PENENTUAN *RATING REVIEW* FILM MENGGUNAKAN METODE *MULTINOMIAL NAÏVE BAYES CLASSIFIER* DENGAN *FEATURE SELECTION* BERBASIS *CHI-SQUARE* DAN *GALAVOTTI-SEBASTIANI-SIMI COEFFICIENT*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Thio Marta Elisa Yuridis Butar Butar
NIM: 145150200111015

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

Penentuan Kategori *Rating Review* Film Menggunakan Metode *Multinomial Naïve Bayes Classifier* dengan *Feature Selection* Berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Thio Marta Elisa Yuridis Butar Butar
NIM: 145150200111015

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Mochammad Ali Fauzi, S.Kom., M.Kom.	Indriati, S.T., M.Kom.
NIK. 201502 890101 1 001	NIP. 19831013 201504 2 002

Mengetahui
Ketua Jurusan Teknik Informatika

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

PERNYATAAN ORISINALITAS

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 didalam 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, 2 Agustus 2018

Thio Marta Elisa Yuridis Butar Butar

NIM: 145150200111015

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul "**Penentuan *Rating Review* Film Menggunakan Metode *Multinomial Naïve Bayes Classifier* dengan *Feature Selection* Berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient***" ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Mochammad Ali Fauzi, S.Kom., M.Kom. dan Ibu lndriati, S.T., M.Kom. selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Bayu Priyambadha, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika,
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika,
4. Bapak Achmad Arwan, S.Kom., M.Kom. selaku dosen Penasihat Akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi,
5. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini,
6. Seluruh civitas academica Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 2 Agustus 2018

Penulis

Thaz070196@student.ub.ac.id

ABSTRAK

Thio Marta Elisa Yuridis Butar Butar, Penentuan *Rating Review* Film Menggunakan Metode *Multinomial Naïve Bayes Classifier* dengan *Feature Selection* Berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*

Pembimbing: Mochammad Ali Fauzi, S.Kom., M.Kom. dan Indriati, S.T, M.Kom.

Pada era saat ini terdapat berbagai ragam film, meskipun cara pendekatannya berbeda-beda, semua film dapat dikatakan mempunyai satu sasaran, yaitu menarik perhatian orang terhadap muatan-muatan masalah yang dikandung. Dari muatan-muatan film tersebut terdapat banyak respons dari penulis dan menuliskannya dalam *review* singkatnya. Dengan adanya *review* bisa membantu penonton untuk lebih selektif lagi dalam memilih suatu film. Dan dari pihak produksi bisa terbantu untuk mengukur seberapa jauh kualitas film yang mereka hasilkan. Namun dari pihak produksi sendiri terkadang mengalami kesulitan dalam memilah dan mengkategorikan *review*, apakah produk tersebut kualitasnya tergolong bagus, cukup bagus, tidak bagus, dan sebagainya. Dalam penelitian ini peniliaian suatu film berdasarkan *review* yang diberikan adalah *Rating*. Sehingga dibutuhkan sebuah sistem prediksi *Rating* untuk memprediksi dan menentukan *Rating* yang tepat berdasarkan *review* yang diberikan oleh pengguna terhadap suatu film. Untuk mendukung sistem yang dibangun dibutuhkan metode untuk menyelesaikan permasalahan tersebut, dalam penelitian ini peneliti menggunakan Metode *Multinomial Naïve Bayes* serta *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*. *Multinomial Naïve Bayes* adalah metode untuk klasifikasi sedangkan *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* adalah *feature selection* untuk lebih mengoptimalkan hasil dari klasifikasi. Dari hasil pengujian, didapat tingkat akurasi terbaik pada saat penggunaan *feature* sebesar 90%, dan 100% dengan tingkat akurasi sebesar 36%. Hasil tersebut adalah hasil terbaik dari hasil dengan prosentase penggunaan *feature* yang lain. Dari hasil tersebut *CHI-GSS* terbukti bisa melakukan pemilihan kata yang dianggap relevan maupun tidak relevan untuk dilakukan klasifikasi.

Kata kunci: prediksi *Rating*, *review* film, *Multinomial Naïve Bayes*, *Chi-Square* , *Galavotti-Sebastiani-Simi Coefficient*

ABSTRACT

Thio Marta Elisa Yuridis Butar Butar, Predicting Rating Movie Review Using Multinomial Naïve Bayes Classifier Method with Feature Selection Based on Chi-Square and Galavotti-Sebastiani-Simi Coefficient

Supervisors: Mochammad Ali Fauzi, S.Kom., M.Kom. dan Indriati, S.T, M.Kom.

In the current era there are various kinds of movies, although the way of approach varies, all movies can be said to have one goal, namely to attract people's attention to the contents of the problem. From the contents of the movie there are many responses from the author and write them in a short review. With review can help consumers to be more selective again in choosing a movie. And from the production side can be helped to measure how far the quality of the movies they produce. But from the production itself sometimes have difficulty in sorting and categorize the review, whether the movie is good quality, good enough, not good, and so forth. In this study the assessment of a moview based on the review given is Rating. So it takes a Rating prediction system to predict and determine the right Rating based on the reviews given by the users of a movie. To support the system built required methods to solve the problem, in this study researchers used the method of Multinomial Naïve Bayes along Chi-Square and Galavotti-Sebastiani-Simi Coefficient. Multinomial Naïve Bayes is a method for classification whereas Chi-Square and Galavotti-Sebastiani-Simi Coefficient is a feture selection to futher optimize the results of classification. From the test results, obtained the best accuracy level when the use features by 90%, and 100% with an accuracy of 36%. These results are the best results of the results with other features usage percentages. From these results CHI-GSS proven to make the selection of words that are considered relevant or irrelevant to do classification.

Keywords: Rating prediction, movie review, Multinomial Naïve Bayes, Chi-Square, Galavotti-Sebastiani-Simi Coefficient

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Menurut Effendy (1986: 134), film adalah media komunikasi yang bersifat audio visual untuk menyampaikan suatu pesan kepada sekelompok orang yang berkumpul di suatu tempat tertentu. Pesan film pada komunikasi massa dapat berbentuk apa saja tergantung dari misi film tersebut. Akan tetapi, umumnya sebuah film dapat mencakup berbagai pesan, baik itu pesan pendidikan, hiburan dan informasi. Pesan dalam film adalah menggunakan mekanisme lambang – lambang yang ada pada pikiran manusia berupa isi pesan, suara, perkataan, percakapan dan sebagainya.

Film juga dianggap sebagai media komunikasi yang ampuh terhadap massa yang menjadi sasarannya, karena sifatnya yang audio visual, yaitu gambar dan suara yang hidup. Dengan gambar dan suara, film mampu bercerita banyak dalam waktu singkat. Ketika menulis film penulis seakan-akan dapat menembus ruang dan waktu yang dapat menceritakan kehidupan dan bahkan dapat mempengaruhi audiens.

Oleh begitu pesatnya perkembangan teknologi dewasa ini banyak juga suatu situs konten film yang berisikan tentang bagaimana film tersebut dan *review* penulis yang pernah menyaksikan film tersebut. Sehingga dari *review* tersebut bisa merekomendasi pecinta film untuk menyaksikan film tertentu. Salah satu contoh situs web yang dimaksud ialah seperti pada https://montasefilm.com dan http://www.ulasanpilem.com.

Pada era saat ini terdapat berbagai ragam film, meskipun cara pendekatannya berbeda-beda, semua film dapat dikatakan mempunyai satu sasaran, yaitu menarik perhatian orang terhadap muatan-muatan masalah yang dikandung. Dari muatan-muatan film tersebut terdapat banyak respons dari penulis dan menuliskannya dalam *review* singkatnya. Dari *review* singkat tersebut dapat dikategorikan film tersebut termasuk dalam *Rating* 1, 2, 3, 4, atau 5. Namun dari beberapa *review* singkat dari penulis tersebut, ada *review* yang tidak begitu jelas termasuk kategori yang mana. Jadi, kami akan membuat sebuah aplikasi yang menerapkan *Text Mining* menggunakan metode *Naïve Bayes* yang dapat menentukan *review* singkat termasuk *Rating* 1, 2, 3, 4, atau 5.

Pada penelitian sebelumnya yang berkaitan dengan *Chi-Square* (*CHI*) (Kandarp Dave, 2011) menyatakan bahwa pengkategorisasian teks sangat penting, tetapi permasalahan *feature selection* sama banyak atau lebih penting daripada kategori teks. Dalam penelitian tersebut dibahas banyak topik penting mulai dari pengumpulan data, hingga pemrosesan data dan akhirnya menggunakan data yang telah diproses tersebut untuk dilakukan tes secara efisien menggunakan algoritme *feature selection*. Penelitian tersebut menunjukkan beberapa peningkatan dramatis menggunakan hasil tersebut. Dan metode pemilihan *feature* harus diteliti lebih lanjut, pada data skala sangat besar. Jumlah pelatihan dan dokumen uji yang digunakan dalam penelitian

tersebut sangat kecil dibandingkan apa yang tersebar luas di dunia maya. Selain itu, dalam penelitian tersebut, hanya ada 9 kategori. Di dunia nyata, ada ratusan kategori. Untuk memiliki pengkategorisasi berskala beasar. Algoritme *feature selection* harus secara kuat dikembangkan. Dan topik ini dapat diteliti dan diuji lebih lanjut. Dengan melakukan penelitian lebih lanjut tentang topik yang disebutkan tersebut akan membantu, karena pada akhirnya dapat membantu mengkategorikan semua dokumen di dunia.

Pada penelitian sebelumnya yang berkaitan dengan *feature selection* serta mencakup *Galavotti-Sebastiani-Simi Coefficient* (*GSS*) di dalamnya (Øystein Løhre Garnes, 2009) membahas tentang langkah-langkah untuk membangun pengklasifikasian menggunakan kumpulan file vektor (*feature* yang dipilih), mengevaluasi kinerjanya, dan menyimpan hasilnya di file. File berisi satu baris untuk setiap run, fold, dan dataset. Oleh karena itu dalam kasus penelitian tersebut, kalau hasil file dari percobaan *Naïve Bayes* berisi 600 baris hasil: 6 set ukuran *feature* (dari 500 hingga 10.000 *feature*) kali 10 kali fold 10 kali run (setiap kali lipat dijalankan), sedangkan hasil file dari percobaan *Support Vector Machine* berisi 150 baris hasil: 3 set ukuran *feature* (500, 1000, dan 2000 *feature*) kali 10 kali fold 5 kali run. Maka, dapat disimpulkan bahwa *Naïve Bayes* kinerjanya jauh lebih baik dibandingkan *Support Vector Machine*.

Pada penelitian sebelumnya yang berkaitan dengan *feature selection* (Simeon, 2008) menyatakan bahwa sebuah *feature* dapat meningkatkan akurasi proses perhitungan. Metode *feature selection* adalah metode yang sangat populer dalam berbagai penelitian. *Feature selection* digunakan untuk mengurangi dimensi dan mempercepat proses perhitungan. Selain itu *feature selection* juga mampu meningkatakan efisiensi dan akurasi dalam proses *document extraction* yang subset dengan pemilihan *feature* yang dianggap lebih relevan.

Sebagaimana masalah yang telah dijabarkan, untuk menentukan *Rating review* film, diperlukan penentuan *Rating* secara akurat. Alur dari sistem untuk metode yang digunakan penulis pada penelitian ini yakni menggunakan Metode *Multinomial Naïve Bayes Classifier* untuk mesin penentuan *Rating review* film. Oleh karena itu, pada penelitian ini akan dikembangkan penentuan *Rating* review film.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah yang terbentuk adalah:

1. Bagaimana perbandingan hasil akurasi Penentuan *Rating review* Film menggunakan Metode *Multinomial Naïve Bayes Classifier* baik dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* ataupun tidak?
2. Bagaimana pengaruh pengurangan dimensi *feature*?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Untuk mengetahui perbandingan hasil akurasi Penentuan *Rating review* Film menggunakan Metode *Multinomial Naïve Bayes Classifier* baik dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* ataupun tidak.
2. Untuk mengetahui pengaruh dimensi *feature*.

1.4 Manfaat

Manfaat dari penelitian ini adalah:

1. Bagi penulis
   a. Dapat memahami pengembangan sistem aplikasi dengan menerapkan *Text Mining* dengan Metode *Multinomial Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* untuk penentuan kategori dari *review* singkat film.
   b. Dapat memperluas wawasan dan mengembangkan ilmu yang telah diperoleh selama menempuh masa studi, yang utamanya adalah berhubungan dengan Metode *Multinomial Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*.
2. Bagi pengguna
   a. Dapat mengetahui *review* singkat film itu termasuk dalam kategori 1, 2, 3, 4, atau 5.
   b. Dapat mengetahui seberapa besar respon penulis dalam menuliskan *review* film berdasarkan *Rating* yang ditampilkan.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Aplikasi menerima *input review* film dan *Rating* film.
2. Aplikasi menampilkan *ouput review* film berupa *Rating* film.
3. Bahasa yang digunakan dalam *review* film adalah Bahasa Indonesia.
4. Proses penentuan *Rating* pada penelitian ini menggunakan *dataset offline*.
5. Terdapat 5 kelas pada penentuan kategori yaitu nilai *Rating* 1, 2, 3, 4, atau 5.
6. Jumlah data yang digunakan ada 300 data, yang terdiri dari 250 data latih (dengan pesebaran data 50 data untuk tiap-tiap *Rating*) dan 50 data uji.

1.6 Sistematika Pembahasan

Sistematika laporan pada penelitian ini dibagi dalam tujuh bab masing-masing diuraikan sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan mengenai dasar-dasar teori dan metode yang digunakan sebagai dasar untuk menyelesaikan permasalahan yang ada, seperti: Tinjauan Penelitian, *Text Mining*, *pre-processing*, Metode *Multinomial Naïve Bayes Classifier*, *Chi-Square* , *Galavotti-Sebastiani-Simi Coefficient*, dan evaluasi.

BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan tentang metode penelitian yang digunakan, meliputi: tahapan penelitian, teknik pengumpulan data, analisis kebutuhan, dan pengujian metode.

BAB IV ANALISIS DAN PERANCANGAN

Pada bab ini dijelaskan mulai dari deskripsi permasalahan, deskripsi umum sistem, *pre-processing*, *feature selection*, penyelesaian Metode *Multinomial Naïve Bayes Classifier*, perhitungan manual, perancangan antarmuka, perancangan pengujian, hingga kepada penarikan kesimpulan.

BAB V IMPLEMENTASI

Pada bab ini dijelaskan mengenai batasan implementasi, implementasi aplikasi, *pre-processing*, proses penetuan *Rating*, proses evaluasi, dan implementasi antarmuka.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang segala hal yang membahas mengenai pengujian, baik pengujian klasifikasi *Naïve Bayes Classifier* tanpa kombinasi *CHI-GSS*, maupun pengujian klasifikasi *Naïve Bayes Classifier* dengan kombinasi *CHI-GSS*.

BAB VII PENUTUP

Pada bab ini dijelaskan mengenai kesimpulan dan saran agar dapat mengembangkan aplikasi berikutnya yang lebih baik lagi.

BAB 2 LANDASAN KEPUSTAKAAN

Tinjauan pustaka berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Dalam landasan kepustakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian.

2.1 Tinjauan Penelitian

Dalam penulisan penelitian ini, dilakukan kajian terhadap penelitian-penelitian sebelumnya. Penelitian tersebut meliputi *Multinomial Naïve Bayes Classifier*, *Chi-Square* , dan *Galavotti-Sebastiani-Simi Coefficient*.

Pada penelitian sebelumnya yang berkaitan dengan *Multinomial Naïve Bayes Classifier* (NBC) yang berjudul Klasifikasi Emosi Untuk Teks Bahasa Indonesia Menggunakan Metode *Naïve Bayes* oleh Destuardi dan Surya, 2009 menyatakan bahwa Metode *Multinomial Naïve Bayes* merupakan algoritme yang naïve karena mengasumsikan indepedensi diantara kemunculan kata-kata dalam dokumen, tanpa memperhitungkan urutan kata dan informasi konteks dalam kalimat atau dokumen secara umum. Selain itu metode tersebut memperhitungkan jumlah kemunculan kata dalam dokumen.

Pada penelitian sebelumnya yang berkaitan dengan *Chi-Square* yang berjudul *Study of Feature Selection Algorithms for Text-Categorization* oleh Kandarp Dave, 2011 membahas tentang penelitian yang telah menunjukkan 6 algoritme pemilihan *feature* yang berbeda, dan *Chi-Square* dan NGL telah melakukan algoritme lain. Hasil ini tampaknya sesuai dengan hasil Kotcz, Prabakarmurthi, Kalita, dan Yang & Pedersen, karena mereka juga menemukan *Chi-Square* sangat efektif. Dasgupta dkk. menjelaskan bahwa "seringkali kesulitan untuk mengklaim lebih dari sekedar pemahaman intuitif yang samar tentang mengapa algoritme *feature selection* tertentu berkinerja baik saat terjadi".

Pada penelitian sebelumnya yang berkaitan dengan *feature selection* serta mencakup *Galavotti-Sebastiani-Simi Coefficient* di dalamnya yang berjudul *Feature Selection for Text Categorization* oleh Øystein Løhre Garnes, 2009 menyatakan bahwa *CHI* adalah salah satu metode pemilihan *feature* terbaik di seluruh penelitiannya. Untuk NB, secara kontinu mengklasifikasikan persentase dokumen tertinggi dengan benar, dan menunjukkan hasil terbaik dari semua metode. Untuk SVM, juga dihasilkan dengan sangat baik, yang berada di kelompok paling atas pada semua ukuran set *feature*.

Selanjutnya pada penilitian tersebut, dua metrik yang diawasi dan berkinerja terbaik, yaitu: *CHI* dan *GSS* digabungkan, menghasilkan hasil yang sangat baik. Karena koefisien *GSS* didasarkan pada *Chi-Square* , orang berhak menganggapnya sebagai kandidat kombinasi yang buruk. Matriks korelasi pada penelitian menunjukkan korelasi 79% pada 5000 *feature*. Ketika keduanya menampakkan kinerja yang baik, ini menunjukkan bahwa mungkin ada kemungkinan kinerja

yang lebih tinggi apabila keduanya dikombinasikan. Begitu juga halnya dengan angka yang ditunjukkan. Upaya kombinasi mencapai persentase yang lebih tinggi dari dokumen yang dikategorikan dengan benar daripada kedua metode untuk 500 sampai 3000 *feature*. Sementara *CHI* adalah metode single terbaik, kombinasi *CHI* + *GSS* memiliki hasil terbaik dari semua klasifikasi NB yang dievaluasi, yang diukur dalam persentase dokumen yang diklasifikasikan dengan benar.

Terakhir, peneliti telah melakukan serangkaian kombinasi penelitian, dan hasil terbaik yang cocok digunakan pada NB adalah dengan menggabungkan metrik *CHI* dan *GSS*.

2.2 *Text Mining*

Text Mining adalah proses ekstraksi pola (informasi dan pengetahuan yang berguna) dari sejumlah besar sumber data tak terstruktur. Penambangan teks memiliki tujuan dan menggunakan proses yang sama dengan penambangan data, namun memiliki masukan yang berbeda. Masukan untuk penambangan teks adalah data yang tidak (atau kurang) terstruktur, seperti dokumen Word, PDF, kutipan teks, dll., sedangkan masukan untuk penambangan data adalah data yang terstruktur (Feldman, 2007). Penambangan teks dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini dengan menggunakan teknik dan alat yang sama dengan penambangan data. Area penerapan penambangan teks yang paling populer adalah:

1. Ekstraksi informasi (*information extraction*): Identifikasi frasa kunci dan keterkaitan di dalam teks dengan melihat urutan tertentu melalui pencocokan pola.
2. Pelacakan topik (*topic tracking*): Penentuan dokumen lain yang menarik seorang pengguna berdasarkan profil dan dokumen yang dilihat pengguna tersebut.
3. Perangkuman (*summarization*): Pembuatan rangkuman dokumen untuk mengefisienkan proses membaca.
4. Kategorisasi (*categorization*): Penentuan tema utama suatu teks dan pengelompokan teks berdasarkan tema tersebut ke dalam kategori yang telah ditentukan.
5. Penggugusan (*clustering*): Pengelompokan dokumen yang serupa tanpa penentuan kategori sebelumnya (berbeda dengan kategorisasi di atas).
6. Penautan konsep (*concept linking*): Penautan dokumen terkait dengan identifikasi konsep yang dimiliki bersama sehingga membantu pengguna untuk menemukan informasi yang mungkin tidak akan ditemukan dengan hanya menggunakan metode pencarian tradisional.
7. Penjawaban pertanyaan (*question answering*): Pemberian jawaban terbaik.

2.3 *Pre-processing*

Struktur data yang baik dapat memudahkan proses komputerisasi secara otomatis. Pada *Text Mining*, informasi yang akan digali berisi informasi-informasi yang strukturnya sembarang. Oleh karena itu, diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhannya untuk proses dalam data mining, yang biasanya akan menjadi nilai-nilai numerik. Proses ini sering disebut Text Preprocessing (Feldman, 2007). Setelah data menjadi data terstruktur dan berupa nilai numerik maka data dapat dijadikan sebagai sumber data yang dapat diolah lebih lanjut. Berberapa proses yang dilakukan adalah *case folding*, *tokenizing*, *filtering*, dan *stemming*.

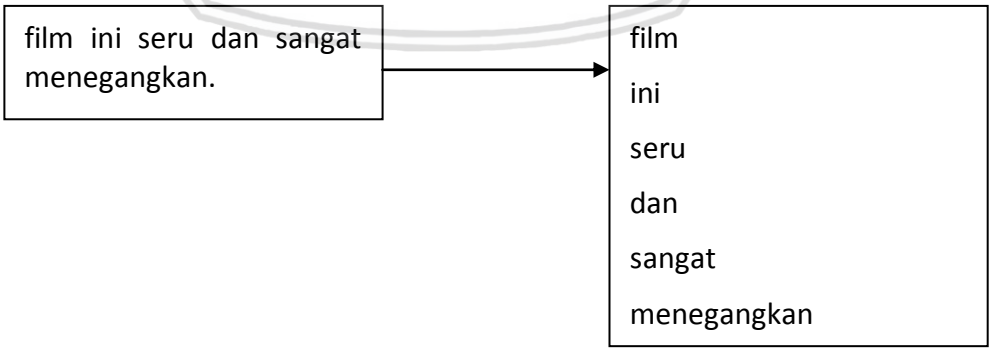
2.3.1 *Case Folding*

case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf "a" sampai dengan "z" yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter (Feldman, 2007), yang ditunjukkan pada Gambar 2.1.

Gambar 2.1 *Case Folding*

2.3.2 *Tokenizing*

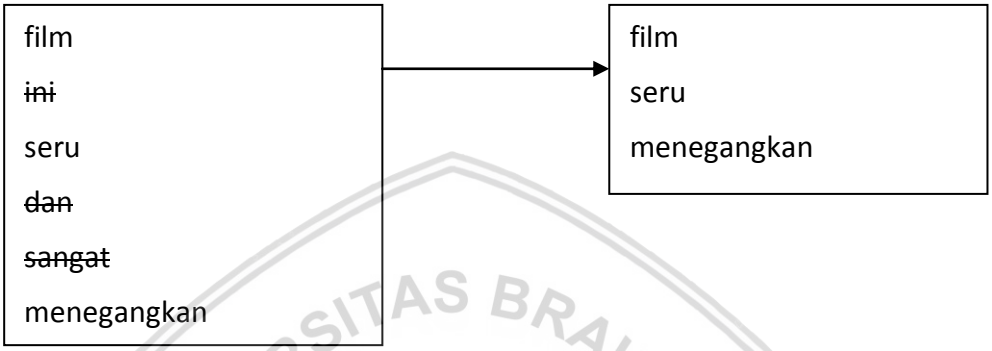
Tahap *tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya, yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 *Tokenizing*

2.3.3 *Filtering*

Tahap *filtering* adalah tahap mengambil kata - kata penting dari hasil token. Bisa menggunakan algoritme stoplist (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words* (Feldman, 2007), yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 *Filtering*

2.3.4 *Stemming*

Pada penelitian yang dilakukan Ledy Agusta, 2009 menyatakan bahwa Algoritme Nazief dan Adriani memiliki prosentase keakuratan (presisi) lebih besar daripada Algoritme Porter. Apabila makin lengkap kamus yang digunakan, maka makin besar akurasi yang didapat, karena kamus mempengaruhi presisi. Algoritme Nazief dan Andriani terdapat derivational affixes yaitu *prefix*es, *suffix*es, atau kombinasi keduanya (con-fixes), yang terdiri dari tahapan berikut ini:

1. Infectional *suffix*es (IS)
   Yaitu membuang akhiran yang tidak mengubah kata dasar, yang terdapat 2 tipe (Moeliono dan Dardjowidjojo, 1998) sebagai berikut:
   a. Particel (P)
      Yatu membuang kata partikel perintah.
      Di antaranya: “-kah”, ”-lah”, ”-tah”, dan ”-pun”.
   b. Possessive Pronouns (PP)
      Yaitu membuang kata ganti kepunyaan.
      Di antaranya: “-ku”, ”-mu”, dan ”-nya”.
2. Derivational *prefix*es (DP)
   Yaitu membuang awalan yang mengubah kata dasar.
   Di antaranya: “be-”, ”di-”, ”ke-”, ”me-”, ”pe-”, ”se-”, dan ”te-”.
3. Derivational *suffix*es (DS)
   Yaitu membuang akhiran yang mengubah kata dasar.
   Di antaranya: “-i”, ”-kan”, dan ”-an”.

4. Derivational Confixes (DC)
   Yaitu membuang awalan dan akhiran yang mengubah kata dasar.
   Di antaranya: “be-an”, ”me-i”, ”me-kan”, ”di-i”, ”di-kan”.

Tidak semua kombinasi *prefix* dan *suffix* diizinkan, ada juga yang tidak diizinkan sebagaimana yang ditunjukkan pada Tabel 2.1 berikut ini:

Tabel 2.1 Kombinasi *Prefix* dan *Suffix* yang Tidak Diizinkan

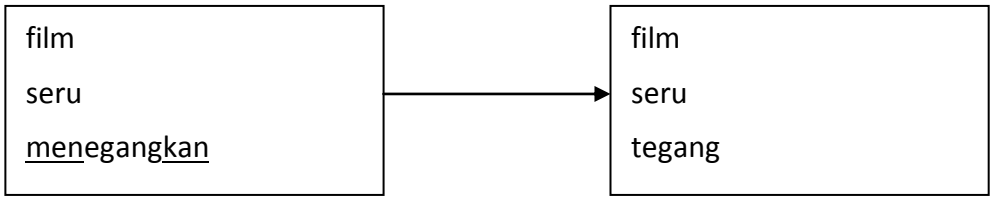
Prefix	***Suffix* yang Tidak Diizinkan**
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

Aturan kata berimbuhan dalam Bahasa Indonesia dapat digambarkan sebagai berikut:

$$\left[\left[\left[DP+\right]DP+\right]DP+\right] root-word[[+DS][+PP][+P]]$$

Gambar 2.4 Aturan Kata Berimbuhan

Tahap *stemming* adalah tahap mencari root kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama, yang ditunjukkan pada Gambar 2.4.



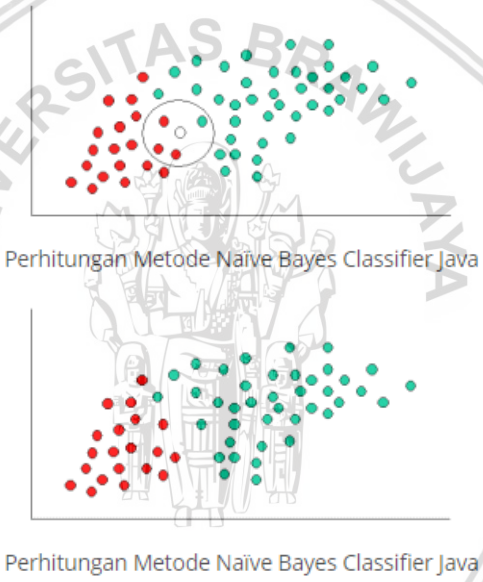
Gambar 2.5 *Stemming*

2.4 *Naïve Bayes Classifier*

Naïve Bayes Classifier merupakan sebuah pengklasifikasi probabilitas sederhana yang mengaplikasikan *Teorema Bayes* dengan asumsi ketidaktergantungan (independent) yang tinggi. *Teorema Bayes* adalah teorema

yang dipakain dalam statistika untuk menghitung peluang untuk suatu hipotesis. *Bayes Optimal Classifier* menghitung peluang dari suatu kelas dari masing-masing kelompok atribut yang ada, menentukan kelas mana yang paling optimal.

Pengklasifikasian menggunakan *Teorema Bayes* ini membutuhkan biaya komputasi yang mahal (waktu prosessor ukuran memori yang besar) karena kebutuhan untuk menghitung nilai probabilitas untuk tiap nilai dari perkalian kartesius untuk tiap nilai atribut tiap nilai kelas. Data latih untuk *Teorema Bayes* membutuhkan paling tidak perkalian kartesius dari seluruh kelompok atribut yang mungkin, jika misalkan ada 16 atribut yang masing- masingnya berjenis boolean tanpa missing value, maka data latih minimal yang dibutuhkan oleh *Teorema Bayes* untuk dipakain dalam klasifikasi adalah 216 = 65.536 data. Untuk mengatasi kekurangan tersebut maka dipakain *Naïve Bayes*.



Gambar 2.6 Model *Naïve Bayes Classifier*

Secara umum model *Naïve Bayes Classifier* adalah sebagai berikut:

$$P(C_k|F) = \frac{P(C_k) \times P(F|C_k)}{P(F)} \tag{2.1}$$

Keterangan:

$P(C_k|F) = peluang\ kategori\ C_k\ \ ketika\ terdapat\ kemunculan\ fitur\ F.$

$P(F|C_k) = conditional\ probability\ fitur\ F\ pada\ ke\ dalam\ kategori\ C_k$.

$P(C_k) = peluang\ kemunculan\ kategori\ C_k$.

$P(F) = peluang\ kemunculan\ fitur\ F.$

Persamaan di atas menjelaskan secara umum pada satu *feature* saja, maka secara garis besar dapat dijabarkan sebagai berikut:

$$P(C_k|F_1, \dots, F_n) = \frac{P(C_k) \times P(F_1, \dots, F_n|C_k)}{P(F_1, \dots, F_n)} \tag{2.2}$$

Atau dengan kata lain persamaan di atas dapat digambarkan sebagai:

$$Posterior = \frac{Prior * Likelihood}{Evidence} \tag{2.3}$$

Peluang kemunculan kata (*Prior*) dapat dihitung menggunakan rumus:

$$P(C_k) = \frac{N_{C_k}}{N} \tag{2.4}$$

Keterangan:

$N_{C_k} = banyak\ dokumen\ berkategori\ C_k\ pada\ dokumen\ latih.$

$N = jumlah\ dokumen\ latih\ yang\ digunakan.$

2.4.1 *Bernoulli Naïve Bayes*

Bernoulli Naïve Bayes digunakan untuk mengklasifikasikan short-text, menggunakan binary *term*-weighted pada setiap *term*, yaitu 0 dan 1. Lain halnya dengan *term*-frequency yang melakukan *term*-weighted pada setiap *term*.

2.4.2 *Gaussian Naïve Bayes*

Gaussian Naïve Bayes digunakan untuk merepresentasikan permasalahan probabilitas bersyarat dari *feature* kontinu pada sebuah kelas $P(X_i|Y)$. Terdapat dua parameter sebagai karakteristiknya, yaitu: mean dan varian.

2.4.3 *Multinomial Naïve Bayes*

Multinomial Naïve Bayes digunakan untuk mengasumsikan independensi kemunculan kata dalam dokumen. Metode ini tidak memperhitungkan urutan kata dan information-context dalam dokumen, namun memperhitungkan jumlah kata dalam dokumen (Destuardi dan Sumpeno, 2009).

Dengan persamaan:

$$(F, C_k) = \frac{count\ (F, C_k) + 1}{(\sum_{F \in V} count(F, C_k)) + |V|} \tag{2.5}$$

Keterangan:

$count(F, C_k) = \ jumlah\ fitur\ F\ yang\ muncul\ dalam\ suatu\ kategori\ C_k.$

$+1 = penambahan\ nilai\ 1\ untuk\ menghindari\ nilai\ zero.$

$\sum_{F \in V} count(F, C_k) = jumlah\ seluruh\ fitur\ F\ pada\ kategori\ \ C_k.$

$|V| = \ jumlah\ seluruh\ fitur\ unik\ di\ seluruh\ kategori.$

Contoh perhitungan *Multinomial Naïve Bayes* ditunjukan dari Tabel 2.2 sampai Tabel 2.5 berikut ini:

Tabel 2.2 Klasifikasi Dokumen

Data	**Dokumen**	**Kalimat**	**Kategori**
Latih	1	Aku kami aku	1
	2	Aku aku kita	1
	3	Aku saya	1
	4	Kalian kamu aku	2
Uji	5	Aku aku aku kalian kamu	?

Dari Tabel 2.2 di atas dengan berdasarkan rumus persamaan 2.4, maka dapat diperoleh nilai *Prior* yang ditunjukkan pada Tabel 2.3 berikut ini:

Tabel 2.3 *Prior* Kategori

Prior 1	***Prior 2***
$\frac{3}{4} = 0{,}75$	$\frac{1}{4} = 0{,}25$

Dari Tabel 2.3 di atas dengan berdasarkan rumus persamaan 2.5, maka dapat diperoleh nilai *Conditional Probability* yang ditunjukkan pada Tabel 2.4 berikut ini:

Tabel 2.4 *Conditional Probability*

Feature	***Conditional Probability 1***	***Conditional Probability 2***
Aku	$\frac{(5+1)}{(8+6)} = \frac{6}{14}$	$\frac{(1+1)}{(3+6)} = \frac{2}{9}$
Kalian	$\frac{(0+1)}{(8+6)} = \frac{1}{14}$	$\frac{(1+1)}{(3+6)} = \frac{2}{9}$
Kamu	$\frac{(0+1)}{(8+6)} = \frac{1}{14}$	$\frac{(1+1)}{(3+6)} = \frac{2}{9}$

Dari Tabel 2.4 di atas dengan berdasarkan rumus persamaan 2.1, maka dapat diperoleh nilai *Posterior* yang ditunjukkan pada Tabel 2.5 berikut ini:

Tabel 2.5 *Posterior* Kategori

Posterior 1	***Posterior 2***
$\frac{6}{14} * \frac{1}{14} * \frac{1}{14} * \frac{3}{4} = 0{,}0003$	$\frac{2}{9} * \frac{2}{9} * \frac{2}{9} * \frac{1}{4} = 0{,}0001$

2.5 *Chi-Square*

Chi-Square adalah metode statistik yang pada awalnya digunakan dalam analisis statistik untuk mengukur bagaimana hasil pengamatan berbeda (yaitu independen) dari hasil yang diharapkan sesuai dengan hipotesis awal (nilai yang lebih tinggi menunjukkan kemandirian yang lebih tinggi). Dalam konteks statistik klasifikasi teks digunakan untuk mengukur seberapa independen sebuah kata dan sebuah kelas (Gulden Uchyigit, 2012).

Dengan persamaan:

$$x^2(F, C_k) = \frac{N \times ((N_{F,C_k} \times N_{\bar{F},\bar{C_k}}) - (N_{F,\bar{C_k}} \times N_{\bar{F},C_k}))^2}{N_F \times N_{\bar{F}} \times N_{C_k} \times N_{\bar{C_k}}}$$

(2.6)

Dan persamaan untuk menghitung *weighted average* adalah sebagai berikut:

$$x^2(F) = \sum_{k=1}^{|C|} \frac{N_{C_k}}{N} x^2(F, C_k)$$

(2.7)

Keterangan:

$N = jumlah\ total\ dokumen\ dalam\ training\ set.$

$N_{C_k} = jumlah\ dokumen\ dalam\ kategori\ C_k.$

$N_{\overline{C_k}} = jumlah\ dokumen\ yang\ tidak\ ada\ dalam\ kategori\ C_k.$

$N_F = jumlah\ dokumen\ yang\ berisi\ fitur\ F.$

$N_{\bar{F}} = jumlah\ dokumen\ yang\ tidak\ berisi\ fitur\ F.$

$N_{F,C_k} = jumlah\ dokumen\ yang\ berisi\ fitur\ F\ dalam\ kategori\ C_k.$

$N_{\bar{F},C_k} = jumlah\ dokumen\ yang\ tidak\ berisi\ fitur\ F\ dalam\ kategori\ C_k.$

$N_{F,\overline{C_k}} = jumlah\ dokumen\ yang\ berisi\ fitur\ F\ tidak\ ada\ dalam\ kategori\ C_k.$

$N_{\bar{F},\overline{C_k}} = jumlah\ dokumen\ yang\ tidak\ berisi\ fitur\ F\ tidak\ ada\ dalam\ kategori\ C_k.$

2.6 *Galavotti-Sebastiani-Simi Coefficient*

GSS Coefficient adalah metode yang diusulkan oleh Galavotti dkk. Yang merupakan sebuah statistik *Chi-Square* yang disederhanakan. Mereka menghapus faktor pada pembilang karena sama untuk semua pasang dan karena itu menjadi berlebihan. Mereka kemudian menghapus ini memiliki nilai rendah untuk kata-kata langka yang berarti bahwa kata-kata tersebut diberi skor tinggi (karena ini adalah bagian dari penyebut). Tapi kata-kata langka yang ditunjukkan oleh adalah yang paling tidak efektif dalam klasifikasi teks. Akhirnya, mereka menghapus faktor dari penyebut karena akan menekankan kategori yang sangat langka (yaitu kategori dengan contoh yang sangat sedikit) (Gulden Uchyigit, 2012).

Dengan persamaan:

$$GSS(F, C_k) = N_{F,C_k} N_{\bar{F},\overline{C_k}} - N_{F,\overline{C_k}} N_{\bar{F},C_k} \tag{2.8}$$

Dan persamaan untuk menghitung nilai max adalah sebagai berikut:

$$GSS(F, C_k) = \max_{k=1}^{|C|} GSS(F, C_k) \tag{2.9}$$

Keterangan:

$N = jumlah\ total\ dokumen\ dalam\ training\ set.$

$N_{F,C_k} = jumlah\ dokumen\ yang\ berisi\ fitur\ F\ dalam\ kategori\ C_k.$

$N_{\bar{F},C_k} = jumlah\ dokumen\ yang\ tidak\ berisi\ fitur\ F\ dalam\ kategori\ C_k.$

$N_{F,\overline{C_k}} = jumlah\ dokumen\ yang\ berisi\ fitur\ F\ tidak\ ada\ dalam\ kategori\ C_k.$

$N_{\bar{F},\overline{C_k}} = jumlah\ dokumen\ yang\ tidak\ berisi\ fitur\ F\ tidak\ ada\ dalam\ kategori\ C_k.$

2.7 Evaluasi

Proses evaluasi dilakukan demi menghitung akurasi dari sebuah aplikasi yang dibuat. Nilai akurasi didapat dari prosentase perbandingan jumlah data uji benar dengan jumlah seluruh data uji.

Dengan persamaan:

$$Akurasi = \frac{jumlah\ data\ uji\ benar}{jumlah\ seluruh\ data\ uji} \times 100\% \quad (2.10)$$

BAB 3 METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan beberapa hal yaitu tipe penelitian, strategi dan rancangan penelitian, serta jadwal penelitian.

3.1 Tipe Penelitian

Pelaksanaan tipe penelitian ini termasuk nonimplementatif karena menitikberatkan pada investigasi terhadap fenomena tertentu atau analisis terhadap hubungan antar fenomena yang sedang dikaji untuk kemudian menghasilkan hasil analisis ilmiah sebagai produk utamanya. Metode yang digunakan untuk menghasilkan produk utama berupa eksperimentasi, studi kasus, penelitian tindakan (*action research*), dan observasi.

Jika ditinjau dari kegiatan penelitiannya, pendekatan pada penelitian tipe ini berupa analitik (*analytical/explanatory*) yaitu sebuah kegiatan penelitian nonimplementatif yang dilakukan untuk menjelaskan derajat hubungan antar elemen dalam objek penelitian dengan fenomena tertentu yang sedang diteliti. Produk utama yang dihasilkan adalah hasil analisis.

3.2 Strategi dan Rancangan Penelitian

Strategi dan rancangan penelitian dibagi menjadi delapan bagian, yaitu strategi/metode, subjek atau partisipan penelitian, lokasi penelitian, metode/teknik pengumpulan data, metode/teknik analisis data, peralatan pendukung, model proses sistem, dan implementasi algoritme.

3.2.1 Strategi/Metode

Strategi/metode secara umum yang digunakan adalah eksperimen yaitu dengan menciptakan hipotesis, metode pengumpulan data, metode analisis, metode eksperimen, dan meghasilkan produk utama. Sebagai indikator yaitu membandingkan minimal dua buah algoritme yang akan dianalisis. Metode eksperimen yang digunakan dalam penelitian ini, yaitu Metode *Multinomial Naïve Bayes Classifier* (NBC) dengan *feature selection* berbasis *Chi-Square* (*CHI*) dan *Galavotti-Sebastiani-Simi Coefficient* (*GSS*). Alasan pemilihan metode tersebut telah dijelaskan pada sebelumnya.

3.2.2 Subjek atau Partisipan Penelitian

Subjek atau partisipan penelitian yang terlibat adalah para *review*er dari web https://montasefilm.com dan http://www.ulasanpilem.com. Alasan pemilihan partisipan tersebut karena kedua web tersebut merupakan web yang telah menggunakan Bahasa Indonesia baku yang berisi berbagai macam kategori sehingga sangat tepat dijadikan partisipan.

3.2.3 Lokasi Penelitian

Lokasi penelitian yang dipilih adalah Laboratorium Komputasi Cerdas milik Fakultas Ilmu Komputer Universitas Brawijaya Malang. Alasan pemilihan lokasi tersebut karena peralatan yang lengkap terkait penentuan *Rating review* film menggunakan Metode *Multinomial Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Cofficient*.

3.2.4 Metode/Teknik Pengumpulan Data

Pengumpulan data merupakan metode untuk mendapatkan data sampel sebagai acuan untuk mengembangkan perangkat lunak. Data sampel yang dimaksud adalah data-data *review* singkat film. Data sampel didapatkan dari dua web, yaitu https://montasefilm.com dan http://www.ulasanpilem.com yang sudah dikategorikan *Rating* 1, 2, 3, 4, atau 5 dan tervalidasi berdasarkan yang telah tercantum.

3.2.5 Metode/Teknik Analisis Data

Analisis data dilakukan dengan menentukan kategori film dari *review* singkat film. Jadi, mengkategorikan beberapa dari *review* singkat film ke dalam *Rating* 1, 2, 3, 4, atau 5. Lalu data tadi diproses dengan Metode *Multinomial Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* untuk implementasi sistem.

3.2.6 Peralatan Pendukung

Peralatan pendukung yang dibutuhkan pada sistem dapat mempengaruhi performa sistem. Peralatan pendukung pada sistem dijelaskan dalam paparan berikut:

1. Hardware:
    a. PC, Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz (4 CPUs), ~2.40GHz.
    b. RAM 8192MB.
    c. Monitor 14inch.
2. Aplikasi berjalan secara offline karena tidak membutuhkan akses ke database dalam server, melalui perangkat lunak dengan spesifikasi:
    a. Ope*Rating* System Windows 7 Ultimate 64-bit.
    b. Aplikasi Netbeans.
    c. Java jdk.

3.2.7 Model Proses Sistem

Model proses sistem adalah cara pengguna dapat berinteraksi dengan sistem adalah sebagai berikut:

1. Sistem dapat menerima input informasi dari pengguna mengenai *review* film beserta *Rating* film dari pengguna.

2. Sistem mampu memproses data input pengguna untuk dapat menghasilkan *review* film dan *Rating* film dari data sampel dan juga sistem dapat menampilkan *Rating review* film dari pengguna.
3. Sistem mampu menampilkan *review* film, perhitungan, dan *Rating* baik dari hasil analisis sistem berdasarkan perhitungan data sampel atau data training, maupun dari data sampel atau data training sebelumnya.

3.2.8 Implementasi Algoritme

Pengujian sistem yang dilakukan berkaitan dengan pengujian validasi sistem. Tahap ini berfungsi untuk memastikan apakah sistem yang dibuat dapat memperbaik permasalahan sebelumnya dan sejauh mana sistem dapat memepengaruhi permasalah yang terjadi. Analisis sistem aplikasi dilakukan dengan membandingkan antara data sampel yang telah dikumpulkan sebelumnya dengan hasil setelah sistem aplikasi diterapkan.

Pengujian dan analisis juga dilakukan untuk mengetahui kinerja sistem dalam melayani pengguna sesuai dengan kebutuhan. Kualitas pemrosesan aplikasi dapat dilihat dari proses, antarmuka, maupun aturan yang ada dalam sistem.

Pengujian parameter dilakukan berkali-kali agar dapat mendeteksi kesalahan yang terjadi sehingga mampu menghasilkan sistem yang valid dan sesuai. Sistem yang valid dapat didapatkan dengan hasil pengujian yang mendekati tingkat akurasi 100%. Tingkat akurasi didapatkan dari prosentase kesalahan dan kebenaran sistem dalam menyimpulkan harga kos.

BAB 4 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

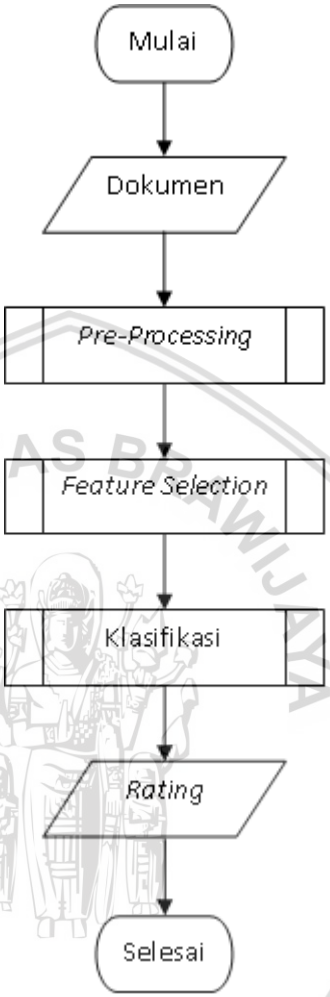
Dalam suatu *review* terhadap suatu karya seperti halnya film sangatlah penting untuk dikaji dari pihak produksi film maupun pihak penonton film. Dari pihak produksi bisa mengkaji *review* tersebut untuk kelanjutan produksi film tersebut apakah akan melakukan perbaikan atau pengembangan kualitas dan kuantitas dari film terkait. Tak lepas juga dari pihak konsumen atau penonton film, dari *review-review* yang disampaikan oleh orang yang sudah pernah menonton film tersebut maka hal tersebut akan menjadi tolak ukur bagi konsumen untuk menonton film tersebut atau tidak. Seringkali penonton memberikan *review* pada suatu film atau memberikan *Rating* atau nilai untuk produk dengan parameter suatu nilai dari pada suatu interval tertentu. Maka dari itu terkadang pihak produksi tidak bisa memprediksi *Rating* atau nilai pada suatu *review* tertentu sehingga sulit untuk mengkategorikan nilai atau *Rating* dari *review* yang diberikan. Dari permasalahan tersebut bisa diselesaikan dengan menggunakan metode prediksi *Rating* untuk memprediksi *Rating* pada suatu *review*.

Banyak metode atau teknik yang digunakan untuk mengklasifikasikan sebuah teks ke dalam suatu kelas atau kategori tertentu. Salah satunya yaitu Metode *Multinomial Naïve Bayes Classifier* untuk melakukan klasifikasi sebuah teks pada kelas atau kategori tertentu. Dalam Metode *Multinomial Naïve Bayes Classifier*, hasil klasifikasi akan tergantung pada probabilitas nilai frekuensi kemunculan sebuah dokumen. Namun karena *review* yang diberikan tergolong tidak terlalu panjang dan cukup banyak maka akan dilakukan *feature selection* untuk mengurangi *noise* atau *feature-feature* yang tidak relevan. *feature selection* yang digunakan pada kasus ini adalah menggunakan *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*, dimana banyak atau porsi suatu kata diperhatikan pada suatu kelas yang bertujuan untuk melihat probabilitas sebuah kata apakah menggambarkan suatu kelas tertentu atau tidak.

4.2 Deskripsi Umum Sistem

Sistem yang akan dikembangkan untuk menyelesaikan permasalahan pada Penentuan *Rating review* Film Menggunakan Metode *Multinomial Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*. Dalam Metode *Multinomial Naïve Bayes Classifier*, hasil klasifikasi tergantung pada nilai probabilitas frekuensi kemunculan dokumen uji terhadap dokumen latih. Optimasi dilakukan untuk mendapatkan nilai akurasi yang lebih tinggi. Penambahan *Chi-Square* dan *Galavotti-Sebastiani-Simi*

Coefficient digunakan untuk mengurangi *noise* atau mengurangi *feature* yang dianggap tidak relevan untuk dilakukan klasifikasi. Secara keseluruhan deskripsi umum sistem digambarkan pada skema Gambar 4.1.



Gambar 4.1 Deskripsi Umum Sistem

4.3 Perhitungan Manual

Perhitungan manual merupakan tahapan untuk memberikan gambaran umum mengenai proses klasifikasi dokumen tanpa menggunakan bantuan sistem. Kategori yang digunakan dalam perhitungan manual adalah sebanyak 5 kategori. Untuk data latih memiliki 10 dokumen. Untuk data uji yang diberikan sebanyak 1 dokumen uji. Kategori yang digunakan adalah *Rating* 1-5.

4.3.1 *Multinomial Naïve Bayes*

Misalkan terdapat *review Editor* terhadap beberapa film pada dua web, yaitu https://montasefilm.com dan http://www.ulasanpilem.com yang ditunjukkan pada Tabel 4.1

Tabel 4.1 *Review Editor* Terhadap Film

No.	Dok	*Review*	*Rating*
1	Latih	Terlalu khayal, tidak begitu sempurna.	2
2		Sangat nyata, rileks sekali.	4
3		Bagus, begitu sempurna, seru, nyata.	4
4		Agak khayal, terlalu sempurna, tampil bagus.	3
5		Cepat bosan, mata jadi merah.	1
6		Tidak berfantasi, begitu sempurna, bagus.	4
7		Tidak rekomen, menyesal amat.	1
8		Mata jenuh, tidak sesuai.	2
9		Bagus sekali, sesuai untuk remaja.	4
10		Terasa ringan, rileks di mata.	5
1	Uji	Bagus amat, sesuai sekali.	?

Kemudian dilakukan tahap *pre-processing*, yaitu *case folding* dan *tokenizing*, *filtering*, dan *stemming*, seperti yang ditunjukkan pada Tabel 4.2, Tabel 4.3, dan Tabel 4.4 berikut ini.

Berdasarkan dengan proses yang ditunjukkan pada Gambar 2.1 dan Gambar 2.2 di atas maka diperoleh Tabel 4.2.

Tabel 4.2 *Case Folding* dan *Tokenizing*

No.	Dok	*Review*	*Rating*
1	Latih	terlalu khayal tidak begitu sempurna	2
2		sangat nyata rileks sekali	4
3		bagus begitu sempurna seru nyata	4
4		agak khayal sempurna tampil bagus	3
5		cepat bosan mata jadi merah	1
6		tidak berfantasi begitu sempurna bagus	4
7		tidak rekomen menyesal amat	1

8		mata jenuh tidak sesuai	2
9		bagus sekali sesuai untuk remaja	4
10		terasa ringan rileks di mata	5
1	Uji	bagus amat sesuai sekali	?

Berdasarkan dengan proses yang ditunjukkan pada Gambar 2.3 di atas maka diperoleh Tabel 4.3.

Tabel 4.3 *Filtering*

No.	Dok	*Review*	*Rating*
1	Latih	terlalu khayal tidak begitu sempurna	2
2		sangat nyata rileks sekali	4
3		bagus begitu sempurna seru nyata	4
4		agak khayal terlalu sempurna tampil bagus	3
5		cepat bosan mata merah	1
6		tidak berfantasi begitu sempurna bagus	4
7		tidak rekomen menyesal amat	1
8		mata jenuh tidak sesuai	2
9		bagus sekali sesuai remaja	4
10		terasa ringan rileks mata	5
1	Uji	bagus amat sesuai sekali	?

Berdasarkan dengan proses yang ditunjukkan pada Gambar 2.5 di atas maka diperoleh Tabel 4.5.

Tabel 4.4 *Stemming*

No.	Dok	*Review*	*Rating*
1	Latih	terlalu khayal tidak begitu sempurna	2
2		sangat nyata rileks sekali	4
3		bagus begitu sempurna seru nyata	4
4		agak khayal terlalu sempurna tampil bagus	3

5		cepat bosan mata merah	1
6		tidak fantasi begitu sempurna bagus	4
7		tidak rekomen sesal amat	1
8		mata jenuh tidak sesuai	2
9		bagus sekali sesuai remaja	4
10		terasa ringan rileks mata	5
1	Uji	bagus amat sesuai sekali	?

Tabel 4.4 menunjukkan data latih dan data uji yang kemudian akan dicari nilai frekuensinya seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Frekuensi *Term*

Kata	**Frekuensi**				
	***Rating* 1**	***Rating* 2**	***Rating* 3**	***Rating* 4**	***Rating* 5**
Terlalu	0	1	1	0	0
Khayal	0	1	1	0	0
Tidak	1	2	0	1	0
Begitu	0	1	0	2	0
sempurna	0	1	1	2	0
Sangat	0	0	0	1	0
Nyata	0	0	0	2	0
Rileks	0	0	0	1	1
Sekali	0	0	0	2	0
Bagus	0	0	1	3	0
Seru	0	0	0	1	0
Agak	0	0	1	0	0
Tampil	0	0	1	0	0
Cepat	1	0	0	0	0
Bosan	1	0	0	0	0
Mata	1	1	0	0	1
Merah	1	0	0	0	0
Fantasi	0	0	0	1	0
Rekomen	1	0	0	0	0
Sesal	1	0	0	0	0
Amat	1	0	0	0	0
Jenuh	0	1	0	0	0
Sesuai	0	1	0	1	0
Remaja	0	0	0	1	0

Terasa	0	0	0	0	1
Ringan	0	0	0	0	1

Dari Tabel 4.4 diperoleh nilai *Prior* dengan rumus Persamaan 2.4 yang ditunjukkan oleh Tabel 4.6.

$$Prior: P(C_k) = \frac{N_{C_k}}{N}$$

Tabel 4.6 *Prior Naïve Bayes*

***Prior* (R1)**	***Prior* (R2)**	***Prior* (R3)**	***Prior* (R4)**	***Prior* (R5)**
$\frac{2}{10}$ =0,200	$\frac{2}{10}$ =0,200	$\frac{1}{10}$ =0,100	$\frac{4}{10}$ =0,400	$\frac{1}{10}$ =0,100

Langkah selanjutya mencari *Conditional Probability* dengan rumus Persamaan 2.5 yang ditunjukkan oleh Tabel 4.7.

$$Conditional\ Probability: P(F, C_k) = \frac{count\ (F,C_k)+1}{(\sum_{F \in V} count(F,C_k))+|V|}$$

Tabel 4.7 *Conditional Probability Naïve Bayes*

Kata	***Rating* 1**	***Rating* 2**	***Rating* 3**	***Rating* 4**	***Rating* 5**
bagus	$\frac{0+1}{8+26}$ = 0,02941176	$\frac{0+1}{9+26}$ = 0,02857143	$\frac{1+1}{6+26}$ = 0,0625	$\frac{3+1}{18+26}$ = 0,09090909	$\frac{0+1}{4+26}$ = 0,03333333
amat	$\frac{1+1}{8+26}$ = 0,05882353	$\frac{0+1}{9+26}$ = 0,02857143	$\frac{0+1}{6+26}$ = 0,03125	$\frac{0+1}{18+26}$ = 0,02272727	$\frac{0+1}{4+26}$ = 0,03333333
sesuai	$\frac{0+1}{8+26}$ = 0,02941176	$\frac{1+1}{9+26}$ = 0,05714286	$\frac{0+1}{6+26}$ = 0,03125	$\frac{1+1}{18+26}$ = 0,04545455	$\frac{0+1}{4+26}$ = 0,03333333
sekali	$\frac{0+1}{8+26}$ = 0,02941176	$\frac{0+1}{9+26}$ = 0,02857143	$\frac{0+1}{6+26}$ = 0,03125	$\frac{2+1}{18+26}$ = 0,06818182	$\frac{0+1}{4+26}$ = 0,03333333

Langkah selanjutnya adalah mencari nilai *Posterior* dengan rumus seperti pada Persamaan 2.1 kemudian nilai tersebut dibandingkan. Nilai tertinggi merupakan keputusan yang diambil kata masuk pada kelas *Rating*. Nilai *Posterior* dapat dilihat pada Tabel 4.8.

$$(C_k|F) = \frac{P(C_k) \times P(F|C_k)}{P(F)}$$

Tabel 4.8 *Posterior Naïve Bayes*

Rating	***Posterior***
Rating 1	0,200 x 0,02941176 x 0,05882353 x 0,02941176 x 0,02941176 = **0,02941176**
Rating 2	0,200 x 0,02857143 x 0,02857143 x 0,05714286 x 0,02857143 = **0,02857143**
Rating 3	0,100 x 0,0625 x 0,03125 x 0,03125 x 0,03125 = **0,015625**
Rating 4	0,400 x 0,09090909 x 0,02272727 x 0,04545455 x 0,06818182 = **0,09090909**
Rating 5	0,100 x 0,03333333 x 0,03333333 x 0,03333333 x 0,03333333 = **0,01333333**

Dari hasil perhitungan manual tersebut menunjukkan bahwa nilai tertinggi ada pada *Rating* 4.

4.3.2 *Multinomial Naïve Bayes* dengan Kombinasi *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*

Misalkan terdapat *review Editor* terhadap beberapa film pada dua web, yaitu https://montasefilm.com dan http://www.ulasanpilem.com yang ditunjukkan pada Tabel 4.9.

Tabel 4.9 *Review Editor* Terhadap Film

No.	**Dok**	***Review***	***Rating***
1	Latih	Terlalu khayal, tidak begitu sempurna.	2
2		Sangat nyata, rileks sekali.	4
3		Bagus, begitu sempurna, seru, nyata.	4
4		Agak khayal, terlalu sempurna, tampil bagus.	3

5		Cepat bosan, mata jadi merah.	1
6		Tidak berfantasi, begitu sempurna, bagus.	4
7		Tidak rekomen, menyesal amat.	1
8		Mata jenuh, tidak sesuai.	2
9		Bagus sekali, sesuai untuk remaja.	4
10		Terasa ringan, rileks di mata.	5
1	Uji	Bagus amat, sesuai sekali.	?

Kemudian dilakukan tahap *pre-processing*, yaitu *case folding* dan *tokenizing*, *filtering*, dan *stemming*, seperti yang ditunjukkan pada Tabel 4.10, Tabel 4.11, dan Tabel 4.12 berikut ini.

Berdasarkan dengan proses yang ditunjukkan pada Gambar 2.1 dan Gambar 2.2 di atas maka diperoleh Tabel 4.10.

Tabel 4.10 *Case Folding* dan *Tokenizing*

No.	**Dok**	***Review***	***Rating***
1	Latih	terlalu khayal tidak begitu sempurna	2
2		sangat nyata rileks sekali	4
3		bagus begitu sempurna seru nyata	4
4		agak khayal sempurna tampil bagus	3
5		cepat bosan mata jadi merah	1
6		tidak berfantasi begitu sempurna bagus	4
7		tidak rekomen menyesal amat	1
8		mata jenuh tidak sesuai	2
9		bagus sekali sesuai untuk remaja	4
10		terasa ringan rileks di mata	5
1	Uji	bagus amat sesuai sekali	?

Berdasarkan dengan proses yang ditunjukkan pada Gambar 2.3 di atas maka diperoleh Tabel 4.11.

Tabel 4.11 *Filtering*

No.	Dok	*Review*	*Rating*
1	Latih	terlalu khayal tidak begitu sempurna	2
2		sangat nyata rileks sekali	4
3		bagus begitu sempurna seru nyata	4
4		agak khayal terlalu sempurna tampil bagus	3
5		cepat bosan mata merah	1
6		tidak berfantasi begitu sempurna bagus	4
7		tidak rekomen menyesal amat	1
8		mata jenuh tidak sesuai	2
9		bagus sekali sesuai remaja	4
10		terasa ringan rileks mata	5
1	Uji	bagus amat sesuai sekali	?

Berdasarkan dengan proses yang ditunjukkan pada Gambar 2.5 di atas maka diperoleh Tabel 4.12.

Tabel 4.12 *Stemming*

No.	Dok	*Review*	*Rating*
1	Latih	terlalu khayal tidak begitu sempurna	2
2		sangat nyata rileks sekali	4
3		bagus begitu sempurna seru nyata	4
4		agak khayal terlalu sempurna tampil bagus	3
5		cepat bosan mata merah	1
6		tidak fantasi begitu sempurna bagus	4
7		tidak rekomen sesal amat	1
8		mata jenuh tidak sesuai	2
9		bagus sekali sesuai remaja	4
10		terasa ringan rileks mata	5
1	Uji	bagus amat sesuai sekali	?

Tabel 4.12 menunjukkan data latih dan data uji yang kemudian akan dicari jumlah kemunculan *term* seperti yang ditunjukkan pada Tabel 4.13.

Tabel 4.13 Tabel Kemunculan *Term*

Dok	1	2	3	4	5	6	7	8	9	10
terlalu	1	0	0	1	0	0	0	0	0	0
khayal	1	0	0	1	0	0	0	0	0	0
tidak	1	0	0	0	0	1	1	1	0	0
begitu	1	0	1	0	0	1	0	0	0	0
sempurna	1	0	1	1	0	1	0	0	0	0
sangat	0	1	0	0	0	0	0	0	0	0
nyata	0	1	1	0	0	0	0	0	0	0
rileks	0	1	0	0	0	0	0	0	0	1
sekali	0	1	0	0	0	0	0	0	0	1
bagus	0	0	1	1	0	1	0	0	1	0
seru	0	0	1	0	0	0	0	0	0	0
agak	0	0	0	1	0	0	0	0	0	0
tampil	0	0	0	1	0	0	0	0	0	0
cepat	0	0	0	0	1	0	0	0	0	0
bosan	0	0	0	0	1	0	0	0	0	0
mata	0	0	0	0	1	0	0	1	0	1
merah	0	0	0	0	1	0	0	0	0	0
fantasi	0	0	0	0	0	1	0	0	0	0
rekomen	0	0	0	0	0	0	1	0	0	0
sesal	0	0	0	0	0	0	1	0	0	0
amat	0	0	0	0	0	0	1	0	0	0
jenuh	0	0	0	0	0	0	0	1	0	0
sesuai	0	0	0	0	0	0	0	1	0	1
remaja	0	0	0	0	0	0	0	0	1	0
terasa	0	0	0	0	0	0	0	0	0	1
ringan	0	0	0	0	0	0	0	0	0	1
Rating	2	4	4	3	1	4	1	2	4	5

Dari Tabel 4.13 di atas dapat diperoleh Tabel *Contingency* antara *feature* dengan *class* dari masing-masing *Rating*, yang ditunjukkan pada Tabel 4.14, Tabel 4.15, Tabel 4.16, Tabel 4.17, dan Tabel 4.18.

Tabel *Contingency* antara *feature* dengan *class* dari *Rating* 1 ditunjukkan pada Tabel 4.14.

Tabel 4.14 Tabel *Contingency* dari *Rating* 1

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Terlalu	0	2	2
Non Terlalu	2	6	8
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Khayal	0	2	2
Non Khayal	2	6	8
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Tidak	1	3	4
Non Tidak	1	5	6
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Begitu	0	3	3
Non Begitu	2	5	7
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Sempurna	0	4	4
Non Sempurna	2	4	6
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Sangat	0	1	1
Non Sangat	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Nyata	0	2	2
Non Nyata	2	6	8
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Rileks	0	2	2
Non Rileks	2	6	8

Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Sekali	0	2	2
Non Sekali	2	6	8
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Bagus	0	4	4
Non Bagus	2	4	6
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Seru	0	1	1
Non Seru	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Agak	0	1	1
Non Agak	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Tampil	0	1	1
Non Tampil	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Cepat	1	0	1
Non Cepat	1	8	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Bosan	1	0	1
Non Bosan	1	8	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Mata	1	2	3

Non Mata	1	6	7
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Merah	1	0	1
Non Merah	1	8	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Fantasi	0	1	1
Non Fantasi	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Rekomen	1	0	1
Non Rekomen	1	8	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Sesal	1	0	1
Non Sesal	1	8	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Amat	1	0	1
Non Amat	1	8	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Jenuh	0	1	1
Non Jenuh	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Sesuai	0	2	2
Non Sesuai	2	6	8
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total

Remaja	0	1	1
Non Remaja	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Terasa	0	1	1
Non Terasa	2	7	9
Total	2	8	10

Feature/Class	*Rating* 1	Non *Rating* 1	Total
Ringan	0	1	1
Non Ringan	2	7	9
Total	2	8	10

Tabel *Contingency* antara *feature* dengan *class* dari *Rating* 2 ditunjukkan pada Tabel 4.15.

Tabel 4.15 Tabel *Contingency* dari *Rating* 2

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Terlalu	1	1	2
Non Terlalu	1	7	8
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Khayal	1	1	2
Non Khayal	1	7	8
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Tidak	2	2	4
Non Tidak	0	6	6
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Begitu	0	3	3

Non Begitu	2	5	7
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Sempurna	1	3	4
Non Sempurna	1	5	6
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Sangat	0	1	1
Non Sangat	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Nyata	0	2	2
Non Nyata	2	6	8
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Rileks	0	2	2
Non Rileks	2	6	8
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Sekali	0	2	2
Non Sekali	2	6	8
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Bagus	0	4	4
Non Bagus	2	4	6
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Seru	0	1	1
Non Seru	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Agak	0	1	1
Non Agak	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Tampil	0	1	1
Non Tampil	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Cepat	0	1	1
Non Cepat	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Bosan	0	1	1
Non Bosan	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Mata	1	2	3
Non Mata	1	6	7
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Merah	0	1	1

Non Merah	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Fantasi	0	1	1
Non Fantasi	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Rekomen	0	1	1
Non Rekomen	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Sesal	0	1	1
Non Sesal	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Amat	0	1	1
Non Amat	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Jenuh	1	0	1
Non Jenuh	1	8	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Sesuai	1	1	2
Non Sesuai	1	7	8
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Remaja	0	1	1
Non Remaja	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Terasa	0	1	1
Non Terasa	2	7	9
Total	2	8	10

Feature/Class	*Rating* 2	Non *Rating* 2	Total
Ringan	0	1	1
Non Ringan	2	7	9
Total	2	8	10

Tabel *Contingency* antara *feature* dengan *class* dari *Rating* 3 ditunjukkan pada Tabel 4.16.

Tabel 4.16 Tabel *Contingency* dari *Rating* 3

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Terlalu	1	1	2
Non Terlalu	0	8	8
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Khayal	1	1	2
Non Khayal	0	8	8
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Tidak	0	4	4

Non Tidak	1	5	6
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Begitu	0	3	3
Non Begitu	1	6	7
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Sempurna	1	3	4
Non Sempurna	0	6	6
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Sangat	0	1	1
Non Sangat	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Nyata	0	2	2
Non Nyata	1	7	8
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Rileks	0	2	2
Non Rileks	1	7	8
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Sekali	0	2	2
Non Sekali	1	7	8
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Bagus	1	3	4
Non Bagus	0	6	6
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Seru	0	1	1
Non Seru	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Agak	1	0	1
Non Agak	0	9	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Tampil	1	0	1
Non Tampil	0	9	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Cepat	0	1	1
Non Cepat	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Bosan	0	1	1
Non Bosan	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total

Mata	0	3	3
Non Mata	1	6	7
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Merah	0	1	1
Non Merah	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Fantasi	0	1	1
Non Fantasi	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Rekomen	0	1	1
Non Rekomen	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Sesal	0	1	1
Non Sesal	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Amat	0	1	1
Non Amat	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Jenuh	0	1	1
Non Jenuh	1	8	9

Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Sesuai	0	2	2
Non Sesuai	1	7	8
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Remaja	0	1	1
Non Remaja	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Terasa	0	1	1
Non Terasa	1	8	9
Total	1	9	10

Feature/Class	*Rating* 3	Non *Rating* 3	Total
Ringan	0	1	1
Non Ringan	1	8	9
Total	1	9	10

Tabel *Contingency* antara *feature* dengan *class* dari *Rating* 4 ditunjukkan pada Tabel 4.17.

Tabel 4.17 Tabel *Contingency* dari *Rating* 4

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Terlalu	0	2	2
Non Terlalu	4	4	8
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total

Khayal	0	2	2
Non Khayal	4	4	8
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Tidak	1	3	4
Non Tidak	3	3	6
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Begitu	2	1	3
Non Begitu	2	5	7
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Sempurna	2	2	4
Non Sempurna	2	4	6
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Sangat	1	0	1
Non Sangat	3	6	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Nyata	2	0	2
Non Nyata	2	6	8
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Rileks	1	1	2
Non Rileks	3	5	8

Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Sekali	1	1	2
Non Sekali	3	5	8
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Bagus	3	1	4
Non Bagus	1	5	6
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Seru	1	0	1
Non Seru	3	6	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Agak	0	1	1
Non Agak	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Tampil	0	1	1
Non Tampil	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Cepat	0	1	1
Non Cepat	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Bosan	0	1	1
Non Bosan	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Mata	0	3	3
Non Mata	4	3	7
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Merah	0	1	1
Non Merah	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Fantasi	1	0	1
Non Fantasi	3	6	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Rekomen	0	1	1
Non Rekomen	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Sesal	0	1	1
Non Sesal	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Amat	0	1	1

Non Amat	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Jenuh	0	1	1
Non Jenuh	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Sesuai	0	2	2
Non Sesuai	4	4	8
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Remaja	1	0	1
Non Remaja	3	6	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Terasa	0	1	1
Non Terasa	4	5	9
Total	4	6	10

Feature/Class	*Rating* 4	Non *Rating* 4	Total
Ringan	0	1	1
Non Ringan	4	5	9
Total	4	6	10

Tabel *Contingency* antara *feature* dengan *class* dari *Rating* 5 ditunjukkan pada Tabel 4.18.

Tabel 4.18 Tabel *Contingency* dari *Rating* 5

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Terlalu	0	2	2
Non Terlalu	1	7	8
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Khayal	0	2	2
Non Khayal	1	7	8
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Tidak	0	4	4
Non Tidak	1	5	6
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Begitu	0	3	3
Non Begitu	1	6	7
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Sempurna	0	4	4
Non Sempurna	1	5	6
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Sangat	0	1	1
Non Sangat	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total

Nyata	0	2	2
Non Nyata	1	7	8
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Rileks	1	1	2
Non Rileks	0	8	8
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Sekali	1	1	2
Non Sekali	0	8	8
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Bagus	0	4	4
Non Bagus	1	5	6
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Seru	0	1	1
Non Seru	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Agak	0	1	1
Non Agak	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Tampil	0	1	1
Non Tampil	1	8	9

Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Cepat	0	1	1
Non Cepat	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Bosan	0	1	1
Non Bosan	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Mata	1	2	3
Non Mata	0	7	7
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Merah	0	1	1
Non Merah	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Fantasi	0	1	1
Non Fantasi	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Rekomen	0	1	1
Non Rekomen	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Sesal	0	1	1
Non Sesal	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Amat	0	1	1
Non Amat	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Jenuh	0	1	1
Non Jenuh	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Sesuai	1	1	2
Non Sesuai	0	8	8
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Remaja	0	1	1
Non Remaja	1	8	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Terasa	1	0	1
Non Terasa	0	9	9
Total	1	9	10

Feature/Class	*Rating* 5	Non *Rating* 5	Total
Ringan	1	0	1

Non Ringan	0	9	9
Total	1	9	10

Dari Tabel 4.18 diperoleh nilai *feature selection CHI* dengan rumus Persamaan 2.6 yang ditunjukkan oleh Tabel 4.19.

$$x^2(F, C_k) = \frac{N \times ((N_{F,C_k} \times N_{\bar{F},\bar{C_k}}) - (N_{F,\bar{C_k}} \times N_{\bar{F},C_k}))^2}{N_F \times N_{\bar{F}} \times N_{C_k} \times N_{\bar{C_k}}}$$

Tabel 4.19 *Feature Selection CHI*

Kata	Nilai *CHI*				
	***Rating* 1**	***Rating* 2**	***Rating* 3**	***Rating* 4**	***Rating* 5**
terlalu	0,625	1,40625	4,444444	1,666666667	0,277777778
khayal	0,625	1,40625	4,444444	1,666666667	0,277777778
tidak	0,104166667	3,75	0,740741	0,625	0,740740741
begitu	1,071428571	0,47619048	0,47619	1,26984127	0,476190476
sempurna	1,666666667	0,10416667	1,666667	0,277777778	0,740740741
sangat	0,277777778	0,27777778	0,123457	1,666666667	0,12345679
nyata	0,625	0,625	0,277778	3,75	0,277777778
rileks	0,625	0,625	0,277778	0,104166667	4,444444444
sekali	0,625	0,625	0,277778	0,104166667	4,444444444
bagus	1,666666667	1,66666667	1,666667	3,402777778	0,740740741
seru	0,277777778	0,27777778	0,123457	1,666666667	0,12345679
agak	0,277777778	0	10	0,740740741	0,12345679
tampil	0,277777778	0,27777778	10	0,740740741	0,12345679
cepat	4,444444444	0,27777778	0,123457	0,740740741	0,12345679
bosan	4,444444444	0	0,123457	0,740740741	0,12345679
mata	0,476190476	0,47619048	0,47619	2,857142857	2,592592593
merah	4,444444444	0,27777778	0,123457	0,740740741	0,12345679
fantasi	0,277777778	0,27777778	0,123457	1,666666667	0,12345679
rekomen	4,444444444	0,27777778	0,123457	0,740740741	0,12345679
sesal	4,444444444	0,27777778	0,123457	0,740740741	0,12345679
amat	4,444444444	0,27777778	0,123457	0,740740741	0,12345679
jenuh	0,277777778	4,44444444	0,123457	0,740740741	0,12345679
sesuai	0,625	1,40625	0,277778	1,666666667	4,444444444
remaja	0,277777778	0,27777778	0,123457	1,666666667	0,12345679
terasa	0,277777778	0,27777778	0,123457	0,740740741	10
ringan	0,277777778	0,27777778	0,123457	0,740740741	10

Dari Tabel 4.18 diperoleh nilai *feature selection GSS* dengan rumus Persamaan 2.8 yang ditunjukkan oleh Tabel 4.20.

$$GSS(F, C_k) = N_{F,C_k} N_{\overline{F},\overline{C_k}} - N_{F,\overline{C_k}} N_{\overline{F},C_k}$$

Tabel 4.20 *Feature Selection GSS*

Kata	Nilai *GSS*				
	***Rating* 1**	***Rating* 2**	***Rating* 3**	***Rating* 4**	***Rating* 5**
terlalu	-4	6	8	-8	-2
khayal	-4	6	8	-8	-2
tidak	2	12	-4	-6	-4
begitu	-6	4	-3	8	-3
sempurna	-8	2	6	4	-4
sangat	-2	-2	-1	6	-1
nyata	-4	-4	-2	12	-2
rileks	-4	-4	-2	2	8
sekali	-4	-4	-2	2	8
bagus	-8	-8	6	14	-4
seru	-2	-2	-1	6	-1
agak	-2	0	9	-4	-1
tampil	-2	-2	9	-4	-1
cepat	8	-2	-1	-4	-1
bosan	8	0	-1	-4	-1
mata	4	4	-3	-12	7
merah	8	-2	-1	-4	-1
fantasi	-2	-2	-1	6	-1
rekomen	8	-2	-1	-4	-1
sesal	8	-2	-1	-4	-1
amat	8	-2	-1	-4	-1
Jenuh	-2	8	-1	-4	-1
Sesuai	-4	6	-2	-8	8
Remaja	-2	-2	-1	6	-1
Terasa	-2	-2	-1	-4	9
Ringan	-2	-2	-1	-4	9

Dari Tabel 4.19 diperoleh nilai *weighted average CHI* dengan rumus Persamaan 2.7 yang ditunjukkan oleh Tabel 4.21.

$$x^2(F) = \sum_{k=1}^{|C|} \frac{N_{C_k}}{N} x^2(F, C_k)$$

Tabel 4.21 Nilai *CHI*

Kata	*Sum*	*wAvg.*
Terlalu	8,420138889	1,54513889
Khayal	8,420138889	1,54513889
Tidak	5,960648148	1,16898148
Begitu	3,76984127	0,91269841
Sempurna	4,456018519	0,70601852
Sangat	2,469135802	0,80246914
Nyata	5,555555556	1,80555556
Rileks	6,076388889	0,76388889
Sekali	6,076388889	0,76388889
Bagus	9,143518519	2,26851852
Seru	2,469135802	0,80246914
Agak	11,14197531	1,36419753
Tampil	11,41975309	1,41975309
Cepat	5,709876543	1,2654321
Bosan	5,432098765	1,20987654
Mata	6,878306878	1,64021164
Merah	5,709876543	1,2654321
Fantasi	2,469135802	0,80246914
Rekomen	5,709876543	1,2654321
Sesal	5,709876543	1,2654321
Amat	5,709876543	1,2654321
Jenuh	5,709876543	1,2654321
Sesuai	8,420138889	1,54513889
Remaja	2,469135802	0,80246914
Terasa	11,41975309	1,41975309
Ringan	11,41975309	1,41975309

Dari Tabel 4.20 diperoleh nilai maximum *GSS* dengan rumus Persamaan 2.9 yang ditunjukkan oleh Tabel 4.22.

$$GSS(F, C_k) = \max_{k=1}^{|C|} GSS(F, C_k)$$

Tabel 4.22 Nilai *GSS*

Kata	*Max*
Terlalu	8
Khayal	8
Tidak	12
Begitu	8
Sempurna	6
Sangat	6
Nyata	12
Rileks	8
Sekali	8
Bagus	14
Seru	6
Agak	9
Tampil	9
Cepat	8
Bosan	8
Mata	7
Merah	8
Fantasi	6
Rekomen	8
Sesal	8
Amat	8
Jenuh	8
Sesuai	8
Remaja	6
Terasa	9
Ringan	9

Berdasarkan Tabel 4.21 Nilai *CHI* yang diperoleh dapat diurutkan mulai dari terbesar hingga terkecil, yang ditunjukkan pada Tabel 4.23.

Tabel 4.23 Pengurutan *Term* Berdasarkan Nilai *CHI*

Kata	Nilai *CHI*
Bagus	2,268518519
Nyata	1,805555556
Mata	1,64021164
Sesuai	1,545138889

Terlalu	1,545138889
Khayal	1,545138889
Tampil	1,419753086
Terasa	1,419753086
Ringan	1,419753086
Agak	1,364197531
Cepat	1,265432099
Merah	1,265432099
Rekomen	1,265432099
Sesal	1,265432099
Amat	1,265432099
Jenuh	1,265432099
Bosan	1,209876543
Tidak	1,168981481
Begitu	0,912698413
Sangat	0,802469136
Seru	0,802469136
Fantasi	0,802469136
Remaja	0,802469136
Rileks	0,763888889
Sekali	0,763888889
Sempurna	0,706018519

Berdasarkan Tabel 4.22 Nilai *GSS* yang diperoleh dapat diurutkan mulai dari terbesar hingga terkecil, yang ditunjukkan pada Tabel 4.24.

Tabel 4.24 Pengurutan *Term* Berdasarkan Nilai *GSS*

Kata	Nilai *GSS*
bagus	14
tidak	12
nyata	12
agak	9
tampil	9
terasa	9
ringan	9
terlalu	8
khayal	8
begitu	8
rileks	8

sekali	8
cepat	8
bosan	8
merah	8
rekomen	8
Sesal	8
Amat	8
jenuh	8
sesuai	8
mata	7
sempurna	6
sangat	6
seru	6
fantasi	6
remaja	6

Sebagai contoh dari *feature selection CHI* pada Tabel 4.23 sebesar 50% atau yang diberi warna kuning, dapat diambil *term* seperti yang ditunjukkan oleh Tabel 4.25.

Tabel 4.25 *Term* Hasil *Feature Selection CHI*

Kata
Bagus
Nyata
Mata
Sesuai
Terlalu
Khayal
Tampil
Terasa
Ringan
Agak
Cepat
Merah
Rekomen

Sebagai contoh dari *feature selection CHI* pada Tabel 4.24 sebesar 50% atau yang diberi warna kuning, dapat diambil *term* seperti yang ditunjukkan oleh Tabel 4.26.

Tabel 4.26 *Term* Hasil *Feature Selection GSS*

Kata
Bagus
Tidak
Nyata
Agak
Tampil
Terasa
Ringan
Terlalu
Khayal
Begitu
Rileks
Sekali
Cepat

Dari Tabel 4.25 dan Tabel 4.26 dapat dikombinasi, warna kuning artinya yang hanya di salah satu *feature selection*, sedangkan warna merah artinya yang ada di kedua *feature selection*.

Tabel 4.27 *Conditional Probability* Kombinasi *CHI* dan *GSS*

Kata
Bagus
Nyata
Mata
Sesuai
Terlalu
Khayal
Tampil
Terasa
Ringan
Agak
Cepat
Merah
Rekomen
Tidak
Begitu

Rileks
Sekali

Tabel 4.12 menunjukkan data latih dan data uji yang kemudian akan dicari nilai frekuensinya seperti yang ditunjukkan pada Tabel 4.28.

Tabel 4.28 Frekuensi *Term*

Kata	Frekuensi				
	***Rating* 1**	***Rating* 2**	***Rating* 3**	***Rating* 4**	***Rating* 5**
Bagus	0	0	1	3	0
Nyata	0	0	0	2	0
Mata	1	1	0	0	1
sesuai	0	1	0	1	0
Terlalu	0	1	1	0	0
khayal	0	1	1	0	0
tampil	0	0	1	0	0
terasa	0	0	0	0	1
Ringan	0	0	0	0	1
Agak	0	0	1	0	0
Cepat	1	0	0	0	0
merah	1	0	0	0	0
rekomen	1	0	0	0	0
Tidak	1	2	0	1	0
Begitu	0	1	0	2	0
Rileks	0	0	0	1	1
sekali	0	0	0	2	0

Dari Tabel 4.12 diperoleh nilai *Prior* dengan rumus Persamaan 2.4 yang ditunjukkan oleh Tabel 4.29.

$$Prior\text{: } P(C_k) = \frac{N_{C_k}}{N}$$

Tabel 4.29 *Prior Naïve Bayes*

***Prior* (R1)**	***Prior* (R2)**	***Prior* (R3)**	***Prior* (R4)**	***Prior* (R5)**
$\frac{2}{10}$ =0,200	$\frac{2}{10}$ =0,200	$\frac{1}{10}$ =0,100	$\frac{4}{10}$ =0,400	$\frac{1}{10}$ =0,100

Langkah selanjutya mencari *Conditional Probability* dengan rumus Persamaan 2.5 yang ditunjukkan oleh Tabel 4.30.

Conditional Probability: $P(F, C_k) = \frac{count\ (F,C_k)+1}{(\sum_{F \in V} count(F,C_k))+|V|}$

Tabel 4.30 *Conditional Probability Naïve Bayes*

Kata	***Rating* 1**	***Rating* 2**	***Rating* 3**	***Rating* 4**	***Rating* 5**
Bagus	$\frac{0+1}{5+17} =$ 0,04545455	$\frac{0+1}{7+17} =$ 0,04166667	$\frac{1+1}{5+17} =$ 0,09090909	$\frac{3+1}{12+17} =$ 0,13793103	$\frac{0+1}{4+17} =$ 0,04761905
sesuai	$\frac{0+1}{5+17} =$ 0,04545455	$\frac{1+1}{7+17} =$ 0,11764706	$\frac{0+1}{5+17} =$ 0,04545455	$\frac{1+1}{12+17} =$ 0,06896552	$\frac{0+1}{4+17} =$ 0,04761905
Sekali	$\frac{0+1}{5+17} =$ 0,04545455	$\frac{0+1}{7+17} =$ 0,04166667	$\frac{0+1}{5+17} =$ 0,04545455	$\frac{2+1}{12+17} =$ 0,10344828	$\frac{0+1}{4+17} =$ 0,04761905

Langkah selanjutnya adalah mencari nilai *Posterior* dengan rumus seperti pada Persamaan 2.1 kemudian nilai tersebut dibandingkan. Nilai tertinggi merupakan keputusan yang diambil kata masuk pada kelas *Rating*. Nilai *Posterior* dapat dilihat pada Tabel 4.31.

$$P(C_k|F) = \frac{P(C_k) \times P(F|C_k)}{P(F)}$$

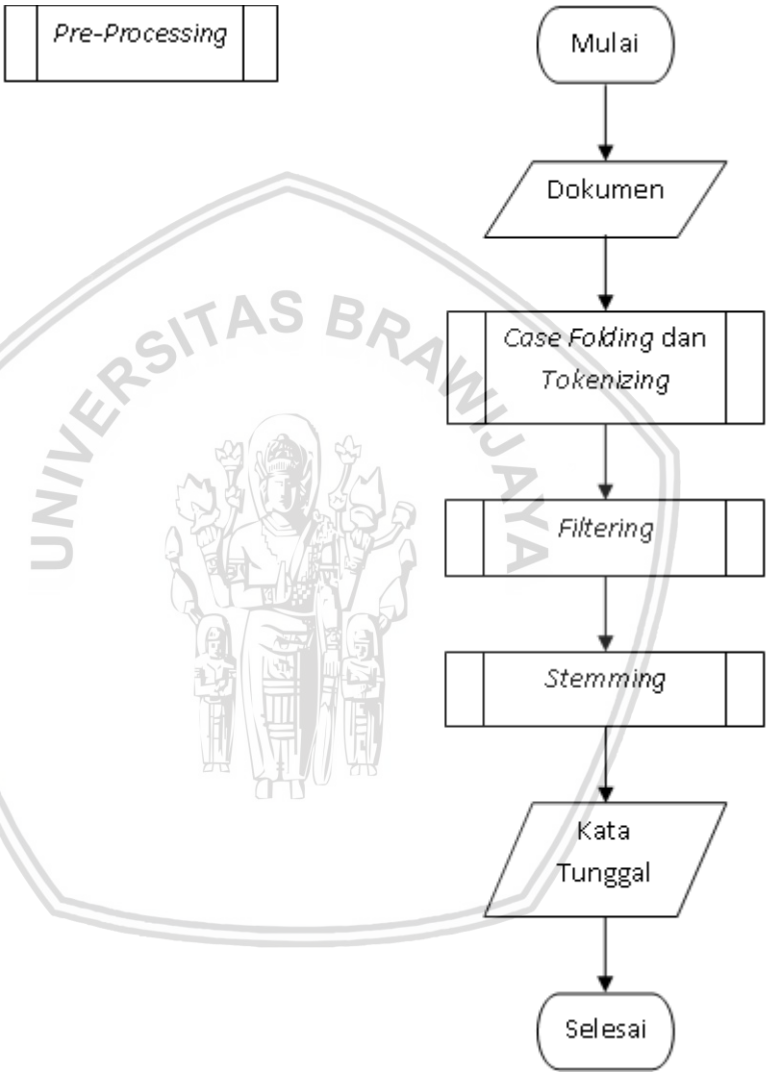
Tabel 4.31 *Posterior Naïve Bayes*

Rating	***Posterior***
Rating 1	0,200 x 0,04545455 x 0,04545455 x 0,04545455 = **0,02727273**
Rating 2	0,200 x 0,04166667 x 0,11764706 x 0,04166667 = **0,04019608**
Rating 3	0,100 x 0,09090909 x 0,04545455 x 0,04545455 = **0,01818182**
Rating 4	0,400 x 0,13793103 x 0,06896552 x 0,10344828 = **0,12413793**
Rating 5	0,100 x 0,04761905 x 0,04761905 x 0,04761905 = **0,01428571**

Dari hasil perhitungan manual tersebut menunjukkan bahwa nilai tertinggi ada pada *Rating* 4.

4.4 *Pre-processing*

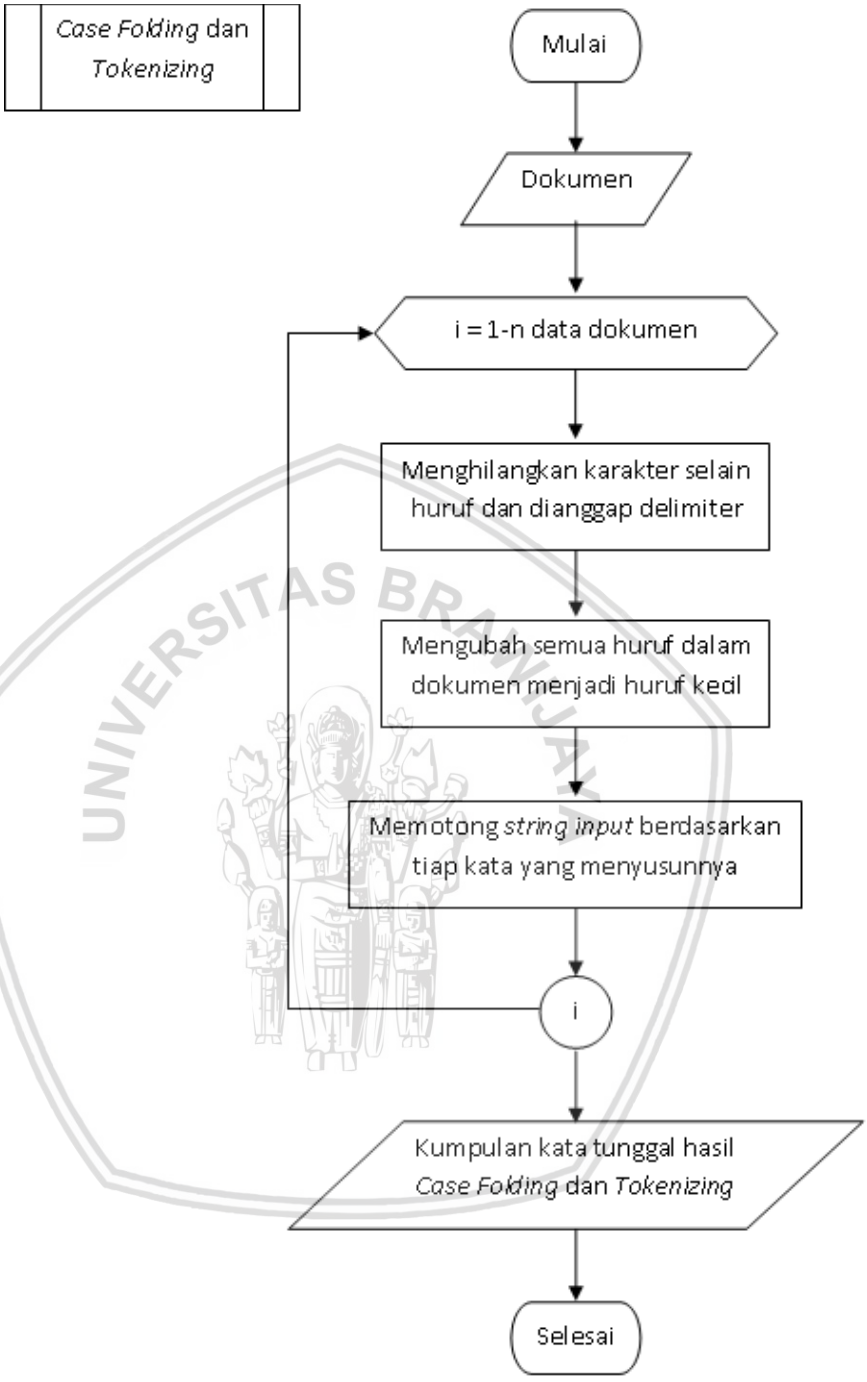
Tahap ini merupakan tahap pengolahan dokumen latih dan uji ke dalam model matematis dimana pada tahapan ini memiliki tahapan seperti *case folding*, *tokenizing*, *filtering*, dan *stemming* untuk mendapatkan inti dari setiap *term* dalam suatu dokumen latih dan uji. Penyelesaian *pre-processing* dijelaskan pada alur Gambar 4.2.



Gambar 4.2 Alur Proses *Pre-processing*

4.4.1 *Case Folding* dan *Tokenizing*

case folding, yaitu sebuah proses awal yang dilakukan untuk mengubah semua huruf kapital menjadi huruf kecil semua, serta menghilangkan tanda baca, angka dan sebuah karakter selain alfabet. Selanjutnya melakukan proses pemenggalan setiap dokumen ke dalam bentuk kata tunggal yaitu *tokenizing*, yang dijelaskan pada alur Gambar 4.3.

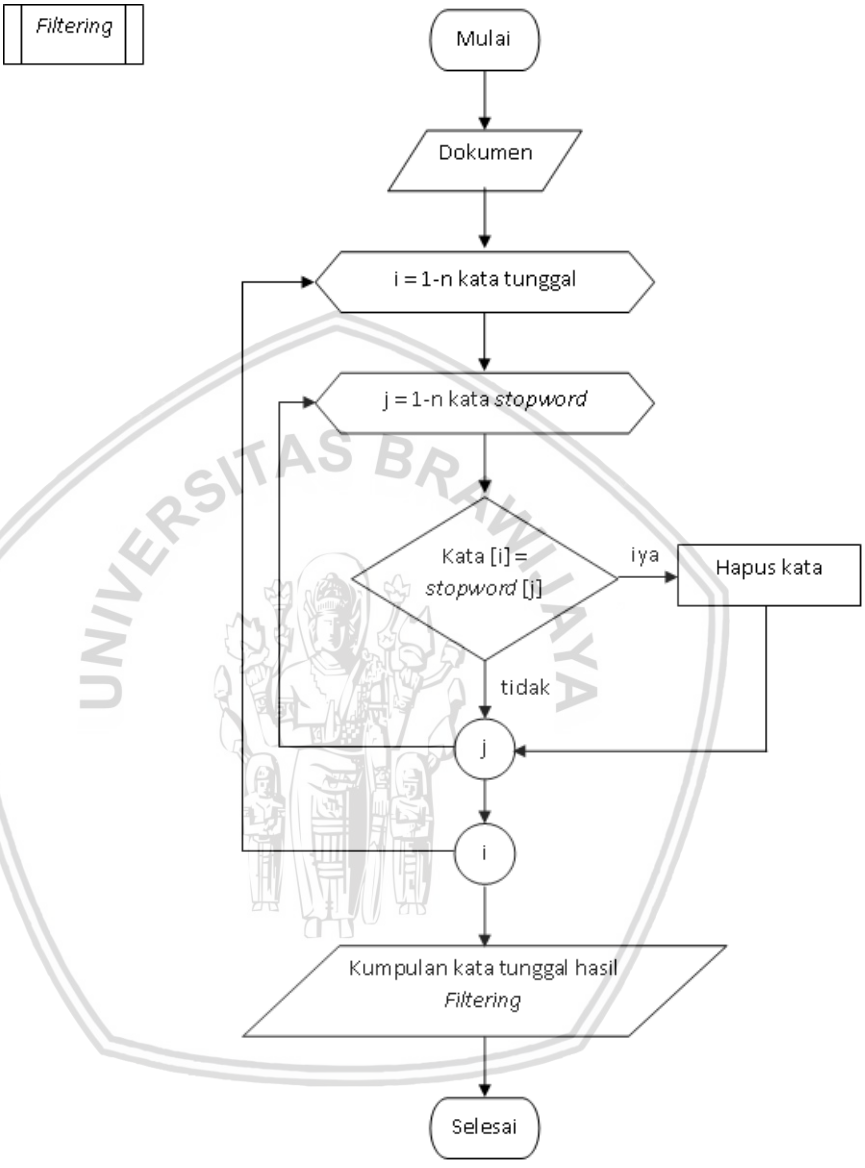


Gambar 4.3 Alur Proses *Case Folding* dan *Tokenizing*

4.4.2 *Filtering*

Dari proses sebelumnya yaitu penghilangan simbol tanda baca dan melakukan *case folding* keluaran dari proses ini akan dilakukan *filtering* atau penyaringan yang hanya mengambil *term* yang mempresentasikan isi dari suatu dokumen dengan mengacu pada kamus *stopword* yang telah ada untuk

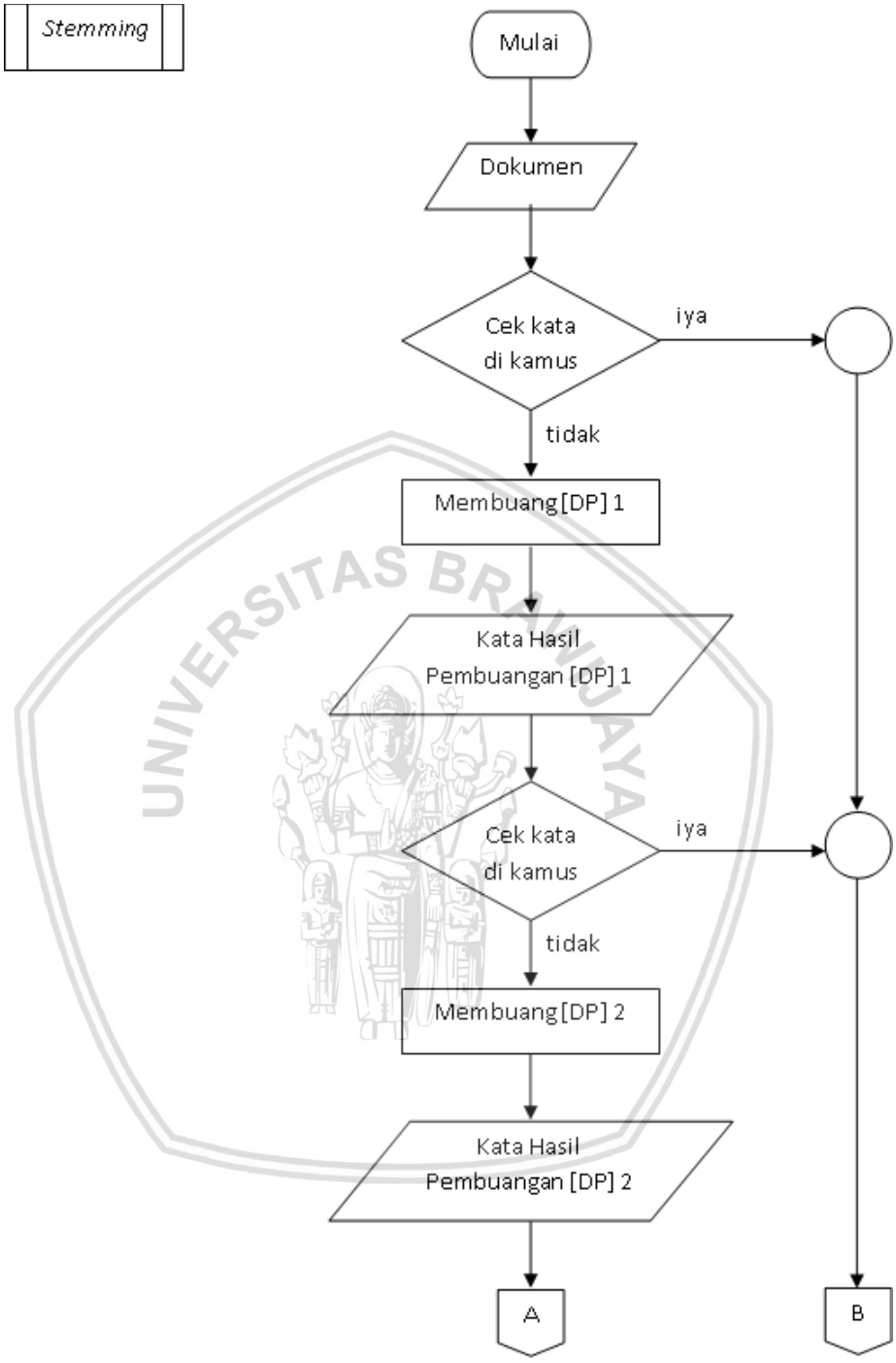
menentukan *term* mana saja yang akan dihilangkan, contoh kata yang sering dihapus seperti dan, apa, itu, yang, dari, akan, di, dll. Penyelesaian *filtering* dijelaskan pada alur proses Gambar. 4.4.

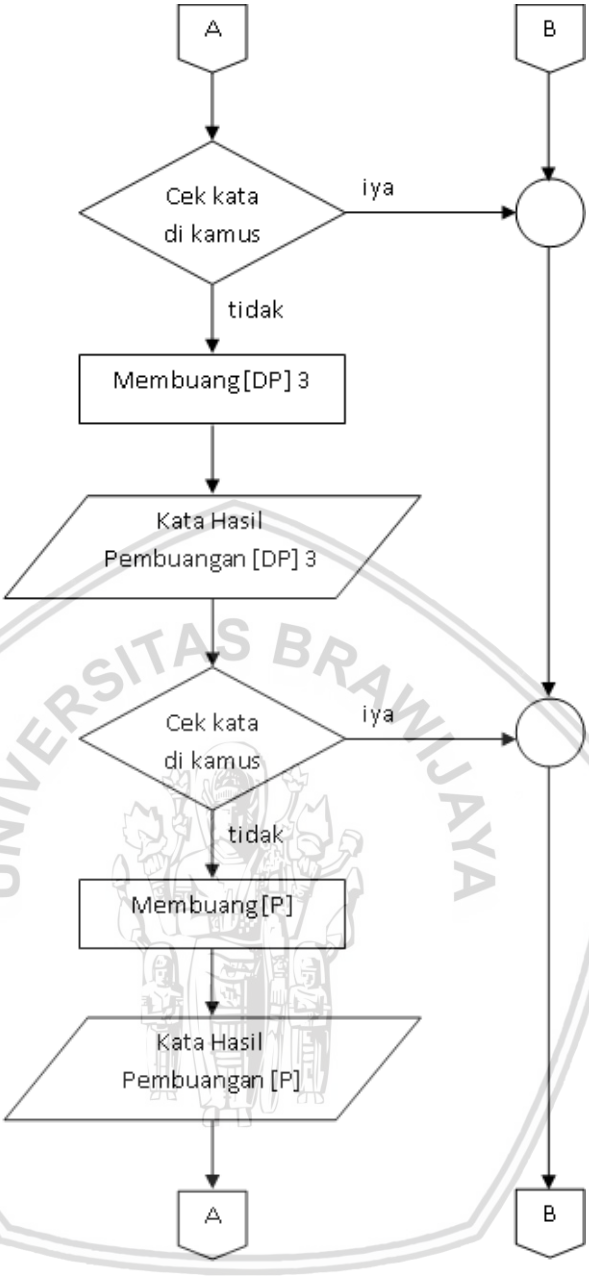


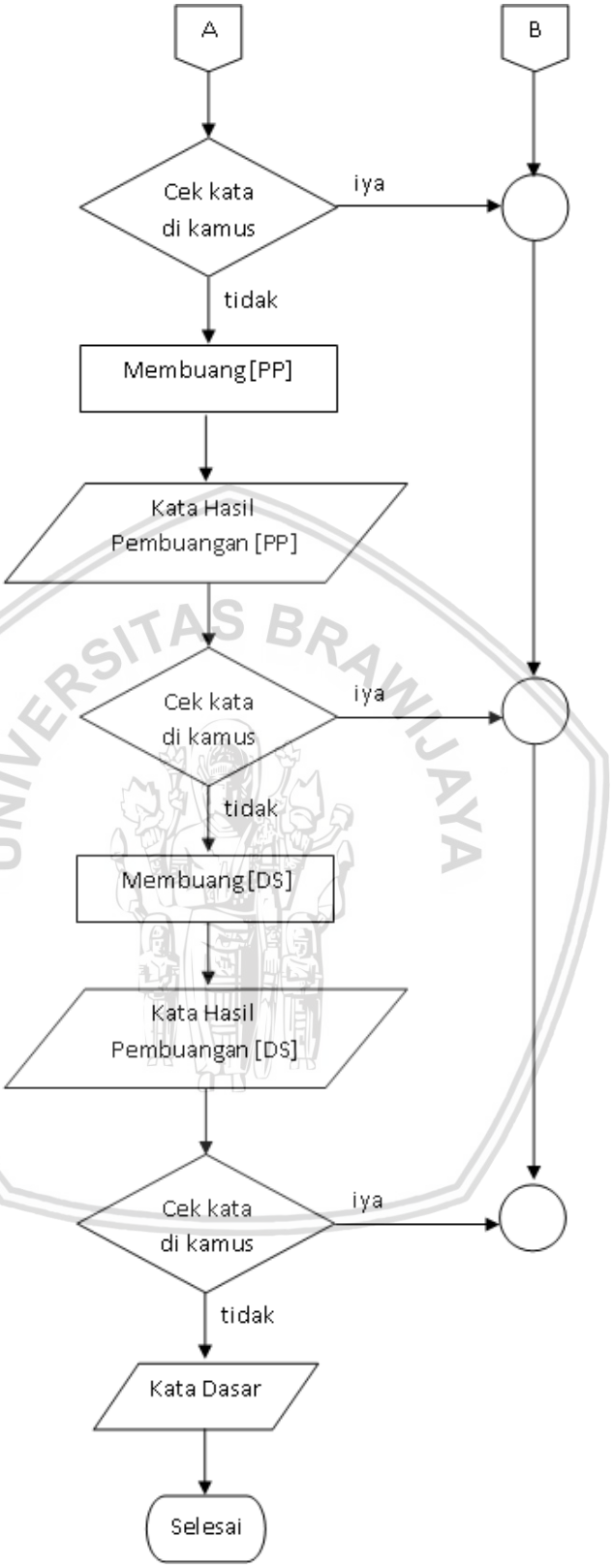
Gambar 4.4 Alur Proses *Filtering*

4.4.3 *Stemming*

Proses *stemming* merupakan tahapan proses yang akan mengambil suatu kata inti/dasar yang diperoleh dari proses *filtering*. Proses ini dilakukan untuk menghilangkan imbuhan-imbuhan seperti awalan (*prefix*) dan akhiran (*suffix*) sehingga akan dihasilkan kata dasar. Penyelesaian *stemming* dijelaskan pada alur proses Gambar 4.5.


Stemming
Mulai
Dokumen
Cek kata di kamus
iya
tidak
Membuang [DP] 1
Kata Hasil Pembuangan [DP] 1
Cek kata di kamus
iya
tidak
Membuang [DP] 2
Kata Hasil Pembuangan [DP] 2
A
B

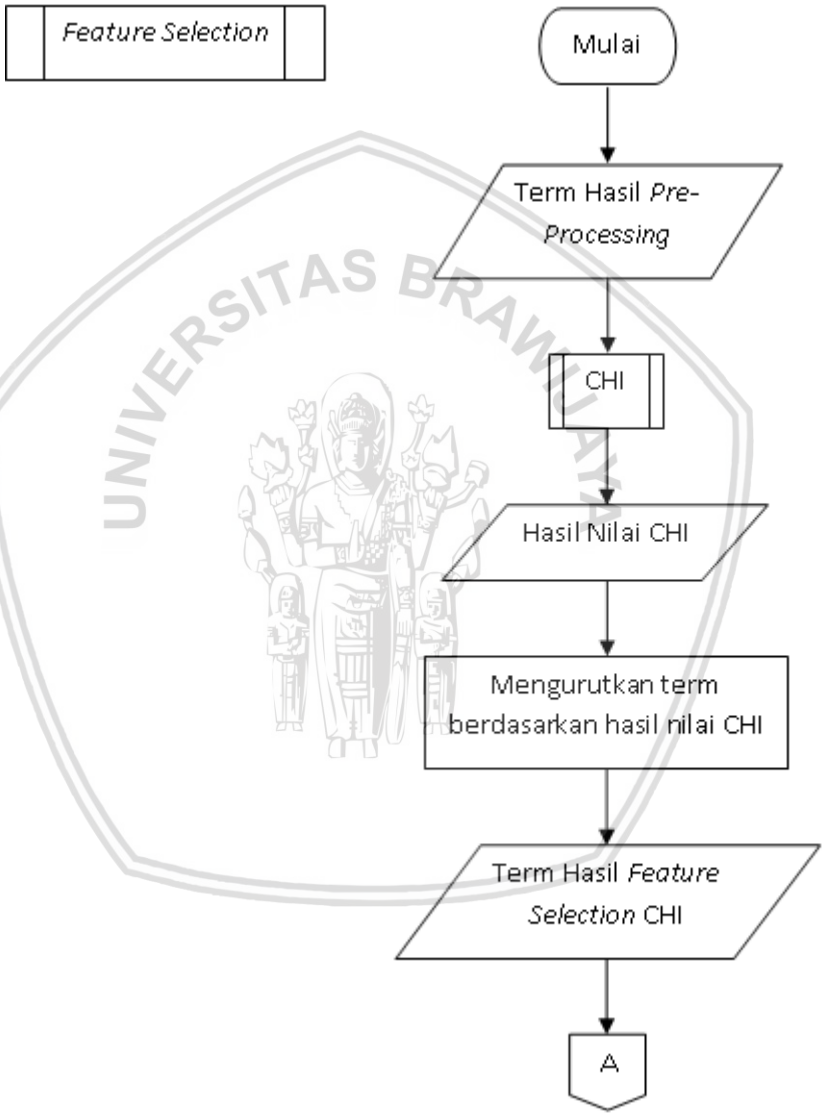

A
B
Cek kata
di kamus
iya
tidak
Membuang[DP] 3
Kata Hasil
Pembuangan [DP] 3
Cek kata
di kamus
iya
tidak
Membuang[P]
Kata Hasil
Pembuangan [P]
A
B



Gambar 4.5 Alur Proses *Stemming*

4.5 *Feature Selection*

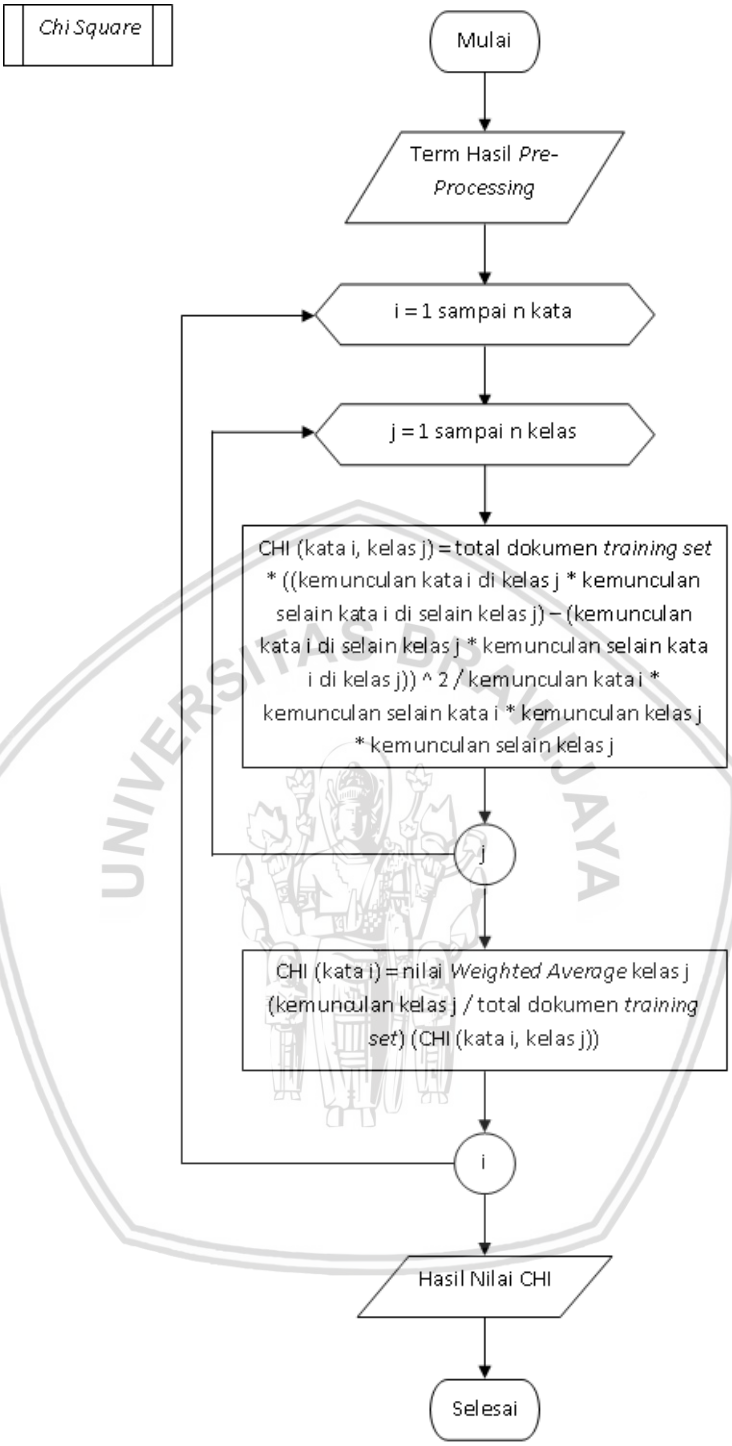
Penerapan *feature selection* pada penelitian ini digunakan untuk mengurangi beberapa *feature* yang dianggap tidak relevan untuk dilakukan klasifikasi. Sehingga *feature* tersebut akan dianggap *noise* dan akan langsung dibuang. *feature selection* yang dipakai pada penelitian ini adalah *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*. Untuk alur penggunaan *feature selection* akan dijelaskan pada diagram alur Gambar 4.6.



Gambar 4.6 *Feature Selection*

4.5.1 *Chi-Square*

Penerapan *Chi-Square* sebagai *feature selection* untuk proses klasifikasi dapat memberikan pengaruh yang lebih baik pada hasil klasifikasi. Dengan mengurangi *noise* atau *feature-feature* yang tidak relevan untuk diklasifikasi maka akan memberikan tingkat keakuratan lebih tinggi pada hasil klasifikasi. Untuk penyelesaian menggunakan *Chi-Square* akan dijelaskan pada diagram alur Gambar 4.7.

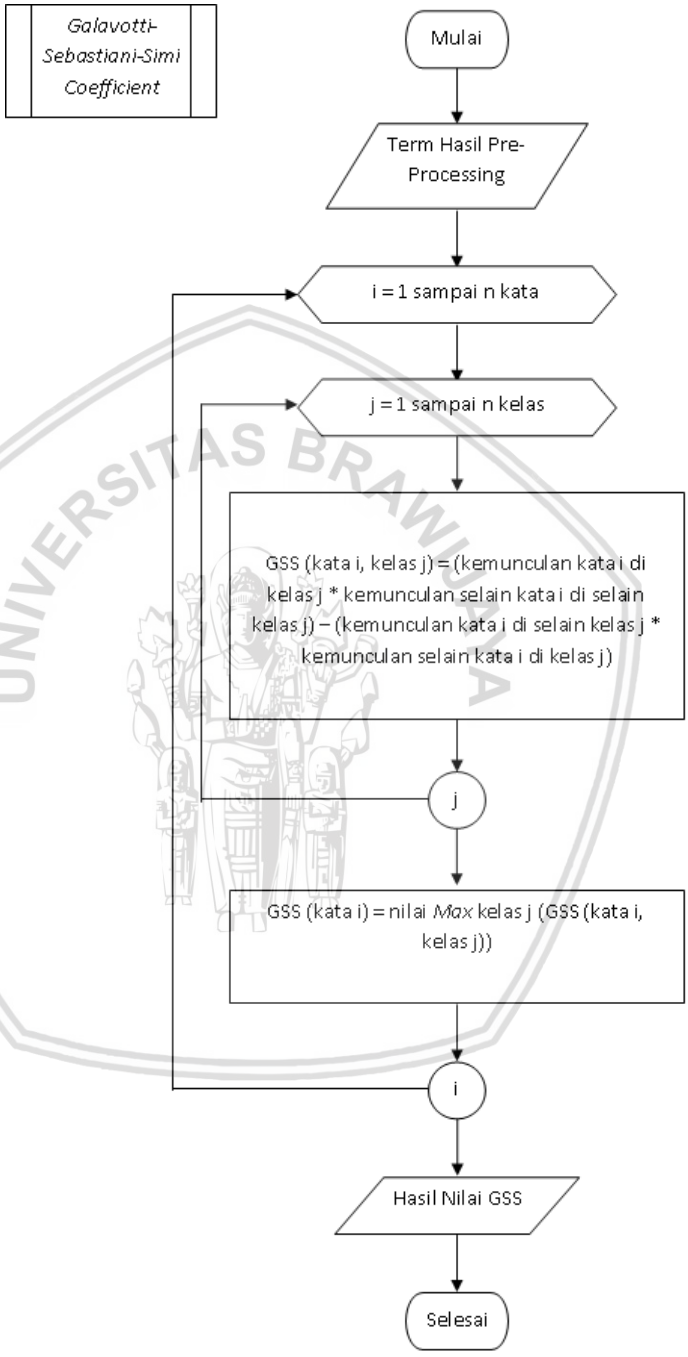


Gambar 4.7 Alur Proses Penyelesaian *Chi-Square*

4.5.2 *Galavotti-Sebastiani-Simi Coefficient*

Penerapan *Galavotti-Sebastiani-Simi Coefficient* sebagai *feature selection* untuk proses klasifikasi dapat memberikan pengaruh yang lebih baik pada hasil klasifikasi. Dengan mengurangi *noise* atau *feature-feature* yang tidak relevan

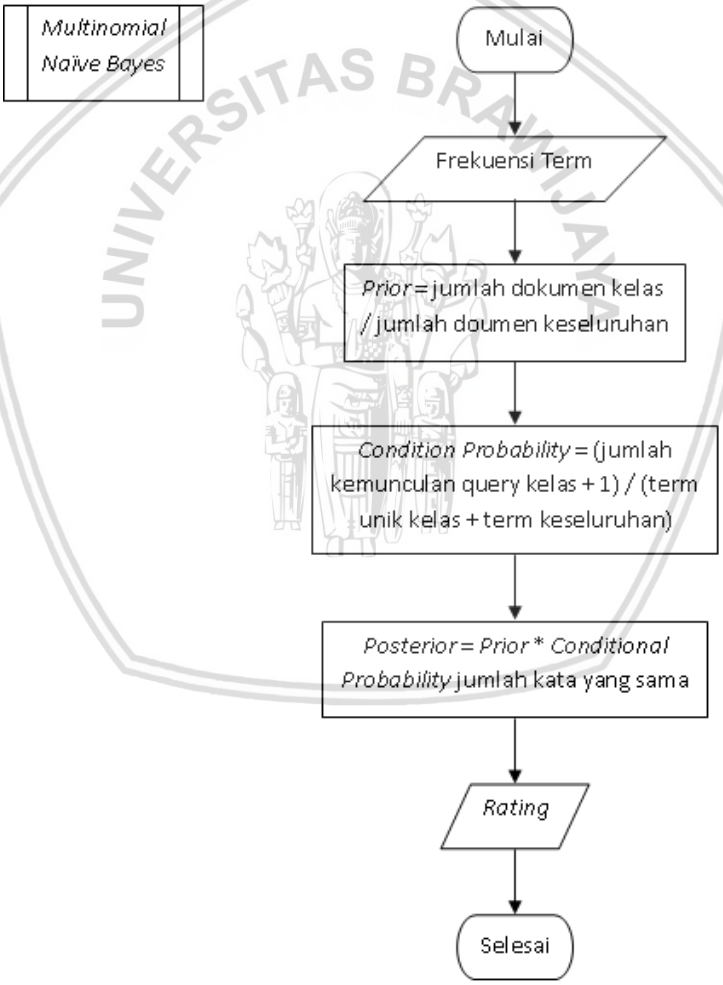
untuk diklasifikasi maka akan memberikan tingkat keakuratan lebih tinggi pada hasil klasifikasi. Untuk penyelesaian menggunakan *Galavotti-Sebastiani-Simi Coefficient* akan dijelaskan pada diagram alur Gambar 4.8.



Gambar 4.8 Alur Proses Penyelesaian *Galavotti-Sebastiani-Simi Coefficient*

4.6 Penyelesaian Metode *Multinomial Naïve Bayes*

Penerapan *Multinomial Naïve Bayes* pada pemrosesan teks, khususnya dalam hal ini klasifikasi dokumen dikarenakan beberapa alasan, seperti efektifitas pengkategorian sebuah teks mempunyai tingkat akurasi yang tinggi. Ada beberapa tahapan dalam *Multinomial Naïve Bayes* diantaranya mencari nilai probabilitas *Prior* yang didapatkan dari probabilitas kategori dokumen positif dan negatif. Likelihood *Conditional Probability* dicari berdasarkan frekuensi kemunculan *term* pada setiap dokumen yang kemudian akan digunakan untuk mencari nilai *Posterior*. Nilai *Posterior* nantinya akan dibandingkan dan dicari nilai tertinggi. Nilai tersebut akan menentukan sebuah dokumen terklasifikasi ke dalam *Rating*. Lebih jelasnya, penyelesaian *Multinomial Naïve Bayes* dijelaskan pada diagram alur Gambar 4.8.

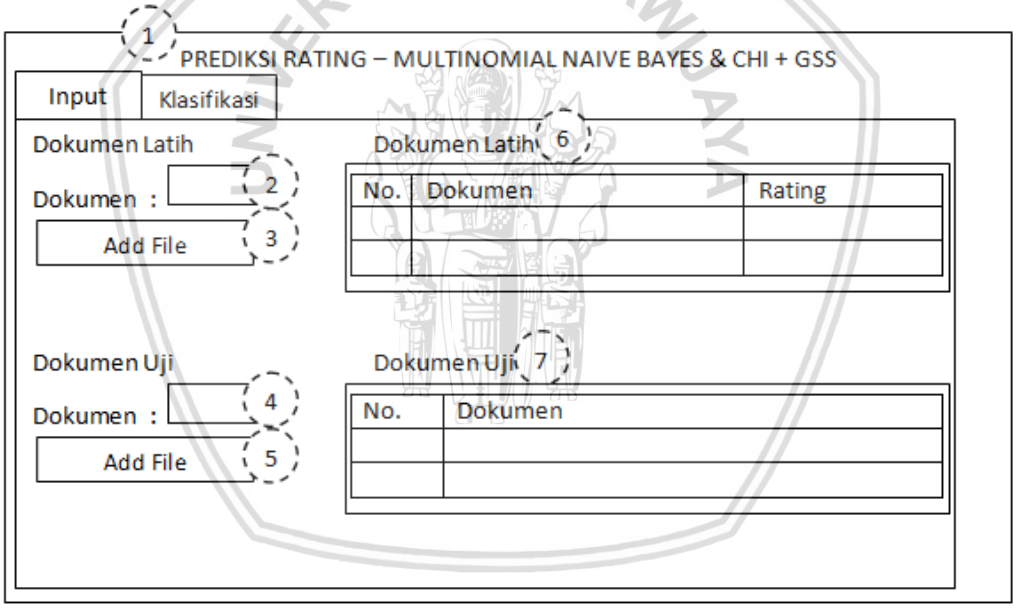


Gambar 4.9 Alur Proses *Multinomial Naïve Bayes*

4.7 Perancangan Antarmuka

Perancangan sistem merupakan tahapan di mana perancang mulai merancang suatu sistem yang mampu memenuhi semua kebutuhan dari aplikasi. Teori dari pustaka yang digunakan dan data sampel yang telah dikumpulkan akan digabungkan untuk pengimplementasian pengembangan aplikasi *Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* untuk penentuan kategori *review* film. Aplikasi memiliki bagian sistem yang memproses data input pengguna untuk menghasilkan output yang sesuai yaitu kategori *review* film.

Antarmuka sistem merupakan mekanisme yang digunakan oleh pengguna dan sistem untuk saling berkomunikasi. Pada tahap ini, antarmuka sistem yang akan dibangun terdiri dari satu *window*, yaitu halaman utama yang memuat perintah untuk menambah, menghapus, dan mengubah data baik data latih maupun data uji. Adapun rancangan detail dari halaman utama yang ditunjukkan pada Gambar 4.10.

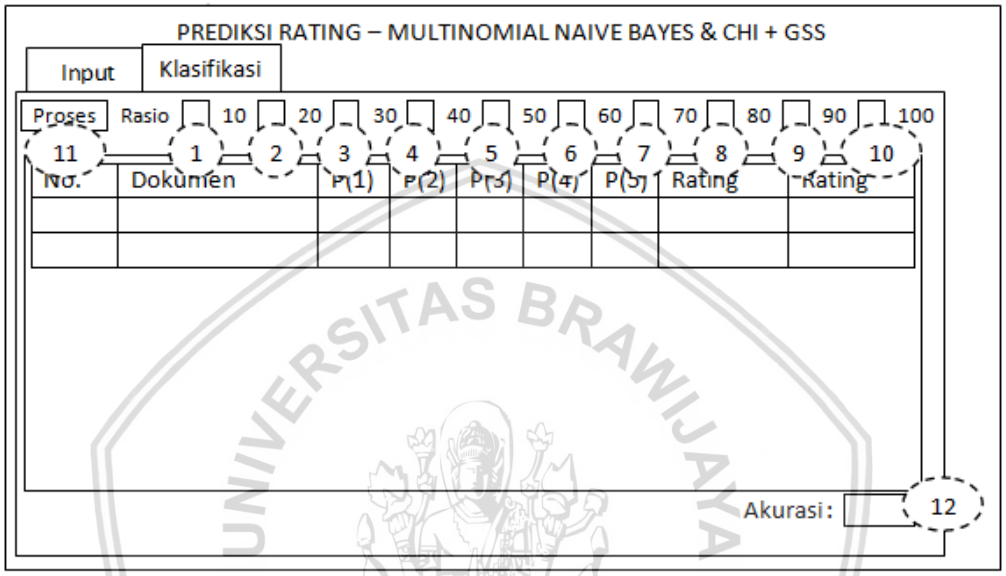


Gambar 4.10 Rancangan Halaman Utama

Keterangan:

1. Judul.
2. Teks masukan dokumen latih.
3. Tombol *Add File* untuk mengolah file yang berisi data latih.
4. Teks masukan dokumen uji.
5. Tomboh *Add File* untuk mengolah file yang berisi data uji.
6. Tabel data latih.
7. Tabel data uji.

Antarmuka selanjutnya adalah halaman *pre-processing* dan klasifikasi dengan Metode *Multinomial Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* yang memuat perintah untuk mengklasifikasikan sebuah dokumen ke dalam kelas atau *Rating*. Adapun rancangan detail dari halaman utama yang ditunjukkan pada Gambar 4.11.



Gambar 4.11 Rancangan Halaman Klasifikasi

Keterangan:

1. *Check Box* 10% untuk memilih prosentase *feature* 10%.
2. *Check Box* 20% untuk memilih prosentase *feature* 20%.
3. *Check Box* 35% untuk memilih prosentase *feature* 30%.
4. *Check Box* 40% untuk memilih prosentase *feature* 40%.
5. *Check Box* 50% untuk memilih prosentase *feature* 50%.
6. *Check Box* 60% untuk memilih prosentase *feature* 60%.
7. *Check Box* 70% untuk memilih prosentase *feature* 70%.
8. *Check Box* 80% untuk memilih prosentase *feature* 80%.
9. *Check Box* 90% untuk memilih prosentase *feature* 90%.
10. *Check Box* 100% untuk memilih prosentase *feature* 100%.
11. Tombol Proses untuk melakukan proses klasifikasi.
12. Teks keluaran hasil akurasi.

4.8 Perancangan Pengujian

Pengujian sistem yang dilakukan berkaitan dengan pengujian validasi sistem. Tahap ini berfungsi untuk memastikan apakah sistem yang dibuat dapat memperbaik permasalahan sebelumnya dan sejauh mana sistem dapat memepengaruhi permasalah yang terjadi. Analisis sistem aplikasi dilakukan dengan membandingkan antara data sampel yang telah dikumpulkan sebelumnya dengan hasil setelah sistem aplikasi diterapkan.

4.9 Penarikan Kesimpulan

Pengambilan kesimpulan merupakan metode yang diilakukan setelah semua tahapan perancangan, implementasi, dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan yang diambil dilakukan berdasarkan kesesuaian teori dan praktik. Kesimpulan dibuat bertujuan untuk menjawab rumusan yang disusun pada BAB I. Selain kesimpulan, bagian akhir dari penulisan ini adalah saran yang bertujuan untuk memperbaiki kesalahan yang terjadi dan menyempurnakan penulisan yang ditujukan pada penulisan dan perancangan aplikasi berikutnya.

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem berdasarkan metodologi penelitian serta analisis dan perancangan yang telah dijelaskan pada bab sebelumnya.

5.1 Batasan Implementasi

Batasan implementasi merupakan batasan proses yang dapat dilakukan oleh sistem berdasarkan perancangan yang telah diuraikan pada bab sebelumnya. Batasan implementasi bertujuan untuk membuat sistem sesuai dengan ruang lingkup yang jelas dan tidak keluar dari tujuan utama dari sistem. Adapun beberapa batasan implementasi sistem dalam penelitian ini sebagai berikut:

1. Prediksi *Rating* pada *review* film dirancang dan dijalankan mengunakan aplikasi dekstop berbahasa Java.
2. Metode penyelesaian masalah yang digunakan adalah *Multinomial Naïve Bayes Classifier*, *Chi-Square* , dan *Galavotti-Sebastiani-Simi Coefficient*.
3. Data yang digunakan sebagai data latih dan data uji merupakan komentar atau *review* pada film di https://montasefilm.com dan http://www.ulasanpilem.com yang berbentuk teks.
4. Keluaran yang dikeluarkan berupa hasil prediksi *Rating* yaitu *Rating* 1, *Rating* 2, *Rating* 3, *Rating* 4, dan *Rating* 5.
5. Penentuan *Rating* berdasarkan kamus kata dengan menghitung frekuensi kemunculan kata.

5.2 *Text Mining*

Implementasi aplikasi menjelaskan bagaimana tahapan aplikasi prediksi *Rating* pada *review* film menggunakan metode *Multinomial Naïve Bayes Classifier*, *Chi-Square* , dan *Galavotti-Sebastiani-Simi Coefficient*. Tahapan tersebut akan dijelaskan berdasarkan alur kerja sistem yang dijelaskan pada bab sebelumnya. Penjelasan tahapan akan melibatkan beberapa potongan kode program yang dijelaskan oleh aplikasi.

Aplikasi prediksi *Rating* pada *review* film menggunakan metode *Multinomial Naïve Bayes Classifier*, *Chi-Square* , dan *Galavotti-Sebastiani-Simi Coefficient* terdiri dari beberapa proses utama di antaranya yaitu *pre-processing* meliputi *case folding*, *tokenizing*, *filtering*, dan *stemming*. Sedangkan klasifikasi terdiri dari *Naïve Bayes Classifier*, *Chi-Square* , dan *Galavotti-Sebastiani-Simi Coefficient*. Data diolah berdasarkan proses tersebut sehingga aplikasi dapat melakukan klasifikasi prediksi *Rating* berupa *Rating* 1 sampai *Rating* 5. Adapun beberapa fungsi yang digunakan pada aplikasi seperti pada Tabel 5.1.

Tabel 5.1 Daftar Fungsi Aplikasi Prediksi *Rating* Pada Film

No	Proses	Nama Fungsi	Keterangan
1	*pre-processing*	*caseFolding()*	Fungsi untuk mengubah suatu kata atau kalimat menjadi huruf kecil, serta menghapus angka dan tanda baca.
		tokenizing()	Fungsi untuk memotong kalimat menjadi sebuah kata tunggal.
		filtering()	Fungsi untuk menghilangkan kata yang tidak penting berdasarkan kamus.
		stemming()	Fungsi untuk mencari kata dasar dari suatu kata.
2	*Chi-Square*	*sortByCHIValue()*	Fungsi untuk menghitung nilai *CHI* suatu *term* pada setiap kelas dan mendapatkan nilai *CHI* bobot rata-rata untuk suatu *term* di antara nilai *CHI* yang sudah didapatkan pada setiap kelas.
3	*Galavotti-Sebastiani-Simi Coefficient*	*sortByGssValue()*	Fungsi untuk menghitung nilai *GSS* suatu *term* pada setiap kelas dan mendapatkan nilai *GSS* berat rata-rata untuk suatu *term* di antara nilai *GSS* yang sudah didapatkan pada setiap kelas.
4	Kombinasi	*kombinasi()*	Fungsi untuk mengkombinasi kedua *feature selection*.
5	Klasifikasi	*naiveBayes()*	Fungsi yang digunakan untuk menghitung nilai *Prior* pada setiap kelas, menghitung nilai *Conditional Probability* suatu *term* pada setiap kelas, dan menghitung nilai *Posterior* pada setiap kelas.
		Kemunculanterm()	Fungsi yang digunakan untuk mencari dan menghitung kemunculan *term* sebuah kata pada setiap kelas.
		frekuensi()	Fungsi yang digunakan untuk mencari dan menghitung

			frekuensi sebuah kata pada setiap kelas.

5.3 *Pre-processing*

pre-processing merupakan teknik data mining yang melibatkan perubahan data mentah menjadi sebuah format yang terstruktur dan dimengerti. Data mentah seringkali tidak lengkap, tidak konsisten, dan mungkin mengandung banyak kesalahan. Teknik *pre-processing* terbukti dapat menyelesaikan masalah tersebut.

5.3.1 Proses *Case Folding*

Langkah pertama dari teknik *pre-processing* adalah *case folding*, yaitu mengubah semua huruf kapital menjadi huruf kecil semua, serta menghilangkan tanda baca, angka, dan karakter selain alfabet, seperti yang ditunjukkan pada potongan Kode Program 5.1.

Kode Program 5.1 Implementasi Kode Proses *Case Folding*

```
1       public static String caseFolding(String kalimatAsli) {
            //System.out.println("kalimatAsli = " + kalimatAsli);
2           String kalimatHasil = "";
3
4           kalimatHasil = kalimatAsli.replaceAll("[0-9]", "");
5           kalimatHasil  =  kalimatHasil.replaceAll("[^a-zA-Z]",
6       " ");
            kalimatHasil = kalimatHasil.toLowerCase();
7           return kalimatHasil;
8         }
9
```

Pembahasan:

Baris 1 nama method untuk *case folding*.

Baris 2-7 isi dari method *case folding* yang berfungsi untuk menghilangkan angka, mengubah huruf menjadi huruf kecil.

Baris 8 nilai kembalian dari *case folding*.

5.3.2 Proses *Tokenizing*

Langkah selanjutnya dari teknik *pre-processing* adalah memotong kalimat menjadi sebuah kata tunggal yaitu *tokenizing*. Potongan kode program proses *tokenizing* dapat dilihat pada Kode Program 5.2.

Kode Program 5.2 Implementasi Kode Proses *Tokenizing*

```
1       public static String[] tokenizing(String kalimatAsli) {
2               String[] kata = kalimatAsli.split("\\s+");
3               return kata;
4           }
```

Pembahasan:

Baris 1 nama method untuk *tokenizing*.

Baris 2 isi dari method *tokenizing* yang berfungsi untuk memotong kalimat menjadi sebuah kata tunggal.

Baris 3 nilai kembalian dari tokenizng.

5.3.3 Proses *Filtering*

Proses *filtering* adalah proses penyaringan sebuah kata dari hasil proses *tokenizing*. Teknik yang digunakan adalah *stopword* yaitu membuang kata-kata yang tidak diperlukan dengan mencocokan kata berdasarkan kamus *stopword*. Kata yang dibuang merupakan kata penghubung seperti “dan”, “dengan”, “atau” dan lain sebagainya. Potongan kode program proses *filtering* dapat dilihat pada Kode Program 5.3.

Kode Program 5.3 Implementasi Kode Proses *Filtering*

```
1       public static String[] filtering(String[] kataAwal) throws
        FileNotFoundException, IOException {
2              //System.out.println("MASUK FILTERING");
3               ArrayList<String> termUnik = new ArrayList<>();
4              //System.out.println(Arrays.deepToString(kataAwal));
5               for (int i = 0; i < kataAwal.length; i++) {
6                   boolean kt = stopList(kataAwal[i]);
7                  //System.out.println("kt: "+kt);
8                   if (!kt) {
9                       termUnik.add(kataAwal[i]);
10                  }
11              }
12
13              String[] kataUnik = new String[termUnik.size()];
14              for (int i = 0; i < kataUnik.length; i++) {
15                 //System.out.println("termUnik.get("+i+"):       "
        +termUnik.get(i));
16                  kataUnik[i] = termUnik.get(i);
17              }
18             //System.out.println("kata           unik:           "
        +Arrays.deepToString(kataUnik));
19              return kataUnik;
20          }
21
22          public static boolean stopList(String kata) throws
        FileNotFoundException, IOException {
23              boolean ada = false;
24
25              ArrayList<String> dataStopList = new ArrayList<>();
```

```
26          //ArrayList<String>     dataNoStopList     =     new
            ArrayList<>();
27          String[] stoplist;
28          BufferedReader   br2   =   new   BufferedReader(new
            FileReader("C:
            \\Users\\SAMSUNG\\Documents\\NetBeansProjects\\marta_skripsi
            \\src\\Stoplist Bahasa Indonesia.txt"));
29
30          try {
31              StringBuilder sb2 = new StringBuilder();
32              String line = br2.readLine();
33              while (line != null) {
34                  sb2.append(line);
35                  sb2.append(System.lineSeparator());
36                  line = br2.readLine();
37                  dataStopList.add(line);
38              }
39
40             //stoplist = sb2.toString().split("[\n]");
41              stoplist = new String[dataStopList.size()];
42              for (int j = 0; j < stoplist.length; j++) {
43                  stoplist[j] = dataStopList.get(j);
44                 //System.out.print(stoplist[j] +" ");
45              }
46          } finally {
47              br2.close();
48          }
49
50          //System.out.println("kata: " + kata);
51          for (int i = 0; i < stoplist.length; i++) {
52             //System.out.println(stoplist[i]);
53              if (kata.equalsIgnoreCase(stoplist[i])) {
54                  ada = true;
55                  break;
56              } else {
57                  ada = false;
58              }
59          }
60          return ada;
61      }
```

Pembahasan:

Baris 1	nama method untuk *filtering*.
Baris 2-18	isi dari method *filtering* yang berfungsi untuk mengambil kata unik dari hasil *tokenizing*.
Baris 19	nilai kembalian dari *case folding*.
Baris 22	nama method untuk stoplist.
Baris 23-59	Isi dari method stoplist yang berfungsin untuk membuang kata yang ada pada stoplist.
Baris 60	Nilai kembalian dari stoplist.

5.3.4 Proses *Stemming*

Proses *stemming* merupakan tahapan proses yang akan mengambil suatu kata inti/dasar yang diperoleh dari proses *filtering*. Proses ini dilakukan untuk menghilangkan imbuhan-imbuhan seperti awalan (*prefix*) dan akhiran (*suffix*) sehingga akan dihasilkan kata dasar. Potongan kode program proses *stemming* dapat dilihat pada Kode Program 5.4.

Kode Program 5.4 Implementasi Kode Proses *Stemming*

```
1   public static String stemming(String kata) throws
    FileNotFoundException, IOException, Exception {
2           List dictionary = new
    Dictionary().read("src/IndonesianStemmer/dictionary/dictiona
    ries.properties").getDictionaryData();
3           Stemmer stemmer = new Stemmer(dictionary);
4          //System.out.println("kata: " +kata);
5           if (kata.length() < 5) {
6               return kata;
7           } else {
8               String kata_dasar = stemmer.getRootWord(kata);
9              //System.out.println("kata dasar: " +kata_dasar);
               if (kata_dasar == null) {
10                 return kata;
11              } else {
12                 return kata_dasar;
13              }
14          }
15      }
16
```

Pembahasan:

Baris 1 nama method untuk *stemming*.

Baris 2-12 isi dari method *stemming* yang berfungsi untuk menghilangkan imbuhan, baik awalan maupun akhiran dengan mencocokkan dengan kata baku.

Baris 13 nilai kembalian dari *stemming*.

5.4 Proses Prediksi *Rating*

Pada proses ini terdiri dari dua proses yaitu perhitungan klasifikasi dengan menggunakan *Multinomial Naïve Bayes Classifier* dan juga proses *feature selection* dengan menggunakan *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* yang nantinya akan dibandingkan dan dievaluasi.

5.4.1 Proses Klasifikasi Menggunakan *Multinomial Naïve Bayes Classifier*

Proses klasifikasi menggunakan *Multinomial Naïve Bayes Classifier* ini adalah melakukan klasifikasi pada *review* film pada situs web https://montasefilm.com dan http://www.ulasanpilem.com yang akan diuji. Hasil akhir dari proses ini adalah berupa klasifikasi *Rating* dari data *review* film pada situs web https://montasefilm.com dan http://www.ulasanpilem.com yang nantinya akan diuji ke dalam kelas *Rating* 1 sampai *Rating* 5. Potongan kode program fungsi klasifikasi *Multinomial Naïve Bayes Classifier* dapat dilihat pada Kode Program 5.5.

Kode Program 5.5 Implementasi Kode *Multinomial Naïve Bayes Classifier*

```
1   public static double[] naiveBayes(String[][] d, int[][] f,
    int[] index) {
2          //hitung Prior
3           double[] sum = new double[5];
4           double sum_all = 0.0;
5           for (int i = 0; i < sum.length; i++) {
6               for (int j = 0; j < d.length; j++) {
7                   if (d[j][2].equals(String.valueOf(i + 1))) {
8                       sum[i]++;
9                   }
10              }
11              sum_all += sum[i];
12          }
13
14          for (int i = 0; i < 5; i++) {
15              Prior[i] = sum[i]/sum_all;
16              System.out.println("Prior  "  +  i  +  ":  "  +
    Prior[i]);
17          }
18
19          double[] sumf = new double[5];
20          for (int i = 0; i < 5; i++) {
21              for (int j = 0; j < f.length; j++) {
22                  sumf[i] += f[j][i];
23              }
24          }
25         //hitung Conditional Probability
26          double[][] cp = new double[index.length][5];
27          for (int i = 0; i < index.length; i++) {
28              for (int j = 0; j < 5; j++) {
29                  if (index[i] != -1) {
30
    //System.out.println("(f[index["+i+"]]["+j+"]              =
    "+f[index[i]][j] +")/(sumf["+j+"] = " +sumf[j] +" f.length)
    = " +f.length);
31                      cp[i][j] = (f[index[i]][j] + 1)/(sumf[j]
    + f.length);
32                  } else {
33                      cp[i][j] = 0;
34                  }
35                 //System.out.println("cp["+i+"]["+j+"]   =   "
    +cp[i][j]);
36              }
37          }
38
39         //hitung Posterior
40          double[] Posterior = new double[5];
```

```
41          for (int i = 0; i < 5; i++) {
42              for (int j = 0; j < index.length; j++) {
43                  Posterior[i] += cp[j][i];
44              }
45              Posterior[i] *= Prior[i];
46          }
47
48          return Posterior;
49      }
```

Pembahasan:

Baris 1	nama method untuk naive bayes.
Baris 2-13	isi dari method naive bayes yang berfungsi untuk menghitung *Prior*.
Baris 14-24	isi dari method naive bayes yang berfungsi untuk menghitung *Conditional Probability*
Baris 25-46	isi dari method naive bayes yang berfungsi untuk menghitung *Posterior*.
Baris 47	nilai kembalian dari *Posterior*.

5.4.2 Proses *Feature Selection* Menggunakan *Chi-Square* (*CHI*)

Pada proses *feature selection* yaitu dengan menggunakan *Chi-Square* adalah proses perhitungan nilai *CHI* untuk suatu *term* tertentu pada masing-masing kelas berdasarkan frekuensi kemunculan *term*/kata pada data latih. Setelah dilakukan perhitungan nilai *CHI* pada setiap kelas maka akan dipilih nilai *CHI* bobot rata-rata untuk dijadikan nilai *CHI* untuk *term* tersebut. Proses tersebut dilakukan pada semua *term* hasil dari *pre-processing* pada setiap *review*. Setelah semua *term* mendapatkan nilai *CHI* selanjutnya akan dilakukan sorting berdasarkan nilai *CHI* pada masing-masing *term*. Hasil akhir dari proses ini adalah berupa susunan *term* yang nantinya akan diprioritaskan dalam proses klasifikasi. Potongan kode program untuk proses *feature selection* dengan menggunakan *Chi-Square* dapat dilihat pada Kode Program 5.6.

Kode Program 5.6 Implementasi Kode *Chi-Square* (*CHI*)

```
1   public static String[] sortByChiValue(List<ArrayList<int[]>>
    cont, String[] term, int feature) {
2           double[][] CHI = new double[cont.size()][5];
3          //System.out.println("NILAI CHI");
4           for (int i = 0; i < cont.size(); i++) {
5               for (int j = 0; j < cont.get(i).size(); j++) {
6                   CHI[i][j]    =    (cont.get(i).get(j)[0]    +
    cont.get(i).get(j)[1]      +      cont.get(i).get(j)[2]      +
    cont.get(i).get(j)[3])  *  Math.pow((cont.get(i).get(j)[0]  *
    cont.get(i).get(j)[3])     -     (cont.get(i).get(j)[1]     *
    cont.get(i).get(j)[2]), 2)
7                       /((cont.get(i).get(j)[0]                +
```

```
        cont.get(i).get(j)[1])     *     (cont.get(i).get(j)[2])     +
        cont.get(i).get(j)[3])     *     (cont.get(i).get(j)[0])     +
        cont.get(i).get(j)[2])     *     (cont.get(i).get(j)[3])     +
        cont.get(i).get(j)[1]));
8                   //System.out.print(CHI[i][j]+" ");
9                }
10              //System.out.println();
11          }
12
13          double[][] weighted_avg = new double[CHI.length][2];
14          for (int i = 0; i < weighted_avg.length; i++) {
15              //weighted_avg[i][0] = CHI[i][0];
16              //weighted_avg[i][1] = Prior[0] * CHI[i][0];
17               for (int j = 0; j < 5; j++) {
18                   weighted_avg[i][0] += CHI[i][j];
19                  //System.out.println("Prior["+j+"]:          "
        +Prior[j] + " CHI["+i+"]["+j+"]: " + CHI[i][j]);
20                   weighted_avg[i][1]     +=     (Prior[j]     *
        CHI[i][j]);
21                  //System.out.println("       =       "       +
        weighted_avg[i][1]);
22               }
23          }
24         //System.out.println("CHI VALUE");
25          double[]          CHI_value          =          new
        double[weighted_avg.length];
26
27          for (int i = 0; i < CHI_value.length; i++) {
28              CHI_value[i] = weighted_avg[i][1];
29              //System.out.println(term[i]    +    "    |    "    +
        CHI_value[i]);
30          }
31
32          double[]    dataBaru    =    Arrays.copyOf(CHI_value,
        CHI_value.length);
33          String[]     termBaru      =      Arrays.copyOf(term,
        term.length);
34
35          String tmps;
36          double tmp;
37
38          for (int i = 0; i < CHI_value.length; i++) {
39              for (int j = i + 1; j < CHI_value.length; j++) {
40                  if (dataBaru[i] < dataBaru[j]) {
41                      tmp = dataBaru[i];
42                      dataBaru[i] = dataBaru[j];
43                      dataBaru[j] = tmp;
44
45                      tmps = termBaru[i];
46                      termBaru[i] = termBaru[j];
47                      termBaru[j] = tmps;
48                  }
49              }
50          }
51
52         //INI
53          double newLength = termBaru.length * feature/100;
54          String[] term_used = new String[(int) newLength];
55          for (int i = 0; i < term_used.length; i++) {
56              term_used[i] = termBaru[i];
57              //       System.out.println("term_used["+i+"]:   "
        +term_used[i]);
58          }
59          return term_used;
60      }
```

Pembahasan:

Baris 1	nama method untuk nilai *CHI*.
Baris 2-12	isi dari method nilai *CHI* yang berfungsi untuk menghitung nilai *CHI* masing-masing *term* tiap *Rating*.
Baris 13-31	isi dari method nilai *CHI* yang berfungsi untuk menghitung nilai *weighted average* masing-masing *term*.
Baris 32-51	isi dari method nilai *CHI* yang berfungsi untuk menghitung nilai *CHI* terbesar masing-masing *term*.
Baris 52-59	isi dari method nilai *CHI* yang berfungsi untuk mengambil *term* yang digunakan sesuai prosentasenya.
Baris 60	nilai kembalian dari nilai *CHI*.

5.4.3 Proses *Feature Selection* Menggunakan *Galavotti-Sebastiani-Simi Coefficient* (*GSS*)

Pada proses *feature selection* yaitu dengan menggunakan *Galavotti-Sebastiani-Simi Coefficient* adalah proses perhitungan nilai *GSS* untuk suatu *term* tertentu pada masing-masing kelas berdasarkan frekuensi kemunculan *term*/kata pada data latih. Setelah dilakukan perhitungan nilai *GSS* pada setiap kelas maka akan dipilih nilai *GSS* tertinggi untuk dijadikan nilai *GSS* untuk *term* tersebut. Proses tersebut dilakukan pada semua *term* hasil dari *pre-processing* pada setiap *review*. Setelah semua *term* mendapatkan nilai *GSS* selanjutnya akan dilakukan sorting berdasarkan nilai *GSS* pada masing-masing *term*. Hasil akhir dari proses ini adalah berupa susunan *term* yang nantinya akan diprioritaskan dalam proses klasifikasi. Potongan kode program untuk proses *feature selection* dengan menggunakan *Galavotti-Sebastiani-Simi Coefficient* dapat dilihat pada Kode Program 5.7.

Kode Program 5.7 Implementasi Kode *Galavotti-Sebastiani-Simi Coefficient* (*GSS*)

```
1   public static String[] sortByGssValue(List<ArrayList<int[]>>
    cont, String[] term, int feature) {
2          //System.out.println("GSS");
3           double[][] GSS = new double[cont.size()][5];
4
5           for (int i = 0; i < cont.size(); i++) {
6               for (int j = 0; j < cont.get(i).size(); j++) {
7                   GSS[i][j]    =    (cont.get(i).get(j)[0]    *
    cont.get(i).get(j)[3])     -     (cont.get(i).get(j)[1]     *
    cont.get(i).get(j)[2]);
8                  //       if (j == 0) {
9                  //            System.out.print(term[i] + ": "
    +GSS[i][j] + " ");
```

```
10                  //        }else{
11                  //              System.out.print(GSS[i][j] + "
    ");
12                  //        }
13              }
14             //System.out.println();
15          }
16
17          double[] max_GSS = new double[GSS.length];
18          for (int i = 0; i < max_GSS.length; i++) {
19              max_GSS[i] = GSS[i][0];
20              for (int j = 1; j < 5; j++) {
21                  if (max_GSS[i] < GSS[i][j]) {
22                      max_GSS[i] = GSS[i][j];
23                  }
24              }
25          }
26
27          double[]     dataBaru     =     Arrays.copyOf(max_GSS,
    max_GSS.length);
28          String[]      termBaru      =      Arrays.copyOf(term,
    term.length);
29
30          String tmps;
31          double tmp;
32
33          for (int i = 0; i < max_GSS.length; i++) {
34              for (int j = i + 1; j < max_GSS.length; j++) {
35                  if (dataBaru[i] < dataBaru[j]) {
36                      tmp = dataBaru[i];
37                      dataBaru[i] = dataBaru[j];
38                      dataBaru[j] = tmp;
39
40                      tmps = termBaru[i];
41                      termBaru[i] = termBaru[j];
42                      termBaru[j] = tmps;
43                  }
44              }
45          }
46
47         //INI
48          double newLength = termBaru.length * feature/100;
49          String[] term_used = new String[(int) newLength];
50          for (int i = 0; i < term_used.length; i++) {
51              term_used[i] = termBaru[i];
52             //       System.out.println("term_used["+i+"]:   "
    +term_used[i]);
53          }
54
55          return term_used;
56      }
```

Pembahasan:

Baris 1	nama method untuk nilai *GSS*.
Baris 2-16	isi dari method nilai *GSS* yang berfungsi untuk menghitung nilai *GSS* masing-masing *term* tiap *Rating*.
Baris 17-26	isi dari method nilai *GSS* yang berfungsi untuk menghitung nilai *weighted average* masing-masing *term*.
Baris 27-	isi dari method nilai *GSS* yang berfungsi untuk menghitung nilai

46	*GSS* terbesar masing-masing *term*.
Baris 47-54	isi dari method nilai *GSS* yang berfungsi untuk mengambil *term* yang digunakan sesuai prosentasenya.
Baris 55	nilai kembalian dari nilai *GSS*.

5.4.4 Proses Kombinasi *Feature Selection Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient*

Pada proses kombinasi *feature selection* yaitu dengan mengkombinasikan *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* adalah proses perhitungan nilai *CHI* dan *GSS* untuk suatu *term* tertentu pada masing-masing kelas berdasarkan frekuensi kemunculan *term*/kata pada data latih. Setelah dilakukan perhitungan nilai *CHI* dan *GSS* pada setiap kelas maka akan dipilih nilai *CHI* dan *GSS* tertinggi untuk dijadikan nilai *CHI* dan *GSS* untuk *term* tersebut. Proses tersebut dilakukan pada semua *term* hasil dari *pre-processing* pada setiap *review*. Setelah semua *term* mendapatkan nilai *CHI* dan *GSS* selanjutnya akan dilakukan sorting berdasarkan nilai *CHI* dan *GSS* pada masing-masing *term*. Hasil akhir dari proses ini adalah berupa kombinasi susunan *term* yang nantinya akan diprioritaskan dalam proses klasifikasi. Potongan kode program untuk proses *feature selection* dengan mengkombinasikan *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* dapat dilihat pada Kode Program 5.8.

Kode Program 5.8 Implementasi Kode Kombinasi *Chi-Square* (*CHI*) dan *Galavotti-Sebastiani-Simi Coefficient* (*GSS*)

```
1   public static String[] kombinasi(String[] term_by_CHI, String[] term_by_GSS) {
2           ArrayList<String> term_kombinasi_list = new ArrayList<>();
3
4           for (int i = 0; i < (term_by_CHI.length); i++) {
5               if (!exists(term_by_CHI[i], term_kombinasi_list)) {
6                   term_kombinasi_list.add(term_by_CHI[i]);
7               }
8           }
9
10          for (int i = 0; i < (term_by_GSS.length); i++) {
                if (!exists(term_by_GSS[i], term_kombinasi_list)) {
11                  term_kombinasi_list.add(term_by_GSS[i]);
12              }
13          }
14          String[] term_kombinasi = new String[term_kombinasi_list.size()];
15
16          for (int i = 0; i < term_kombinasi.length; i++) {
17              term_kombinasi[i] = term_kombinasi_list.get(i);
18          }
19          return term_kombinasi;
20      }
21
```

Pembahasan:

Baris 1 nama method untuk nilai kombinasi.

Baris 2-15 isi dari method nilai kombinasi yang berfungsi untuk mengkombinasikan kedua *feature selection*.

Baris 16-20 isi dari method nilai kombinasi yang berfungsi untuk mengambil *term* baru.

Baris 21 nilai kembalian dari nilai kombinasi.

5.5 Proses Evaluasi

Proses evaluasi merupakan kegiatan yang membandingkan antara hasil implementasi dengan kriteria standar yang telah ditetapkan untuk melihat keberhasilannya. Dari hasil evaluasi nantinya akan tersedia informasi mengenai sejauh mana suatu kegiatan tertentu telah dicapai sehingga bisa diketahui bila terdapat selisih standar yang telah ditetapkan dengan hasil yang bisa dicapai. Evaluasi dilakukan dengan menghitung nilai persentase akurasi dari sistem dengan membandingkan hasil klasifikasi sistem dengan kriteria yang telah ditetapkan.

5.5.1 Proses Menghitung Akurasi

Proses menghitung akurasi merupakan tahapan terakhir dari pengujian dengan menghitung jumlah keberhasilan sistem dibagi dengan seluruh dokumen kemudian dikalikan seratus maka akan didapatkan nilai persentase akurasi dari sistem dapat dilihat pada Kode Program 5.8.

Kode Program 5.9 Implementasi Kode Proses Evaluasi

```
1   int data_teruji = 0;
2           for (int i = 0; i < data_uji.length; i++) {
3               if              (Rating_result[i]              ==
    Integer.parseInt(data_uji[i][2])) {
4                   akurasi++;
5               }
6               if (Rating_result[i] != 0) {
7                   data_teruji++;
8               }
9           }
10
11          akurasi = akurasi/data_teruji * 100;
12          return dataTabel;
13      }
```

Pembahasan:

Baris 1 Inisialisasi akurasi tanpa *feature selection* dan dengan *feature*

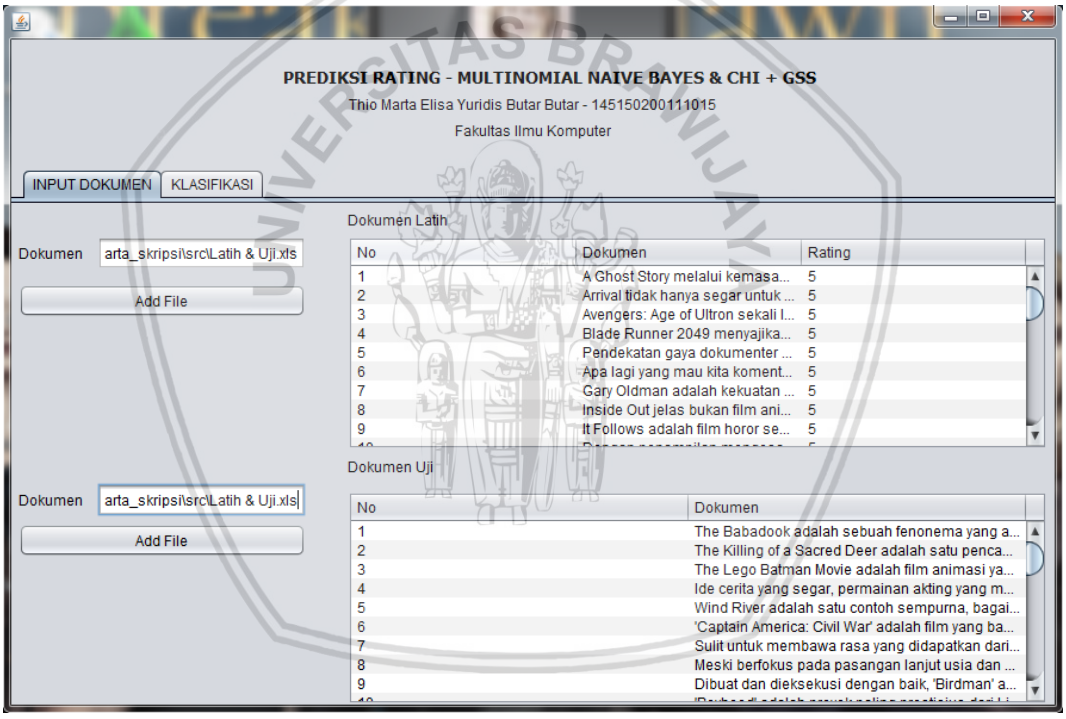
selection.

Baris 2-10 proses perulangan dan if-condition untuk akurasi tanpa *feature selection* dan dengan *feature selection*.

Baris 11-13 Hasil akhir dari proses pencarian akurasi tanpa *feature selection* dan dengan *feature selection*.

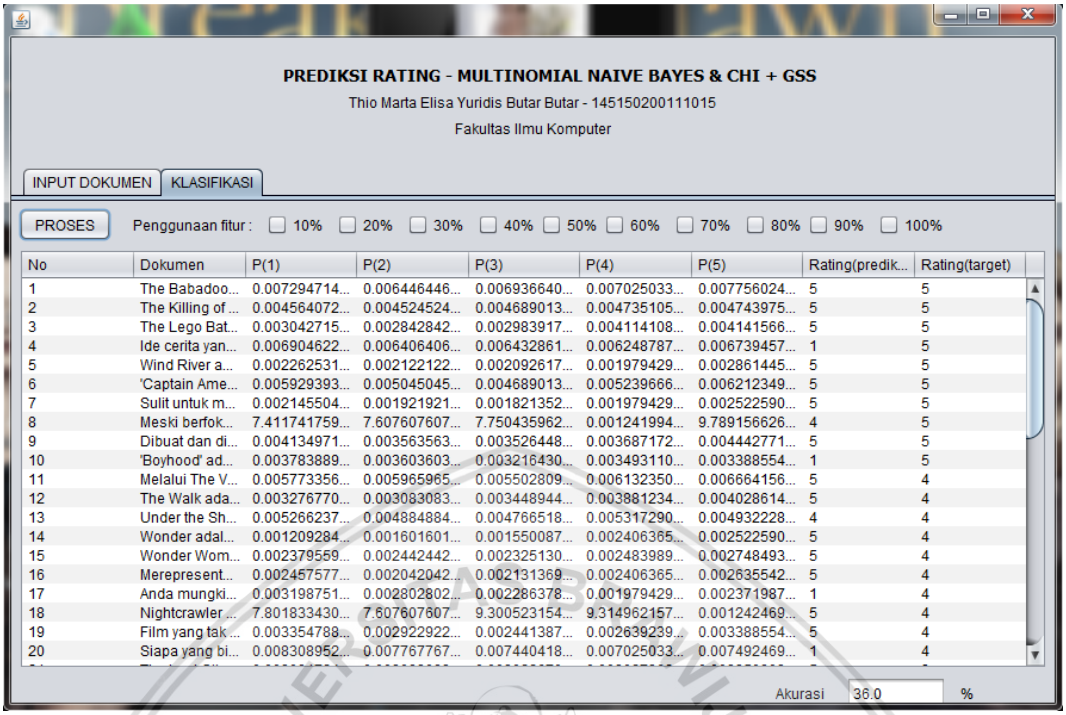
5.6 Implementasi Antarmuka

Antarmuka aplikasi digunakan oleh pengguna untuk berinteraksi dengan sistem. Aplikasi berbasis java dekstop dapat menangani operasi logika yang kompleks. Tampilan antarmuka sistem dapat dilihat pada Gambar 5.1.



Gambar 5.1 Antarmuka Sistem Prediksi *Rating* (menu input dokumen)

Pada Gambar 5.1 adalah antarmuka sistem prediksi *Rating* degan menggunakan metode *Multinomial Naïve Bayes Classifier*, *Chi-Square* , dan *Galavotti-Sebastiani-Simi Coefficient*. Terdapat dua menu yaitu yang pertama adalah menu input dokumen yang digunakan untuk menginputkan dokumen latih dan dokumen uji. Dokumen yang diinputkan bisa berupa file atau inputan secara manual. Dan yang kedua adalah menu klasifikasi yang digunakan untuk melakukan proses klasifikasi berdasarkan prosentase *feature* yang digunakan. Tampilan menu klasifikasi dapat dilihat pada Gambar 5.2.



Gambar 5.2 Menu Klasifikasi Prediksi *Rating* (menu output dokumen)

BAB 6 PENGUJIAN DAN ANALISIS

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

6.1 Pengujian Klasifikasi *Multinomial Naïve Bayes Classifier* Tanpa *Feature Selection*

Pengujian ini menjelaskan tentang pengujian hasil klasifikasi dengan menggunakan metode *Multinomial Naïve Bayes Classifier* tanpa *feature selection* atau tanpa dilakukan proses pengurangan *feature*/*term* pada suatu data *review*.

6.1.1 Skenario Pengujian Klasifikasi *Multinomial Naïve Bayes Classifier* Tanpa *Feature Selection*

Pengujian ini dilakukan untuk mengetahui tingkat akurasi pada klasifikasi prediksi *Rating* dengan menggunakan metode *Multinomial Naïve Bayes Classifier* dan tanpa menggunakan *feature selection*. Kumpulan *term* yang dihasilkan dari proses *pre-processing* akan langsung dilakukan klasifikasi tanpa harus dikurangi. Pada pengujian ini, data yang diuji benar-benar data asli *review* film yang diambil pada situs web https://montasefilm.com dan http://www.ulasanpilem.com, data uji yang dipakai pada pengujian ini sebanyak 50 data dengan komposisi data urut dari yang terbesar untuk *Rating* 1 sampai dengan *Rating* 5, sedangkan untuk data latih yang digunakan sebanyak 250 data dengan komposisi data 50 data *Rating* 1, 50 data *Rating* 2, 50 data *Rating* 3, 50 data *Rating* 4, 50 data *Rating* 5. Hasil pengujian untuk tingkat akurasi menggunakan *Multinomial Naïve Bayes Classifier* adalah sebesar 36%.

6.1.2 Analisis Hasil Pengujian Klasifikasi *Multinomial Naïve Bayes Classifier* Tanpa *Feature Selection*

Hasil pengujan seperti Tabel 6.1 menunjukkan tingkat akurasi dari proses klasifikasi menggunakan *Multinomial Naïve Bayes Classifier* sebesar 36%. Hasil tersebut menunjukkan bahwa hanya sebanyak 18 data dari 50 data uji yang diklasifikasi benar, dan peneliti beranggapan hasil tersebut tidak terlalu bagus. Data *review* pada situs web https://montasefilm.com dan http://www.ulasanpilem.com hampir semuanya bisa dikatakan tipe *review* semi long text dengan rata-rata jumlah kata antara 50-100 kata pada setiap *review*. Pada *review* tersebut juga sering juga ditemukan kata-kata yang tidak baku dimana nanti kata-kata tersebut juga akan dilakukan klasifikasi. Kata-kata yang tidak baku tersebut di antaranya seperti singkatan dan kata-kata yang walaupun sebenarnya adalah kata yang bersifat sentimen namun karena penulisannya akhirnya kata tersebut termasuk kata yang tidak baku juga. Hal tersebut mengakibatkan klasifikasi dokumen menjadi lebih lambat karena lebih banyak kata-kata yag harus diproses daripada yang sebenarnya harus diproses. Dan hal tersebut juga dapat mengurangi akurasi karena sistem harus

mempertimbangkan kata-kata yang tidak perlu untuk dilakukan klasifikasi. Untuk itu peneliti menggunakan *feature selection* untuk memperkecil dimensi *feature*, artinya *feature selection* di sini berfungsi untuk memilih kata-kata yang dianggap relevan atau diprioritaskan untuk dilakukan klasifikasi.

Selain beberapa faktor di atas, setelah dilakukan proses *pre-processing* masih terdapat kata-kata yang masih belum dalam bentuk kata dasar, artinya proses *stemming* yang dilakukan masih kurang maksimal. Sehingga bisa saja kata tersebut menjadi kata yang tidak baku walaupun terkadang kata tersebut termasuk kata yang bersifat sentimen dan perlu untuk dilakukan klasifikasi.

Faktor lain yang dapat mempengaruhi hasil akurasi ialah dikarenakan kata-kata pada data uji sama sekali tidak terdapat pada data latih. Sehingga kata-kata tersebut tidak dapat diklasifikasikan ke dalam *Rating* manapun. Begitu dengan *Rating* yang berdekatan juga mempengaruhi hasil akurasi, seperti: *Rating* 1 dan *Rating* 2, *Rating* 2 dan *Rating* 3, *Rating* 3 dan *Rating* 4, serta *Rating* 4 dan *Rating* 5. Pada kedua *Rating* yang berdekatan tersebut memiliki kata-kata yang relatif sama.

6.2 Pengujian Klasifikasi *Multinomial Naïve Bayes Classifier-CHI-GSS* Dengan Variasi Prosentase *Feature*

Pengujian ini menjelaskan tentang pengujian hasil klasifikasi dengan menggunakan metode *Multinomial Naïve Bayes Classifier-CHI-GSS* dengan variasi prosentase *feature* yang digunakan pada saat klasifikasi. Pada pengujian ini akan dilakukan pengurangan dimensi *feature* atau *term* hasil *pre-processing* yang digunakan pada saat klasifikasi adalah sebanyak prosentase yang telah ditetapkan yaitu sebesar 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, dan 100%.

6.2.1 Skenario Pengujian Klasifikasi *Multinomial Naïve Bayes Classifier-CHI-GSS* Dengan Variasi Prosentase *Feature*

Pengujian ini dilakukan untuk mengetahui seberapa besar pengaruh dari penggunaan *term* pada prosentase tertentu pada suatu data untuk dilakukan klasifikasi. Peran *CHI-GSS* di sini adalah sebagai *feature selection* dimana *term* dari hasil *pre-processing* nantinya akan dilakukan seleksi atau pemilihan *term* mana saja yang lebih diprioritaskan untuk dilakukan klasifikasi berdasarkan nilai *CHI-GSS* pada masing-masing *term*. Untuk data yang digunakan sama seperti pengujian sebelumnya yaitu sebanyak 50 data uji dengan komposisi data urut dari yang terbesar untuk *Rating* 1 sampai dengan *Rating* 5, sedangkan untuk data latih yang digunakan sebanyak 250 data dengan komposisi data 50 data *Rating* 1, 50 data *Rating* 2, 50 data *Rating* 3, 50 data *Rating* 4, dan 50 data *Rating* 5. Untuk hasil pengujian ini dapat dilihat pada Tabel 6.2.

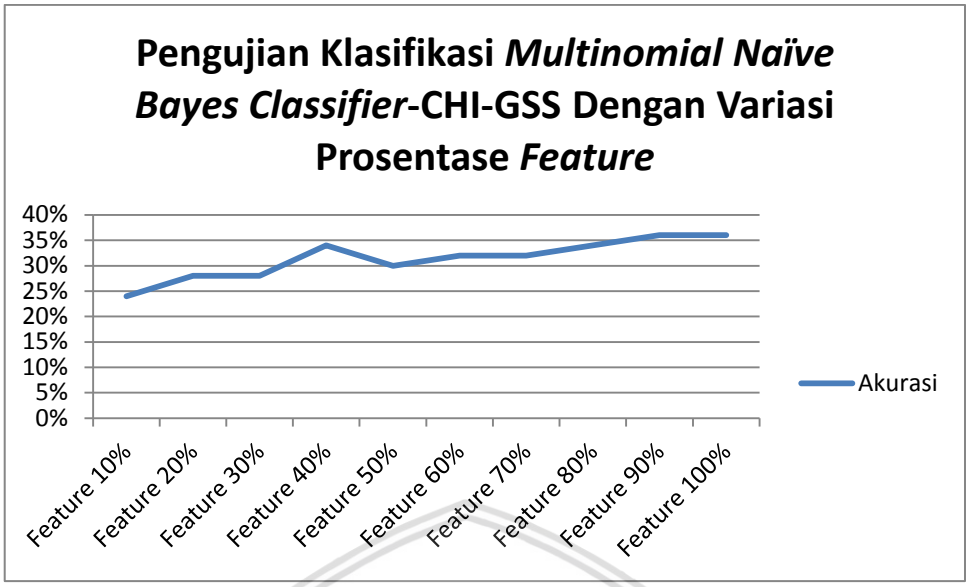
Tabel 6.1 Pengujian Klasifikasi *Multinomial Naïve Bayes Classifier-CHI-GSS* Dengan Variasi Prosentase *Feature*

Prosentase *Feature*	Akurasi
10%	24%
20%	28%
30%	28%
40%	34%
50%	30%
60%	32%
70%	32%
80%	34%
90%	36%
100%	36%

6.2.2 Analisis Hasil Pengujian Klasifikasi *Multinomial Naïve Bayes Classifier-CHI-GSS* Dengan Variasi Prosentase *Feature*

Pada pengujian ini sudah dilakukan pengurangan dimensi *feature* sesuai prosentase yang sudah ditetapkan oleh peneliti. Dengan data yang sama pengujian ini mendapatkan nilai akurasi terbaik sebanyak 36% pada penggunaan *feature* sebanyak 90% dan 100%. Dengan kata lain penggunaan *CHI-GSS* di sini dapat mempengaruhi nilai akurasi dengan mengurangi atau membuang *term-term* yang dianggap tidak relevan untuk diklasifikasi.

Dengan mengurangi dimensi *feature* yang digunakan akan membantu sistem lebih maksimal dalam memproses *term-term* untuk dilakukan klasifikasi. Karena pada suatu data *review* pasti terdapat beberapa *term* yang relevan dan tidak relevan untuk dilakukan klasifikasi. Karena itu, penggunaan *feature CHI-GSS* di sini sebagai *feature selection* bisa membantu dalam pemilihan *term* yang dianggap relevan maupun yang tidak relevan. Namun dalam pengurangan dimensi *feature* tidak menjamin tingkat akurasi lebih baik ketika dimensi *feature* semakin kecil, tergantung seberapa besar prosentase pengunaan *feature* itu sendiri. Untuk hasil pengujian ini dapat dilihat pada Gambar 6.1.



Gambar 6.1 Grafik Akurasi *Multinomial Naïve Bayes Classifier-CHI-GSS* dengan Variasi Prosentase *Feature*

Pada hasil pengujian yang didapatkan seperti pada Gambar 6.1 menunjukkan bahwa klasifikasi dengan menggunakan *Multinomial Naïve Bayes Classifier-CHI-GSS* bisa mendapatkan hasil yang lebih akurat. Pada Gambar 6.1 didapatkan tingkat akurasi tertinggi sebesar 36%. pada penggunaan *feature* sebanyak 90%, dan 100%, peneliti menganalisis bahwa selian *CHI-GSS* bisa memberikan pengaruh pada tingkat akurasi, prosentase *feature* yang digunakan juga bisa mempengaruhi dari hasil klasifikasi. Pada penelitian ini nilai akurasi terbaik yang didapatkan adalah pada saat prosentase *feature* yang digunakan sebesar 90% dan 100% dan didapat juga nilai akurasi yang paling rendah pada saat prosentase penggunaan *feature* sebesar 10%. Hal tersebut bisa menjadi alasan ketika kita menggunakan *feature selection* bukan berarti semakin kita memperkecil dimensi *feature* yang digunakan maka akan memberikan tingkat akurasi yang lebih bagus. Karena pada kasus ini penggunaan *feature* yang terlalu sedikit justru menyebabkan tingkat akurasi yang didapatkan paling rendah. Hal tersebut dikarenakan *term* hasil *pre-processing* dan *CHI-GSS* terlalu sedikit untuk dilakukan klasifikasi artinya sistem bisa saja kekurangan informasi karena terbatasnya informasi yang terlalu sedikit dan mendapatkan hasil yang kurang maksimal. Untuk analisis yang dilakukan peneliti pada setiap prosentase *feature* yang digunakan sudah dirincikan pada point-point di bawah ini:

1. Pada prosentase penggunaan *feature* sebesar 90%, dan 100% (*feature* sepenuhnya) sistem berhasil mendapatkan hasil yang signifikan lebih baik dari penggunaan *feature* sebesar lainnya. Tingkat akurasi yang didapatkan adalah sebesar 36% artinya sistem sudah berhasil mendapatkan hasil klasifikasi yang tepat untuk 18 data dari 50 data uji. Pada prosentase ini *CHI-GSS* hanya menggunakan *term* sebesar 90%, dan 100% (*feature* sepenuhnya) untuk dilakukan klasifikasi dan sisanya akan dibuang. Hal ini menunjukkan

bahwa kumpulan *term* hasil *pre-processing* kurang lebih separuhnya adalah *term* yang sebenarnya tidak perlu atau tidak diprioritaskan untuk dilakukan klasifikasi. Dengan penggunaan *term* sebesar 90%, 100% (*feature* sepenuhnya) tersebut bisa mendapatkan hasil yang mempertahankan akurasi maksimalnya.

2. Pada prosentase penggunaan *feature* sebesar 40% dan 80% tingkat akurasi yang didapatkan lebih bagus dari prosentase penggunaan *feature* sebesar 90%, dan 100%. Tingkat akurasi yang didapatkan sebesar 34%, artinya adalah kumpulan *term* uji hasil *pre-processing* akan dikurangi sebesar 60% dan 20% berdasarkan nilai *CHI-GSS* terendah pada masing-masing *term* sehingga 40% dan 80% *term* akan dilakukan klasifikasi. Walaupun tingkat akurasi yang didapatkan tidak jauh beda dengan penggunaan *feature* sepenuhnya, dapat dikatakan peran *CHI-GSS* di sini berhasil memPrioritaskan untuk dilakukan klasifikasi dan memberikan hasil yang lebih bagus.
3. Pada prosentase penggunaan *feature* sebesar 60% dan 70% tingkat akurasi yang didapatkan lebih bagus dari prosentase penggunaan *feature* sebesar 80%. Tingkat akurasi yang didapatkan sebesar 32%, artinya adalah kumpulan *term* uji hasil *pre-processing* akan dikurangi sebesar 40% dan 30% berdasarkan nilai *CHI-GSS* terendah pada masing-masing *term* sehingga 60% dan 70% *term* akan dilakukan klasifikasi. Walaupun tingkat akurasi yang didapatkan tidak jauh beda dengan penggunaan *feature* sepenuhnya, dapat dikatakan peran *CHI-GSS* di sini berhasil memPrioritaskan untuk dilakukan klasifikasi dan memberikan hasil yang lebih bagus.
4. Pada saat penggunaan *feature* sebesar 50% tingkat akurasi yang didapatkan hanya sebesar 30%, selain beberapa faktor yang sudah disebutkan di atas hal ini bisa saja disebabkan karena setelah dilakukan *pre-processing* masih terdapat kata-kata pada yang seharusnya tidak perlu dilakukan klasifikasi tapi tetap dilakukan. Kata-kata yang dimaksud adalah seperti kata-kata yang tidak baku. Hal tersebut tentunya akan mempengaruhi hasil dari klasifikasi dan tidak mendapatkan hasil yang maksimal.
5. Pada saat penggunaan *feature* sebesar 20% dan 30% tingkat akurasi yang didapatkan hanya sebesar 28%, selain beberapa faktor yang sudah disebutkan di atas hal ini bisa saja disebabkan karena setelah dilakukan *pre-processing* masih terdapat kata-kata pada yang seharusnya tidak perlu dilakukan klasifikasi tapi tetap dilakukan. Kata-kata yang dimaksud adalah seperti kata-kata yang tidak baku. Hal tersebut tentunya akan mempengaruhi hasil dari klasifikasi dan tidak mendapatkan hasil yang maksimal.
6. Sedangkan pada prosentase penggunaan *feature* sebesar 10% sistem justru mendapatkan hasil yang tidak maksimal. Hasil yang didapatkan pada prosentase ini adalah hanya sebesar 24% artinya hasil tersebut adalah yang terendah dari prosentase penggunaan *feature* yang lain. Peneliti menganalisis hal ini disebabkan karena pengunaan *feature* yang terlalu sedikit sehingga informasi yang diperlukan oleh sistem terlalu sedikit untuk dilakukan klasifikasi, dan bisa saja terdapat *term* yang semestinya diprioritaskan untuk dilakukan klasifikasi tetapi ikut terbuang atau tidak dilakukan klasifikasi,

sehingga hal tersebut menyebabkan sistem kurang maksimal dalam menghasilkan hasil klasifikasi.

BAB 7 PENUTUP

Pada bab ini akan dibahas mengenai kesimpulan dari peneliti yang telah dilakukan dan juga saran yang diberikan penulis untuk penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis dari prediksi *Rating* pada *review* film menggunakan Metode *Multinomial Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* dapat disimpulkan sebagai berikut:

1. Klasifikasi menggunakan *Naïve Bayes Classifier* dengan *feature selection* berbasis *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* dapat memberikan hasil yang lebih baik daripada menggunakan *Naïve Bayes Classifier* biasa. Hasil terbaik yang didapat pada saat penggunaan *feature* sebanyak 90%, dan 100%. dengan tingkat akurasi sebesar 36%. Hal tersebut membuktikan *CHI-GSS* berhasil melakukan pemilihan kata yang lebih diprioritaskan untuk dilakukan klasifikasi dan membuang kata-kata yang dianggap tidak relevan untuk dilakukan klasifikasi.
2. Pengurangan dimensi *feature* dengan menggunakan *Chi-Square* dan *Galavotti-Sebastiani-Simi Coefficient* yang diterapkan dengan menggunakan data dari situs web https://montasefilm.com dan http://www.ulasanpilem.com tidak menjamin ketika semakin kita memperkecil dimensi *feature* yang digunakan maka akan memberikan tingkat akurasi yang lebih baik. Karena pada pengujian yang dilakukan pada saat penggunaan *feature* sebesar 10% sistem memberikan nilai akurasi yang paling rendah yaitu hanya sebesar 24%, artinya adalah *term-term* yang diproses pada saat klasifikasi tidak memberikan hasil yang maksimal karena bisa saja terdapat kata-kata yang sebelumnya relevan untuk dilakukan klasifikasi tetapi ikut terbuang sehingga sistem kekurangan informasi untuk memberikan hasil yang maksimal.

7.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka berikut merupakan beberapa saran untuk pengembangan penelitian selanjutnya:

1. Pada penelitian yang dilakukan, data yang diambil pada situs web https://montasefilm.com dan http://www.ulasanpilem.com masih terdapat beberapa data atau *review* yang mengandung kata-kata serapan. Sehingga ketika dilakukan klasifikasi hasil yang didapatkan kurang maksimal dan data *review* yang digunakan terdapat juga *review* itu sendiri, sehingga bisa dikatakan sulit untuk menggambarkan suatu *review* untuk *Rating* tertentu.
2. Pada data yang digunakan, *review* yang memiliki *Rating* yang berdekatan, seperti: *Rating* 1 dan *Rating* 2, *Rating* 2 dan *Rating* 3, *Rating* 3 dan *Rating* 4, serta *Rating* 4 dan *Rating* 5 terkadang makna yang diutarakan oleh pengguna sama tetapi *Rating* yang diberikan berbeda. Dan ada juga dengan *Rating* yang

sama namun memiliki makna *review* yang berbeda. Maka dari itu, untuk pengembangan lebih lanjut peneliti menyarankan untuk menggunakan algoritme leksikal untuk membedakan jenis dari setiap *term* sehingga nanti mudah untuk membedakan *term* mana saja yang lebih menggambarkan suatu kelas tertentu.

3. Pada penelitian yang dilakukan menggunakan perhitungan tiap kata, bukan tiap frasa. Seperti yang telah diteliti, yaitu menggunakan uni-gram. Oleh sebab itu, untuk pengembangan selanjutnya dapat menggunakan perhitungan tiap frasa. Contohnya : bi-gram dan tri-gram. Agar mendapat akurasi yang lebih maksimal.

DAFTAR PUSTAKA

Adel, A., Omar, N., & Al-Shabi, A. (2014). *A Comparative Study Of Combined Feature Selection Methods For Arabic Text Classification. Journal Of Computer Science, 10(11), 2232-2239.*

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M. M., Williams, H. E. (2007). *Stemming* Indonesian: A Confix-Stripping Approach. Acm J. Educ. Resourch. Computer. 6, 4, Article 13, 33 Pages.

Agusta, Ledy. (2009). Perbandingan Algoritma *Stemming* Porter Dengan Algoritma Nazief & Adriani Untuk *Stemming* Dokumen Teks Bahasa Indonesia. Konferensi Nasional Sistem Dan Informatika 2009: Bali, November 14, 2009.

Andilala. (2016). Movie *Review* Sentimen Analisis Dengan Metode *Naïve Bayes* Base On *Feature Selection*. Journal Pseudocode, Iii(1).

Bhoir, P. & Kolte, S. (2015). *Sentiment Analysis Of Movie Reviews Using Lexicon Approach. Ieee International Conference On Computational Intelligence And Computing Research.*

Dave, Kandarp. (2011). *Study Of Feature Selection Algorithms For Text-Categorization. Las Vegas: University Of Nevada.*

Destuardi & Surya, S. (2009). Klasifikasi Emosi Untuk Teks Bahasa Indonesia Menggunakan Metode *Naïve Bayes*. Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.

Effendy, Onong Uchjana. (1986). Dimensi-Dimensi Komunikasi. Bandung: Alumni.

Feldman, R. & Sanger, J. (2007). *The Text Mining Handbook: Advanced Approaches In Analyzing Unstructured Data. Cambridge University Press.*

Forman, G. (2008). *Feature Selection For Text Classification, In H. Liu And H. Motoda, Eds, 'Computational Methods Of Feature Selection', Chapman And Hall/Crc, Pp. 257–276.*

Garnes, Øystein Løhre. (2009). *Feature Selection For Text Categorization. Norwegian University Of Science And Technology.*

Guo, Q. (2010). *An Effective Algorithm For Improving The Performance Of Naïve Bayes For Text Classification. Cambridge University Press.*

Jong, J. (2011). *Predicting Rating With Sentiment Analysis.*

Kurniawan, B., Fauzi, M. A., & Widodo, A. W. (2017). Klasifikasi Berita Twitter Menggunakan Metode *Improved Naïve Bayes.*

Medhat, W., Hassan, A., & Korashy, H. (2014). *Sentiment Analysis Algorithms And Applications: A Survey. Ain Shams Engineering Journal, 5(4), 1093-1113.*

Mustafa, A., Akbar, A., & Sultan, A. (2009). *Knowledge Discovery Using Text Mining: A Programmable Implementation On Information Extraction And*

Categorization. International Journal Of Multimedia And Ubiquitous Engineering, 4(2), 183-188.

Rosi, F., Fauzi, M. A., & Perdana, R. S. (2018). Prediksi *Rating* Pada *Review* Produk Kecantikan Menggunakan Metode *Naïve Bayes* Dan Categorical Proportional Difference (Cpd).

Sahu, T. P. & Ahuja, S. (2016). *Sentiment Analysis Of Movie Reviews: A Study On Feature Selection & Classification Algorithms. Ieee.*

Uchyigit, Gulden. (2012). *Experimental Evaluation Of Feature Selection Methods For Text Classification. Ieee 9th International Conference On Fuzzy Systems And Knowledge Discovery.*

Wijaya, M. C., Tjiharjadi, S. (2010). Aplikasi Klasifikasi Dokumen Menggunakan Metoda Naïve Baysian.

Zheng, Z., Srihari, R., Srihari, S. (2003). *A Feature Selection Framework For Text Filtering. Proceedings Of The Third Ieee International Conference On Data Mining.*