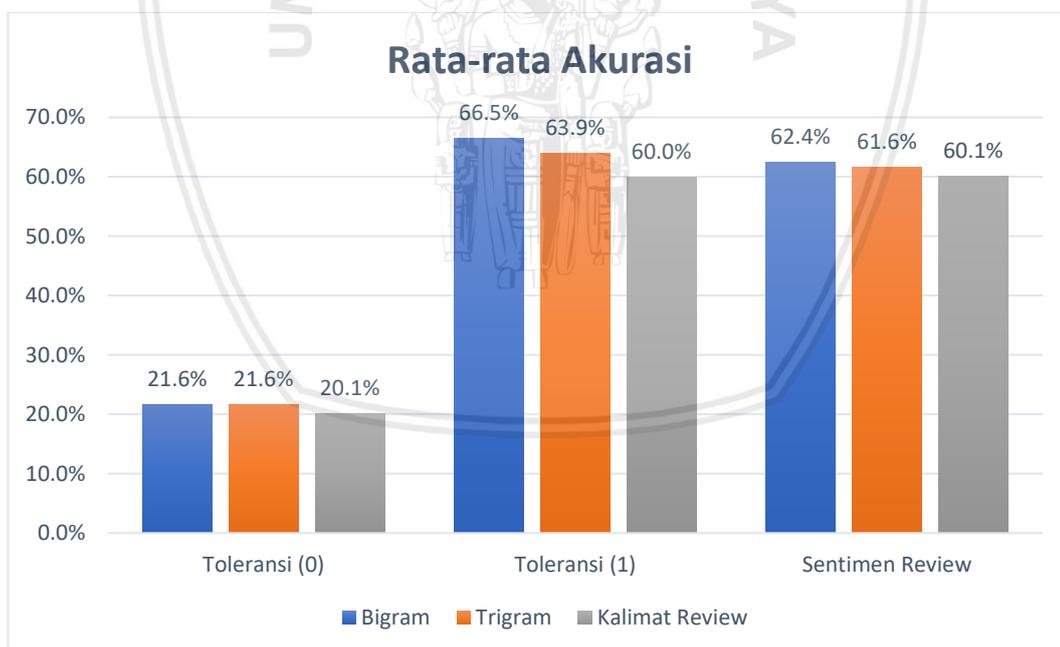
**Tabel 6.1 Hasil pengujian Contextual Valence Shifter dan N-gram**

Fold	N-gram	Model Pengujian		
		Toleransi (0)	Toleransi (1)	Sentimen review
1	Bigram	20%	68%	62%
	Trigram	22%	63%	61%
	Kalimat review	20%	60%	60%
2	Bigram	19%	64%	58%
	Trigram	22%	63%	61%
	Kalimat review	20%	60%	60%
3	Bigram	23%	69%	65%
	Trigram	22%	67%	62%
	Kalimat review	20%	60%	60%
4	Bigram	25%	68%	65%
	Trigram	23%	65%	64%
	Kalimat review	20%	60%	60%
5	Bigram	21%	63%	61%
	Trigram	23%	60%	59%
	Kalimat review	20%	60%	60%
6	Bigram	23%	67%	61%
	Trigram	23%	66%	62%
	Kalimat review	20%	60%	60%
7	Bigram	24%	69%	67%

	Trigram	21%	64%	61%
	Kalimat review	20%	60%	60%
8	Bigram	18%	63%	60%
	Trigram	17%	62%	61%
	Kalimat review	20%	60%	60%
9	Bigram	21%	66%	64%
	Trigram	21%	62%	61%
	Kalimat review	20%	60%	60%
10	Bigram	22%	68%	61%
	Trigram	22%	67%	64%
	Kalimat review	21%	60%	61%
Rata-Rata	Bigram	21,6%	66,5%	62,4%
	Trigram	21,6%	63,9%	61,6%
	Kalimat review	20,1%	60,0%	60,1%

## 6.2 Evaluasi Hasil Pengujian

Pada sub bab ini dilakukan analisis dan penjelasan lebih rinci dari Tabel 6.1, lebih jelasnya dapat dilihat pada Gambar 6.1.



**Gambar 6.1** Grafik Hasil akurasi Pengujian Prediksi *Rating* dengan Metode *Contextual Valence Shifters* dan Regresi Linear

Gambar 6.1 adalah hasil dari pengujian *n-gram* terhadap beberapa model pengujian. Dari hasil pengujian tersebut menunjukkan bahwa hasil dari prediksi dengan metode *Contextual Valence Shifters* dan Regresi Linear ini memiliki akurasi

yang relatif rendah. Hal tersebut dapat disebabkan oleh beberapa hal, antara lain adalah pada *review* yang digunakan baik untuk data *training* maupun data *testing* terdapat banyak kata yang tidak baku seperti singkatan dan kata-kata yang walaupun sebenarnya adalah kata yang bersifat sentimen namun karena penulisannya tidak baku atau salah penulisan maka kata tersebut tidak diberi bobot sehingga dapat mengurangi akurasi dari sistem, selain itu karakteristik setiap orang berbeda dalam melakukan *review*, terkadang *review* yang tergolong pada rating yang rendah para *reviewer* banyak menggunakan kata yang bersifat sentimen positif maupun sebaliknya, hal tersebut dapat berakibat akurasi sistem yang menurun, misalnya pada *review* “produk ini sebenarnya bagus, wanginya enak, harganya murah namun tidak cocok pada kulit saya” dengan rating 1, pada kalimat *review* tersebut terdapat beberapa kata yang memiliki bobot dilihat dari kamus *lexicon*, yaitu “bagus”, “enak”, “murah”, “tidak” dan “cocok” dimana kata “bagus”, “enak” dan “murah” adalah kata-kata yang memiliki bobot positif, sedangkan kata “tidak” adalah kata negasi yang berarti membuat kata setelahnya berubah menjadi kata negatif, jika semua kata yang memiliki bobot itu ditambahkan akan menghasilkan bobot yang positif namun memiliki rating yang rendah, oleh karena itu dapat menurunkan akurasi dari sistem saat sistem melakukan prediksi dengan metode Regresi Linear.

Hasil skenario pengujian dimodelkan menjadi tiga, yaitu Toleransi 0, Toleransi 1, dan sentimen *review*. Ketiga model tersebut bertujuan untuk mengetahui kinerja dari sistem yang telah dibuat secara mendalam. Model pengujian dengan Toleransi 0 adalah hasil dari proses prediksi rating harus sama dengan rating dari data uji. Hasil yang diperoleh tergolong rendah, dengan akurasi tertinggi hanya 21,6% data yang benar. Namun saat dilakukan pengujian dengan model Toleransi 1, akurasi yang didapatkan jauh lebih tinggi jika dibandingkan dengan Toleransi 0 dengan mendapatkan hasil akurasi tertinggi sebesar 66,5%. hal ini menunjukkan bahwa hasil prediksi akurasi banyak yang perbedaannya hanya 1 tingkat rating. Kesalahan prediksi disebabkan antara rating yang berdekatan seperti *rating* 2 dengan *rating* 3 atau *rating* 4 dengan *rating* 5. Model pengujian ketiga yaitu Sentimen *review* menunjukkan hasil yang hanya sedikit dibawah model Toleransi 1 dengan akurasi tertinggi sebesar 62,4%. Hal ini menunjukkan bahwa sistem yang dibangun juga memiliki kemampuan untuk melakukan analisis sentimen namun dengan kinerja yang tidak cukup baik ketika menggunakan kategori positif dan negatif.

### 6.2.1 Analisis Hasil Pengujian dengan *Bigram*

Hasil dari skenario pengujian dengan *Bigram* dengan tiga model variasi pengujian toleransi 0, toleransi 1, dan sentimen *review* mendapatkan hasil 21,6%, 66,5%, dan 62,4%. Hasil dari pengujian menunjukkan bahwa akurasi yang dihasilkan oleh *bigram* cenderung lebih baik dibandingkan dengan yang lainnya. Oleh karena itu dapat diambil kesimpulan bahwa metode *Contextual Valence Shifters* ini lebih baik saat menggunakan *bigram* dalam pengambilan katanya. Hal itu dikarenakan *reviewer* cenderung lebih sering menggunakan dua kata dalam

penyampaian ungkapannya seperti “barangnya bagus” atau “bagus sekali” dan lain-lain.

### 6.2.2 Analisis Hasil Pengujian dengan *Trigram*

Hasil dari skenario pengujian dengan *Trigram* dengan tiga model variasi pengujian Toleransi 0, Toleransi 1, dan sentimen *review* mendapatkan hasil 21,6%, 63,9%, dan 61,6%. Hasil yang diperoleh menunjukkan bahwa akurasi yang didapatkan lebih rendah jika dibandingkan dengan pengujian *Bigram* namun masih sedikit lebih tinggi jika dibandingkan dengan pengujian dengan Kalimat *review*. Hal ini juga dikarenakan *reviewer* terkadang cenderung sering menggunakan tiga ungkapan dalam penyampaiannya seperti “barangnya bagus sekali” atau “produknya kurang bagus” dan lain-lain.

### 6.2.3 Analisis Hasil Pengujian dengan Kalimat *review*

Hasil dari skenario pengujian dengan Kalimat *review* dengan tiga model variasi pengujian Toleransi 0, Toleransi 1, dan sentimen *review* mendapatkan hasil 20,1%, 60,0% dan 60,1%. Hasil yang diperoleh menunjukkan bahwa jika dibandingkan dengan dua pengujian sebelumnya maka pengujian ini memiliki tingkat akurasi yang paling rendah. Pengujian ini ada dikarenakan terkadang *reviewer* menggunakan kata ungkapan yang banyak dalam satu *review* seperti “barangnya bagus, harganya murah, namun tidak cocok pada kulit saya” dengan *rating* yang rendah, jika menggunakan *bigram* ataupun *trigram* bobot yang didapat oleh kalimat *review* tersebut akan bernilai positif sedangkan yang dimaksud *reviewer* adalah sebaliknya, oleh karena itu penggunaan kalimat *review* untuk pembobotannya dapat mengatasi masalah tersebut dikarenakan terdapat kata negasi “tidak” maka seluruh bobot akan ditambahkan terlebih dahulu lalu hasil tersebut akan diubah menjadi negatif.

## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan pengujian dari sistem Prediksi *Rating* pada *Review* Produk Kecantikan Menggunakan Metode *Contextual Valence Shifters* dan Regresi Linear dapat disimpulkan sebagai berikut:

1. Metode *Contextual Valence Shifters* dan Regresi Linear dapat digunakan untuk prediksi *rating*. Penelitian ini menggunakan data yang diperoleh dari *website* <https://femaledialy.com> sebanyak 1000 data dengan 900 data *training* dan 100 data *testing* dengan langkah-langkahnya adalah tahap proses *pre-processing* yaitu *case folding* dan *tokenizing*, hasil yang diperoleh dari proses *pre-processing* dilakukan proses pembobotan *Lexicon Based Features* dan *Contextual Valence Shifters*, lalu proses terakhir adalah proses prediksi dengan regresi linear.
2. Penggunaan n-gram lebih tepatnya *bigram* berpengaruh baik untuk hasil dari prediksi *rating* jika dibandingkan dengan *trigram* maupun kalimat. Akurasi yang diperoleh dari penggunaan *bigram* dengan Toleransi 0 adalah 21,6% atau sama dengan *trigram*, namun jika menggunakan Toleransi 1 hasil yang diperoleh *bigram* lebih baik dari pada *trigram* dengan akurasi 66,5% sedangkan *trigram* mendapatkan akurasi sebesar 63,9%.

### 7.2 Saran

Terdapat beberapa saran yang dapat dilakukan untuk penelitian selanjutnya dengan mengacu pada kekurangan-kekurangan dari hasil penelitian yang telah dilakukan adalah:

1. Membenahi kata yang salah ketik. Algoritme yang digunakan memang belum ada dalam Bahasa Indonesia untuk sekarang, namun dapat menggunakan kamus bernama "ID-Kamus-Typo" (Alfiana, 2017) yang dapat digunakan untuk perbandingan dan koreksi kata yang dianggap salah ketik.
2. Menambahkan suatu mekanisme yang dapat meningkatkan akurasi dari sistem pada penelitian ini seperti apabila ingin mendapatkan suatu prediksi *rating* dari pengguna A dan merek B, maka dapat mempertimbangkan dari segi *rating* dari produk merek B (*objective rating*) dan *rating* dari pengguna A (*subjective rating*).

## DAFTAR REFERENSI

- Alfiana, I. (2017, November 13). Retrieved from <https://github.com/ialfina/ID-Kamus-Typo#id-kamus-typo>
- Andini, T. I., Witanti, W., & Renaldi, F. (2016). Prediksi Potensi Pemasaran Produk Baru dengan Metode Naïve Bayes Classifier dan Regresi Linear.
- IBM. (2015). *IBM SPSS Modeler V17.1.0 documentation*. Retrieved from IBM Knowledge Center: [https://www.ibm.com/support/knowledgecenter/en/SS3RA7\\_17.1.0/ta\\_guide\\_ddita/textmining/shared\\_entities/tm\\_intro\\_tm\\_defined.html](https://www.ibm.com/support/knowledgecenter/en/SS3RA7_17.1.0/ta_guide_ddita/textmining/shared_entities/tm_intro_tm_defined.html)
- Lee, Y.-C. (2007). Application of support vector machines to corporate credit rating prediction.
- Mahmudy, W. F., & Widodo, A. W. (2014). Klasifikasi Artikel Berita Secara Otomatis Menggunakan Metode Naive Bayes Classifier Yang Dimodifikasi.
- Martin, P. (2012). *Dispel Tutorial 0.8 documentation*. Retrieved from [http://homepages.inf.ed.ac.uk/pmartin/tutorial/case\\_studies.html#k-fold-cross-validation](http://homepages.inf.ed.ac.uk/pmartin/tutorial/case_studies.html#k-fold-cross-validation).
- Ngoc, P. T., & Yoo, M. (2014). The Lexicon-based Sentiment Analysis for Fan Page Ranking in Facebook.
- Polanyi, L., & Zaenen, A. (2006). Contextual Valence Shifters.
- Ptaszynski, M., Dybala, P., Shi, W., Rzepka, R., & Araki, K. (2010). Contextual affect analysis: a system for verification of emotion appropriateness supported with Contextual Valence Shifters.
- Rofiqoh, U., Perdana, R. S., & Fauzi, M. A. (2017). Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexicon Based Features.
- Singh, P. K., Singh, S. K., & Paul, S. (2015). Sentiment Classification of Social Issues Using Contextual Valence Shifters.
- Syafruddin, M., Hakim, L., & Despa, D. (2014). Metode Regresi Linier untuk Prediksi Kebutuhan Energi Listrik Jangka Panjang (Studi Kasus Provinsi Lampung).
- Wahid, D. H., & Azhari, S. N. (2016). Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity.
- Zhang, H., Gan, W., & Jiang, B. (2014). Machine Learning and Lexicon based Methods for Sentiment Classification: A Survey.