ANALISIS SENTIMEN PADA ULASAN APLIKASI BCA MOBILE MENGGUNAKAN BM25 DAN *IMPROVED K-NEAREST NEIGHBOR*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Indriya Dewi Onantya
NIM: 155150200111037

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

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

ANALISIS SENTIMEN PADA ULASAN APLIKASI BCA MOBILE MENGGUNAKAN BM25 DAN *IMPROVED K-NEAREST NEIGHBOR*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Indriya Dewi Onantya
NIM: 155150200111037

Skripsi ini telah diuji dan dinyatakan lulus pada
3 Januari 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Indriati, S.T., M.Kom.	Putra Pandu Adikara, S.Kom., M.Kom.
NIP: 19831013 201504 2 002	NIP: 19850725 200812 1 002

Mengetahui
Ketua Jurusan Teknik Informatika

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

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

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

Malang, 3 Januari 2019

METERAI TEMPEL
TGL. 20
D88E2AFF049189276
6000
ENAM RIBU RUPIAH

Indriya Dewi Onantya

NIM: 155150200111037

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PRAKATA

Puji syukur penulis panjatkan atas ke hadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian yang berjudul "Analisis Sentimen Pada Ulasan Aplikasi BCA Mobile Menggunakan Metode BM25 Dan *Improved K-Nearest Neighbor*".

Penelitian ini merupakan syarat untuk memenuhi sebagian persyaratan kurikulum pada Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya. Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada pihak-pihak yang telah membantu penulis selama penyusunan laporan penelitian ini, antara lain:

1. Ibu Indriati, S.T., M.Kom., selaku Dosen Pembimbing I yang dengan tulus ikhlas membimbing dan membantu penulis dalam menyelesaikan penelitian ini.
2. Bapak Putra Pandu Adikara, S.Kom., M.Kom., selaku Dosen Pembimbing II yang dengan sabar membimbing dan mengarahkan penulis dalam penyusunan laporan penelitian ini.
3. Bapak Agus Wahyu Widodo, S.T., M.Cs., selaku Ketua Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D., selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Achmad Basuki, S.T., M.MG., Ph.D., selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama menempuh proses perkuliahan.
6. Keluarga penulis, terutama orang tua, nenek, dan adik penulis yang dengan seluruh kemampuan dan keikhlasannya membantu penulis dalam segala hal, serta selalu memberikan doa dan motivasi kepada penulis dalam menyelesaikan skripsi ini.
7. Ragil Insan Maulana, selaku orang terdekat penulis yang sudah mencurahkan seluruh tenaga dan perhatian kepada penulis ketika pengerjaan skripsi ini.
8. Deri Hendra Binawan, selaku tutor bagi penulis yang dengan sabar membantu penulis dalam menyelesaikan program skripsi.
9. Khalisma Frinta, Dwi Suci Ariska Yanti, Rahmat Faizal, M. Shidqi Fadlillah, dan Kevin Azwega selaku pemberi semangat dan sebagai tempat berbagi keluh kesah penulis selama proses perkuliahan dan pengerjaan skripsi.
10. Seluruh pihak yang telah mendukung dan memberi bantuan kepada penulis yang belum dicantumkan namanya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Penulis menyadari penelitian ini masih banyak kekurangan dan jauh dari kata sempurna, untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun. Semoga penelitian ini dapat bermanfaat dalam membantu penelitian selanjutnya.

Malang, 3 Januari 2019

Penulis

indriyadewi@gmail.com

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRAK

Indriya Dewi Onantya, Analisis Sentimen Pada Ulasan Aplikasi BCA Mobile Menggunakan BM25 Dan *Improved K-Nearest Neighbor*

Pembimbing: Indriati, S.T., M.Kom., dan Putra Pandu Adikara, S.Kom., M.Kom.

Di era informasi seperti saat ini, internet memegang peranan penting dalam segala aspek kehidupan manusia. Hal ini mendorong berbagai aplikasi *mobile* menjadi berkembang dengan sangat cepat. Beberapa aplikasi akhirnya bermunculan dengan tujuan mempermudah dalam memenuhi kebutuhan manusia. Aplikasi *mobile banking* merupakan salah satu aplikasi yang dianggap dapat memberikan kemudahan dalam kegiatan manusia. Salah satu aplikasi *mobile banking* yang banyak digunakan adalah aplikasi BCA Mobile. Setiap aplikasi *mobile* tersebut mengijinkan pengguna untuk memberikan ulasan terhadap pelayanan dan fitur yang diberikan. Namun dari aplikasi *mobile* yang ada saat ini, tidak ditemukan adanya fitur analisis sentimen untuk mengelompokkan atau menyaring antara ulasan positif dan negatif. Untuk mengetahui ulasan tersebut tergolong ulasan positif atau negatif, maka diperlukan analisis sentimen untuk mengelompokkan ulasan tersebut. Dalam penelitian ini, analisis sentimen dilakukan dalam beberapa tahapan atau proses. Proses tersebut terdiri dari *pre-processing* data, pembobotan kata yang diterapkan menggunakan algoritme BM25, dan algoritme *Improved K-Nearest Neighbor* untuk proses klasifikasi dokumen. Hasil pengujian yang dilakukan pada penelitian dengan menggunakan *5-fold cross validation* mendapatkan hasil terbaik ketika nilai *k-values* sama dengan 10 dengan nilai *f-measure* sebesar 0,939, nilai *precision* sebesar 0,946, nilai *recall* sebesar 0,934, dan nilai *accuracy* sebesar 0,942. Maka dari itu dapat disimpulkan bahwa hasil dari analisis sentimen ini dipengaruhi oleh nilai *k-values,* sedangkan jumlah kelas pada tiap data latih tidak memengaruhi hasil evaluasi, karena nilai *k-values* baru yang dihasilkan dalam algoritme klasifikasi ini menyesuaikan dengan jumlah data uji tiap kelas.

Kata kunci: analisis sentimen, aplikasi *mobile,* BM25, *Improved K-Nearest Neighbor*

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Indriya Dewi Onantya, Analisis Sentimen Pada Ulasan Aplikasi BCA Mobile Menggunakan BM25 Dan *Improved K-Nearest Neighbor*

Supervisors: Indriati, S.T., M.Kom., and Putra Pandu Adikara, S.Kom., M.Kom.

In the era of information, the internet requires a lot of things in humans life. This encourages various mobile applications developed very quickly. Some applications have finally sprung up with visual goals in human requirements. Mobile banking application is one application that can provide convenience in human activities. One of the mobile banking applications BCA Mobile application. Every mobile application allows users to provide service and features that given by the application. But from the existing mobile applications, there are no features that can be used to group or filter between positive and negative reviews. To find out positive or negative information, a sentiment analysis is needed to group the reviews. In this study, the sentiment analysis was carried out in several stages or processes. The process consists of pre-processing data, word weighting used the BM25 algorithm, and Improved K-Nearest Neighbor algorithm that used as document classification. The results of the tests using 5-fold cross validation get the best results with a k-value equal to 10 with f-measure value are 0.939, a precision value is 0.946, a recall value is 0.934, and an accuracy value is 0.942. Therefore, we can conclude the results of this sentiment analysis by the value of k-value, while the amount of data on each class does not affect the evaluation results, because the value of the new k-value in this algorithm can adapt to the amount of data based on the class.

Keywords: sentiment analysis, mobile application, BM25, Improved K-Nearest Neighbor

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
PRAKATA iv
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xi
DAFTAR GAMBAR xii
DAFTAR LAMPIRAN xiii
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan masalah 3
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan masalah 3
1.6 Sistematika pembahasan 4
BAB 2 LANDASAN KEPUSTAKAAN 5
2.1 Kajian Pustaka 5
2.2 Ulasan 6
2.3 Analisis Sentimen 7
2.4 *Text Mining* 7
2.5 *Pre-processing* 8
2.5.1 *Case Folding* 8
2.5.2 *Tokenizing* 8
2.5.3 *Filtering* 9
2.5.4 *Stemming* 9
2.6 Metode BM25 10
2.7 *Improved K-Nearest Neighbor* 10
2.8 Evaluasi 11
2.8.1 *K-Fold Cross Validation* 11

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.8.2 Evaluasi *Precision, Recall, F-Measure,* Dan *Accuracy*..... 12
BAB 3 METODOLOGI 14
3.1 Tipe Penelitian 14
3.2 Strategi Penelitian..... 14
3.3 Lokasi Penelitian 14
3.4 Peralatan Pendukung..... 14
3.5 Pengumpulan Data 15
3.6 Teknik Penerapan Algoritme 15
3.7 Penarikan Kesimpulan 16
BAB 4 PERANCANGAN DAN IMPLEMENTASI 17
4.1 Deskripsi Permasalahan..... 17
4.2 Deskripsi Umum Sistem 17
4.3 Proses *Training* Data 18
4.3.1 *Case folding* 19
4.3.2 *Tokenizing* 20
4.3.3 *Filtering* 21
4.3.4 *Stemming* 21
4.4 Proses *Testing* Data..... 22
4.4.1 *Case Folding* 24
4.4.2 *Tokenizing* 24
4.4.3 *Filtering* 24
4.4.4 *Stemming* 24
4.4.5 Perhitungan Nilai TF 24
4.4.6 Perhitungan Nilai DF 25
4.4.7 Perhitungan Nilai IDF 27
4.4.8 Perhitungan *Scoring* BM25 28
4.4.9 Pemeringkatan Hasil *Scoring* BM25 28
4.4.10 Perhitungan Nilai *n* (*k-values* baru)..... 29
4.4.11 Perhitungan Probabilitas..... 30
4.4.12 Penentuan Kelas..... 31
4.5 Manualisasi 32
4.5.1 Manualisasi *Case Folding* 33

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.5.2 Manualisasi *Tokenizing* 33
4.5.3 Manualisasi *Filtering* 34
4.5.4 Manualisasi *Stemming* 35
4.5.5 Manualisasi Metode BM25 35
4.5.6 Manualisasi *Improved K-Nearest Neighbor* 37
4.6 Perancangan Pengujian Sistem 39
4.6.1 Perancangan *Confusion Matrix* 39
4.6.2 Perancangan Pengujian Pengaruh Nilai *k-values* 39
4.7 Penarikan Kesimpulan 40
4.8 Implementasi Sistem 40
4.8.1 Implementasi *Pre-processing* 40
4.8.2 Implementasi Metode BM25 43
4.8.3 Implementasi *Improved K-Nearest Neighbor* 45
4.9 Tampilan Sistem 47
BAB 5 HASIL DAN PEMBAHASAN 49
5.1 Hasil Sistem 49
5.2 Hasil Pengujian 50
5.2.1 Pengujian *5-Fold* Pada *Fold* Ke-1 50
5.2.2 Pengujian *5-Fold* Pada *Fold* Ke-2 51
5.2.3 Pengujian *5-Fold* Pada *Fold* Ke-3 51
5.2.4 Pengujian *5-Fold* Pada *Fold* Ke-4 52
5.2.5 Pengujian *5-Fold* Pada *Fold* Ke-5 53
5.2.6 Rata-Rata Pengujian Kelas Positif 53
5.2.7 Rata-Rata Pengujian Kelas Negatif 54
5.2.8 Rata-Rata Hasil Seluruh Pengujian 54
5.3 Analisis 55
BAB 6 PENUTUP 57
6.1 Kesimpulan 57
6.2 Saran 57
DAFTAR REFERENSI 58

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

Tabel 2.1 *Confusion Matrix* 12
Tabel 4.1 Data Latih 32
Tabel 4.2 Data Uji 33
Tabel 4.3 Manualisasi *Case Folding* (Data Latih) 33
Tabel 4.4 Manualisasi *Case Folding* (Data Uji) 33
Tabel 4.5 Manualisasi *Tokenizing* (Data Latih) 34
Tabel 4.6 Manualisasi *Tokenizing* (Data Uji) 34
Tabel 4.7 Manualisasi *Filtering* (Data Latih) 34
Tabel 4.8 Manualisasi *Filtering* (Data Uji) 35
Tabel 4.9 Manualisasi *Stemming* (Data Latih) 35
Tabel 4.10 Manualisasi *Stemming* (Data Uji) 35
Tabel 4.11 Perhitungan TF 36
Tabel 4.12 Panjang Dokumen Dan Rata-Rata Panjang Dokumen 36
Tabel 4.13 Nilai DF dan IDF 36
Tabel 4.14 Hasil *Scoring* BM25 37
Tabel 4.15 Hasil Pemeringkatan BM25 38
Tabel 4.16 Jumlah Data Latih 38
Tabel 4.17 Hasil Perhitungan Nilai *n* (*k-values* baru) 38
Tabel 4.18 Perancangan *Confusion Matrix* Positif 39
Tabel 4.19 Perancangan *Confusion Matrix* Negatif 39
Tabel 4.20 Perancangan Pengujian Pengaruh Nilai *k-values* 40
Tabel 5.1 Hasil Klasifikasi Sistem 49
Tabel 5.2 Hasil Pengujian *Fold* 1 50
Tabel 5.3 Hasil Pengujian *Fold* 2 51
Tabel 5.4 Hasil Pengujian *Fold* 3 51
Tabel 5.5 Hasil Pengujian *Fold* 4 52
Tabel 5.6 Hasil Pengujian *Fold* 5 53
Tabel 5.7 Rata-Rata Pengujian Kelas Positif 53
Tabel 5.8 Rata-Rata Pengujian Kelas Negatif 54
Tabel 5.9 Rata-Rata Hasil Seluruh Pengujian 54

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2.1 Tahapan *Pre-processing* 8
Gambar 2.2 Contoh *case folding* 8
Gambar 2.3 Contoh *tokenizing* 9
Gambar 2.4 Contoh *filtering* 9
Gambar 2.5 *K-fold Cross Validation* 12
Gambar 3.1 Diagram Alir Teknik Penerapan Algoritme 15
Gambar 4.1 Diagram Alir Perancangan Algoritme, (a) Proses *Training*, (b) Proses *Testing* 18
Gambar 4.2 Proses *Training* Data 18
Gambar 4.3 Alur tahapan *Case Folding* 19
Gambar 4.4 Alur Tahapan *Tokenizing* 20
Gambar 4.5 Alur Tahapan *Filtering* 21
Gambar 4.6 Alur Tahapan *Stemming* 22
Gambar 4.7 Alur Proses *Testing* Data 23
Gambar 4.8 Alur Perhitungan Nilai TF 24
Gambar 4.9 Alur Perhitungan Nilai DF 26
Gambar 4.10 Alur Perhitungan IDF 27
Gambar 4.11 Alur Perhitungan *Scoring* BM25 28
Gambar 4.12 Alur Tahapan Pemeringkatan Hasil *Scoring* BM25 29
Gambar 4.13 Alur Tahapan Perhitungan Nilai *n* 30
Gambar 4.14 Alur Tahapan Perhitungan Probabilitas 30
Gambar 4.15 Alur Tahapan Penentuan Kelas 31
Gambar 4.16 Hasil Tampilan Sistem 48
Gambar 5.1 Grafik Rata-Rata Hasil Pengujian 56

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR LAMPIRAN

LAMPIRAN A DATA 60
LAMPIRAN B HASIL KLASIFIKASI SISTEM 62
LAMPIRAN C CONFUSION MATRIX 67
LAMPIRAN D HASIL PENGUJIAN 77
LAMPIRAN E STOPWORD LIST TALA 82

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 1 PENDAHULUAN

Bab ini berisi tentang masalah atau latar belakang yang mendasari pembuatan sistem analisis sentimen pada ulasan aplikasi BCA Mobile menggunakan BM25 dan *Improved K-Nearest Neighbor*, beserta rumusan masalah, tujuan, manfaat, serta sistematika pembahasan yang akan dilakukan dalam penelitian.

1.1 Latar belakang

Di era informasi seperti saat ini, internet telah memegang peranan penting dalam segala aspek kehidupan manusia. Dengan adanya internet, maka manusia semakin terbantu dalam segala hal. Maka dari itu, saat ini semua hal dapat dikaitkan dengan internet. Dalam penggunaannya, internet sangat erat kaitannya dengan perangkat atau *mobile*. Hal ini juga yang mendorong berbagai aplikasi *mobile* menjadi berkembang dengan sangat cepat. Beberapa aplikasi akhirnya bermunculan dengan tujuan mempermudah dalam memenuhi kebutuhan manusia. Mulai dari kebutuhan rumah tangga, gaya hidup, hobi, olahraga, musik, *mobile banking* dan masih banyak kebutuhan lainnya. Aplikasi *mobile banking* merupakan salah satu aplikasi yang dianggap dapat memberikan kemudahan dalam kegiatan manusia. Aplikasi ini dapat memudahkan pengguna dalam melakukan proses transaksi yang berkaitan dengan keuangan tanpa harus pergi langsung menuju bank terkait. Hal ini tentunya sangat bermanfaat dalam menghemat waktu bagi pengguna yang memiliki kesibukan dalam kesehariannya. Salah satu aplikasi yang berkembang dalam bidang *mobile banking* adalah BCA Mobile. Berdasarkan jumlah unduhan pada toko aplikasi atau Playstore sampai dengan bulan Januari 2019, aplikasi BCA Mobile merupakan aplikasi yang paling banyak di unduh dibandingkan dengan aplikasi serupa lainnya yaitu sebanyak 239.392 kali.

Setiap aplikasi *mobile* mengizinkan pengguna untuk memberikan ulasan terhadap pelayanan dan fitur yang diberikan oleh aplikasi *mobile* tersebut. Ulasan pengguna sebelumnya sangat bermanfaat bagi calon pengguna aplikasi yang ingin menggunakan aplikasi tersebut. Banyak sedikitnya ulasan dapat memengaruhi keputusan untuk menggunakan atau tidak menggunakan aplikasi tersebut. Selain itu, ulasan juga bermanfaat bagi pengembang aplikasi dalam melihat kepuasan konsumen. Maka dari itu, dengan membaca ulasan yang ada diharapkan pengembang aplikasi dapat mengetahui apa yang harus ditingkatkan atau diperbaiki.

Dari aplikasi *mobile* yang ada saat ini, tidak ditemukan adanya fitur analisis sentimen untuk mengelompokkan atau menyaring antara ulasan positif dan negatif. Untuk mengetahui ulasan tersebut tergolong positif atau negatif, maka diperlukan adanya analisis sentimen. Analisis sentimen merupakan bidang ilmu yang menganalisis opini, evaluasi, sentimen, emosi, perilaku, maupun penilaian melalui entitas yang dapat berupa pelayanan, produk, individu, organisasi, isu-isu, kejadian, dan topik (Liu, 2016). Ulasan yang diperoleh dari pengguna layanan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

aplikasi *mobile* diklasifikasikan untuk menentukan ulasan tersebut termasuk positif atau negatif.

Beberapa metode telah dikembangkan dalam pengklasifikasian teks. Salah satu metode tersebut adalah *K-Nearest Neigbor* (K-NN). Algoritme KNN merupakan metode yang digunakan dalam proses klasifikasi terhadap suatu objek berdasarkan dari data yang memiliki jarak terdekat dengan objek tersebut (Nugraha, Faraby dan Adiwijaya, 2015). Algoritme ini bertujuan mengklasifikasikan objek berdasarkan atribut dan sampel data latih. Algoritme ini bekerja dengan menggunakan titik *query*, dan akan menemukan sejumlah *k* obyek atau titik latih yang paling dekat dengan titik *query* tersebut (Hardiyanto, Rahutomo dan Puspitasari, 2016). Algoritme ini memiliki kelemahan yaitu ditetapkannya nilai *k-values* pada semua kategori tanpa memperhitungkan jumlah dokumen latih yang dimiliki masing-masing kategori, sedangkan distribusi dokumen latih dalam data *training* tidak sama. Namun kelemahan ini dapat diatasi dengan menggunakan algoritme *Improved K-Nearest Neighbor*. Penelitian oleh Baoli, Shiwen, dan Qin (2003) menunjukkan bahwa dengan algoritme *Improved K-Nearest Neighbor*, dapat mendapatkan kestabilan pada proses klasifikasi dengan *k-values* yang bervariasi (Baoli, Shiwen dan Qin, 2003). Dalam metode ini, *k-values* yang dimiliki pada setiap kategori disesuaikan dengan jumlah kategori suatu dokumen latih. Sehingga ketika *k-values* semakin tinggi, hasil kategori tidak mendapat pengaruh dari kategori yang memiliki jumlah dokumen latih yang lebih besar (Putri, Ridok dan Indriati, 2013).

Sementara itu dalam pemeringkatan kata, metode yang sering digunakan yaitu metode BM25. Metode ini dikembangkan oleh *City University London* dan didasarkan pada model probabilistik dasar yang mengurutkan dokumen dalam urutan menurun terhadap nilai relevansi sebuah dokumen terhadap informasi yang dibutuhkan. BM25 memeringkatkan dokumen berdasarkan probabilitas dan menggunakan *term frequency* untuk mengurutkan *similarity* (Purwanto, Maulana dan A, 2011). Hasil penelitian yang dilakukan oleh Pardede (2015), menunjukkan bahwa metode BM25 lebih efektif dalam melakukan pemeringkatan dokumen. Berdasarkan hasil pengujian yang dilakukan, metode BM25 memiliki waktu pemrosesan lebih cepat dibandingkan dengan metode *Probabilistic Latent Semantic Analysis* (PLSA). Selain itu, dilihat dari nilai rata-rata *f-measure,* metode BM25 memiliki akurasi lebih tinggi dibandingkan metode PLSA (Pardede, Husada dan Riansyah, 2015).

Dari hasil uraian-uraian penelitian yang telah dilakukan, maka diperlukan adanya sistem yang dapat melakukan analisis sentimen pada suatu ulasan pada aplikasi *mobile*. Oleh karena itu, penulis ingin membuat suatu sistem yang dapat mengelompokkan ulasan berbahasa Indonesia pada ulasan aplikasi BCA Mobile yang menggabungkan metode pemeringkatan BM25 dan *Improved K-Nearest Neighbor*. Dengan metode tersebut diharapkan sistem ini dapat menghasilkan nilai akurasi yang tinggi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijabarkan tentang analisis sentimen pada ulasan aplikasi BCA Mobile, maka penulis merumuskan beberapa pokok permasalahan sebagai berikut.

1. Bagaimana perancangan algoritme BM25 dan *Improved K-Nearest Neighbor* untuk permasalahan analisis sentimen pada ulasan aplikasi BCA Mobile?
2. Bagaimana hasil pengujian yang diperoleh dari penerapan algoritme BM25 dan *Improved K-Nearest Neighbor* untuk permasalahan analisis sentimen pada ulasan aplikasi BCA Mobile menggunakan *precision, recall,* dan *f-measure*?

1.3 Tujuan

Tujuan yang ingin dicapai oleh penulis dalam analisis sentimen pada ulasan aplikasi BCA Mobile adalah sebagai berikut.

1. Mengetahui perancangan algoritme BM25 dan *Improved K-Nearest Neighbor* untuk permasalahan analisis sentimen pada ulasan aplikasi BCA Mobile.
2. Mengetahui hasil pengujian yang diperoleh dari penerapan algoritme BM25 dan *Improved K-Nearest Neighbor* untuk permasalahan analisis sentimen pada ulasan aplikasi Mobile menggunakan *precision, recall,* dan *f-measure*.

1.4 Manfaat

Manfaat dari perancangan sistem analisis sentimen pada ulasan aplikasi *mobile* adalah sebagai berikut.

1. Menerapkan ilmu dan pengetahuan yang telah didapatkan penulis selama menempuh perkuliahan.
2. Dapat membantu para pengembang mengetahui apa yang diinginkan oleh pengguna aplikasi sehingga dapat melakukan peningkatan fitur dan kualitas aplikasi *mobile*.
3. Sebagai bahan pertimbangan pengguna dalam menentukan kualitas dari suatu aplikasi *mobile*.

1.5 Batasan masalah

Untuk memfokuskan penelitian yang akan dilakukan, ruang lingkup atau batasan permasalahan yang digunakan dalam perancangan sistem analisis sentimen pada ulasan aplikasi BCA Mobile adalah sebagai berikut.

1. Data yang digunakan untuk analisis sentimen diambil dari ulasan yang diberikan terhadap aplikasi BCA Mobile melalui toko aplikasi (Google Play Store).
2. Data yang digunakan berupa data sekunder sebanyak 500 data.
3. Data berupa ulasan dengan Bahasa Indonesia.
4. Kelas pada analisis sentimen terdiri dari dua kelas yaitu positif dan negatif.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1.6 Sistematika pembahasan

Sistematika pembahasan yang akan dilakukan dalam penelitia ini terbagi atas enam bab dengan beberapa subbab. Sistematika pembahasan pada penelitian sistem analisis sentimen pada ulasan aplikasi BCA Mobile adalah sebagai berikut.

BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, manfaat, ruang lingkup atau batasan masalah, dan sistematika pembahasan dari penelitian analisis sentimen pada ulasan aplikasi BCA Mobile.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tentang uraian teori-teori dan penelitian yang sebelumnya dilakukan yang diperoleh dari berbagai referensi. Hasil dari referensi tersebut akan digunakan sebagai landasan dalam penelitian ini.

BAB 3 METODOLOGI

Bab ini metodologi menguraikan tentang metode atau teknik yang akan dilakukan dalam menyelesaikan penelitian analisis sentimen ini. Metodologi dapat berupa teknik pengumpulan data, maupun strategi penelitian yang akan dilakukan.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini menguraikan tentang cara penerapan metode BM25 dan *Improved KNN* berupa alur tahapan yang dijelaskan dengan diagram. Selain itu, dijelaskan pula hasil dari implementasi sistem berupa kode program yang telah dibuat.

BAB 5 HASIL DAN PEMBAHASAN

Bab ini memuat hasil dari sistem yang merupakan penjelasan tentang evaluasi yang dilakukan dalam penelitian analisis sentimen ini.

BAB 6 PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan dari hasil sistem dan saran yang dapat digunakan dalam penelitian selanjutnya yang didasari oleh penelitian ini.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang referensi penelitian yang sebelumnya pernah dilakukan berhubungan dengan analisis sentimen terhadap ulasan aplikasi BCA Mobile. Terdapat pula penjabaran tentang teori pendukung yang diperoleh dari literatur ilmiah yang membahas metode BM25 dan *Improved K-Nearest Neighbo*r yang dapat diterapkan dalam sistem analisis sentimen pada ulasan aplikasi BCA Mobile.

2.1 Kajian Pustaka

Landasan kepustakaan yang digunakan pada penelitian terdiri dari beberapa penelitian. Penelitian pertama membahas metode pemeringkatan BM25. Penelitian ini dilakukan oleh Jasman Pardede, Milda Gustiana Husada, dan Rizky Riansyah (2015). Penelitian ini bertujuan membandingkan kinerja metode BM25 dengan metode *Probabilistic Latent Semantic Analysis* (PLSA). Dalam penelitian tersebut, dokumen yang digunakan merupakan dokumen berbahasa Indonesia yang memiliki beberapa format dokumen yaitu *.doc, *.docx dan *.pdf. Tahapan awal yang dilakukan yaitu *pre-processing* yang terdiri dari *tokenizing, stopword removal,* dan *stemming*. Tahapan selanjutnya, sistem yang dibuat akan memeriksa *term query* yang terdapat pada masing-masing dokumen, kemudian sistem akan menghitung bobot dokumen dengan menggunakan persamaan BM25 dan PLSA. Proses terakhir yang dilakukan yaitu pemeringkatan dokumen mana yang memiliki nilai lebih besar berdasarkan *query* yang dimasukkan. Dari penelitian tersebut didapatkan kesimpulan bahwa metode Okapi BM25 lebih efektif dalam melakukan pemeringkatan dokumen dibandingkan dengan metode PLSA. Setelah dilakukan pengujian menggunakan *precision, recall,* dan *f-measure*, menunjukkan bahwa nilai *precision* metode BM25 lebih besar dibandingkan dengan metode PLSA. Hasil akurasi yang didapat mencapai 61,6% untuk rata-rata *f-measure* pada metode BM25 dan sebesar 56,9% pada metode PLSA.

Penelitian selanjutnya yang membahas metode klasifikasi *K Nearest Neighbor* (KNN) adalah penelitian yang dilakukan oleh Erik Hardiyanto, Faisal Rahutomo, dan Dwi Puspitasari (2016). Penelitian tersebut dilakukan dengan menggunakan metode KNN sebagai metode untuk klasifikasi, dan metode TF IDF untuk pembobotan. Data yang digunakan adalah artikel wikipedia berbahasa Indonesia. Langkah awal yang dilakukan yaitu dengan melakukan *pre-processing* yang terdiri dari *case folding, tokenizing*, dan *filtering* lalu dilakukan proses klasifikasi menggunakan metode KNN. Penelitian tersebut memperoleh hasil bahwa metode KNN memiliki kekurangan, yaitu akurasi kebenaran yang didapatkan rendah hanya sebesar 60%. Rendahnya akurasi tersebut disebabkan oleh beberapa faktor, salah satunya yaitu jumlah artikel yang dijadikan *centroid* terlalu sedikit. *Centroid* yang dimaksud merupakan rata-rata nilai tetangga terdekat.

Penelitian lainnya yaitu penelitian yang dilakukan oleh Prima Arfianda Putri, Achmad Ridok, dan Indriati (2013). Penelitian ini menggunakan metode *Improved K-Nearest Neighbor,* untuk mengklasifikasikan analisis sentimen pada *tweet* yang

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

mengelompokkan *tweet* menjadi positif dan negatif. Metode ini dipilih karena mampu menentukan *k-values* yang tepat sehingga menghasilkan akurasi yang tinggi. Algoritme *Improved K-Nearest Neighbor* melakukan modifikasi dalam penentuan *k-values* dalam penerapannya. Pada algoritme ini tetap dilakukan penetapan *k-values*, hanya saja tiap kelas memiliki *k-values* yang berbeda sesuai dengan jumlah data latih. Sehingga ketika *k-values* semakin tinggi, tidak memengaruhi kelas yang memiliki jumlah data latih yang lebih besar. Langkah awal yang dilakukan pada penelitian ini yaitu *pre-processing*. Proses tersebut terdiri dari pembersihan dokumen, *parsing, tokenizing, filtering* atau *stopword removal*, dan *stemming*. Kemudian dilakukan proses pembobotan (*Term Weighting*) menggunakan metode TF-IDF dan *cosine similarity*, lalu dilakukan klasifikasi menggunakan metode *Improved K-Nearest Neighbor.* Hasil yang diperoleh dengan metode tersebut menghasilkan akurasi 82,3% untuk rata-rata p*recision,* 86,5% untuk rata-rata *recall,* dan 84,3% untuk rata-rata *f-measure.*

Terdapat penelitian lain yang membandingkan metode *Naive Bayes, Fuzzy KNN,* dan *Ensemble Classifier* yang dilakukan oleh Julia Ika Ratna Pramastuti (2016). Penelitian ini bertujuan melakukan analisis sentimen dengan menggunakan 1.500 data *review* dari aplikasi *mobile.* Langkah awal penelitian dilakukan *pre-processing* yang terdiri dari *case folding, tokenizing, filtering,* dan *stemming*. Langkah selanjutnya yaitu pembobotan kata dengan menggunakan metode TF-IDF. Penelitian ini membandingkan hasil analisis sentimen menggunakan beberapa metode yaitu metode *Naive Bayes, Fuzzy KNN,* dan *Ensemble Classifier.* Penelitian tersebut menggunakan empat metode pengujian yaitu uji pengaruh nilai *k* untuk metode *Fuzzy KNN,* uji pengaruh perubahan data latih dan data uji, uji pengaruh metode klasifikasi, dan uji pengaruh metode *Ensemble.* Hasil uji pengaruh nilai *k* pada metode *Fuzzy KNN* menghasilkan akurasi rata-rata 90%. Hasil uji pengaruh perubahan data latih dan data uji menghasilkan rata-rata 91,49%. Hasil yang diperoleh menunjukkan bahwa metode *Ensemble* yaitu *Average Of Probabilities* sangat dipengaruhi oleh nilai probabilitas hasil klasifikasi dari seluruh metode yang digunakan. Hasil pengujian yang dilakukan menunjukkan bahwa akurasi yang diperoleh dari ketiga metode tersebut memiliki nilai yang kompetitif.

Dengan adanya penelitian-penelitian sebelumnya, penulis berasumsi bahwa dengan penggabungan beberapa penelitian sebelumnya yaitu pembuatan sistem analisis sentimen menggunakan metode BM25 untuk pemeringkatan dan metode *Improved K-Nearest Neighbor* untuk pengklasifikasian dapat digunakan dalam pembuatan sistem analisis sentimen ulasan aplikasi BCA Mobile.

2.2 Ulasan

Ulasan yang berasal dari kata ulas atau bisa disebut juga dengan kupasan, tafsiran atau komentar menurut Kamus Besar Bahasa Indonesia merupakan suatu tanggapan atas sebuah kejadian. Ulas dapat diartikan sebagai mengulas sesuatu, hal ini juga dapat diartikan dengan mengupas kekurangan dan keunggulan, atau menafsir sesuatu serta memberikan komentar terhadap sesuatu secara objektif.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Ulasan pada suatu produk atau aplikasi dianggap penting karena sebagian besar dari pengguna akan melihat terlebih dahulu ulasan yang diberikan konsumen terlebih dahulu sebelum menggunakan produk tersebut (Febrianti, Indriati dan Widodo, 2018).

2.3 Analisis Sentimen

Analisis sentimen yang merupakan bagian dari *opinion mining*, yaitu proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi (Putri, Ridok dan Indriati, 2013). Informasi yang berupa tekstual tersebut terbagi menjadi fakta dan opini. Fakta adalah suatu ekspresi terhadap suatu objek yang bersifat objektif, sedangkan opini biasanya menggambarkan suatu sentimen, penilaian atau perasaan seseorang pada sebuah objek yang bersifat subjektif. Penelitian dalam bidang *text mining* lebih banyak terfokus pada informasi faktual seperti *information retrieval*, sistem temu kembali, pengelompokan teks dan penelitian dalam *Natural Language Processing* (NLP) lainnya (Liu, 2016).

Analisis sentimen sendiri merupakan pengelompokan teks yang berupa informasi tekstual yang bersifat opini. Maka dari itu sifat dari analisis sentimen adalah subjektif terhadap suatu hal. Yang dimaksud subjektif disini adalah dapat berupa positif atau negatif. Informasi tekstual yang dikelompokkan menjadi positif atau negatif akan mengandung suatu nilai. Nilai tersebut yang selanjutnya akan digunakan sebagai parameter dalam menentukan suatu keputusan terhadap suatu dokumen (Indriati dan Ridok, 2016).

2.4 *Text Mining*

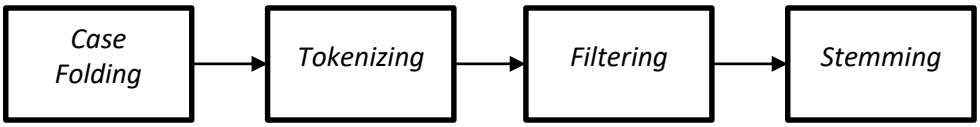
Text mining adalah proses menambang data berupa teks yang sumber datanya biasanya didapat dari suatu dokumen, yang tujuannya yaitu mencari kata-kata yang dapat mewakili isi dokumen sehingga dapat dilakukan analisis keterhubungan antara dokumen (Sianturi, 2014). *Text mining* merupakan bagian dari *data mining*, akan tetapi proses *text mining* memerlukan lebih banyak tahapan dibanding *data mining.* Hal ini dikarenakan data teks memiliki karakteristik yang lebih kompleks daripada data biasa maupun data yang sudah terstruktur. Karena data yang digunakan kebanyakan merupakan data tidak terstruktur, maka proses *text mining* memerlukan proses mengubah data tidak terstruktur menjadi data terstruktur pada awal prosesnya (Indriati dan Ridok, 2016).

Tahapan proses pada *text mining* terbagi menjadi beberapa bagian. Proses-proses tersebut yaitu yang pertama adalah pengumpulan data yang tidak tertruktur (*collecting unstructured* data), pemrosesan tahap awal dan pembersihan terhadap teks (*text processing and cleaning),* tahap pemrosesan (*processing and controlling*), tahap menganalisis pola (*pattern analysis*) dan pemrosesan informasi yang berharga dan relevan (*valuable information processing*). Suatu dokumen teks merupakan *input* dari proses dan menghasilkan *output* berupa pola sebagai hasil interpretasinya (Talib et al., 2016).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.5 *Pre-processing*

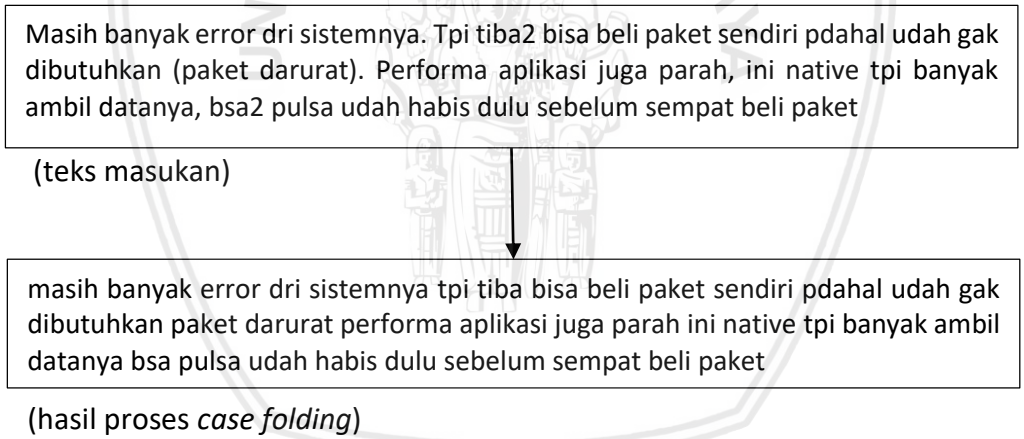
Tahap *pre-processing* merupakan tahapan awal yang digunakan untuk mendapatkan data yang akan digunakan dalam proses selanjutnya. Tahapan dalam *pre-processing* yang terdiri dari proses *case folding, tokenizing, filtering*, dan *stemming* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tahapan *Pre-processing*

2.5.1 *Case Folding*

Case folding merupakan tahap awal *pre-processing* yang digunakan dalam penelitian ini. Berfungsi mengubah huruf awal pada suatu kata masukan menjadi huruf kecil (*lower case*). Setelah dilakukan proses *case folding,* maka dilakukan proses selanjutnya, yaitu *tokenizing.* Gambar 2.2 merupakan contoh dari *case folding.*



Gambar 2.2 Contoh *case folding*

2.5.2 *Tokenizing*

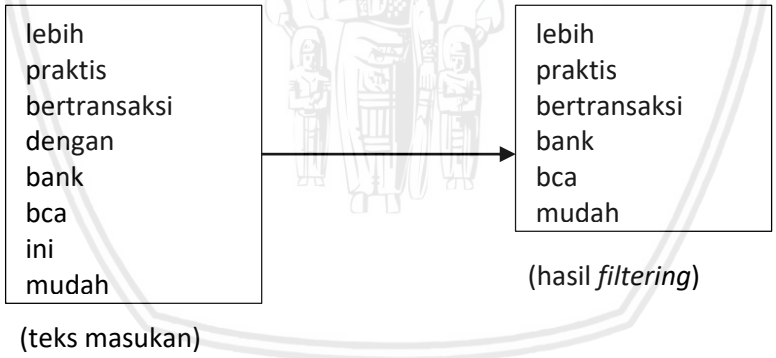
Tokenizing merupakan proses pemisahan kalimat menjadi kata atau token yang merupakan penyusun suatu dokumen. Pada tahap ini, karakter-karakter pemisah kata dan karakter yang tidak diperlukan akan dihilangkan karena tidak memiliki pengaruh terhadap pemrosesan teks. Karakter-karakter yang dimaksud bisa berupa tanda baca, angka, spasi, dan karakter selain huruf. Proses ini juga dilakukan berdasarkan tahap *pre-processing* sebelumnya (Indriati dan Ridok, 2016). Contoh dari proses *tokenizing* ditunjukkan pada Gambar 2.3.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 2.3 Contoh *tokenizing*

2.5.3 *Filtering*

Filtering merupakan proses menghapus kata yang tidak mengandung arti yang bermakna. Proses ini dilakukan berdasarkan hasil *tokenizing*. Tahapan *filtering* dapat dilakukan dengan dua cara yaitu berdasarkan *stoplist* dan berdasarkan *wordlist*. *Stoplist* merupakan kumpulan kata-kata tidak penting. Pada penerapannya setiap kata dalam dokumen akan dibandingkan daftar *stoplist* dan akan dibuang dari hasil *tokenizing*. *Wordlist* adalah kumpulan kata-kata penting yang berpotensi mampu merepresentasikan suatu dokumen. Pada penelitian ini, proses *filtering* dilakukan dengan menggunakan *stopword list* Tala (Indriati dan Ridok, 2016). Contoh proses *filtering* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Contoh *filtering*

(Sumber : *stopword list* Tala)

2.5.4 *Stemming*

Proses *stemming* merupakan proses mengubah bentuk suatu kata menjadi kata dasar dan sesuai dengan struktur morfologi Bahasa Indonesia yang benar. Kata yang memiliki imbuhan berupa awalan ataupun akhiran akan dihapus sehingga akan didapatkan kata dasar. Maka dari itu, proses mengubah menjadi kata dasar ini sangat berguna dalam mengoptimalkan *pre-processing* teks (Herdiawan, 2015). *Library* yang digunakan merupakan *Sastrawi Stemmer.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.6 Metode BM25

Metode BM25 merupakan metode yang dikembangkan oleh City University London. BM25 didasari oleh model probabilistik yang melakukan pemeringkatan dokumen terhadap nilai relevansi sebuah dokumen terhadap informasi yang dibutuhkan. BM25 merupakan metode pemeringkatan yang digunakan untuk mengurutkan hasil kecocokan terhadap dokumen-dokumen, berdasarkan kata kunci atau *query* yang dimasukkan. Metode ini merupakan formula terbaik dalam kelas *best match*, dikarenakan formula ini efektif dan memiliki ketepatan dalam mengurutkan dokumen berdasarkan *query* yang dicari (Pardede, Husada dan Riansyah, 2015). Persamaan 2.1 merupakan persamaan metode BM25.

$$BM25 = \sum_{i=1}^{n} IDF(q_i) . \frac{f(q_i,D).(k_1+1)}{f(q_i,D)+k_1.(1-b+b.\frac{|D|}{avgdl})} \quad (2.1)$$

Keterangan:

$f(q_i, D)$: Banyaknya frekuensi kata pada dokumen *D.*
$|D|$: Banyaknya kata dalam suatu dokumen *D.*
$avgdl$: Rata-rata panjang dokumen dalam seluruh data.
k_1 : 1,2
b : 0,75

Perhitungan nilai (*Index Document Frequency*) IDF akan dijelaskan pada Persamaan 2.2.

$$IDF(q_i) = log \frac{N-n(q_1)+0,5}{n(q_1)+0,5} \quad (2.2)$$

Keterangan:

$IDF(q_i)$: nilai *Index Document Frequency.*
N : Total seluruh dokumen.
$n(q_i)$: Banyak dokumen yang mengandung *query*.

2.7 *Improved K-Nearest Neighbor*

Metode *Improved K-Nearest Neighbor* merupakan metode klasifikasi yang dilakukan dengan memodifikasi nilai *k-values*. Nilai *k-values* pada metode ini merupakan nilai yang ditetapkan di awal, namun dalam proses selanjutnya nilai *k-values* yang dimiliki tiap kelas menjadi berbeda. Perbedaan nilai *k-values* ini disesuaikan dengan jumlah dokumen latih berdasarkan kelasnya. Sehingga ketika *k-values* semakin tinggi, hasil tiap kelas tidak terpengaruh terhadap kelas yang memiliki jumlah dokumen latih yang lebih besar (Putri, Ridok dan Indriati, 2013).

Langkah awal yang dilakukan yaitu dengan pembobotan kata, lalu dilanjutkan dengan melakukan pemeringkatan hasil pembobotan tersebut, kemudian dilakukan perhitungan nilai *k-values* baru atau biasa disebut dengan nilai *n*. Persamaan 2.3 merupakan perhitungan dalam menentukan *k-values* baru.

$$n = \left[\frac{k*N(c_m)}{maks\{n(c_m)|j=1....N_c\}}\right] \quad (2.3)$$

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Keterangan:

n : nilai *k-values* baru.
k : nilai *k-values* yang ditetapkan di awal.
$N(c_m)$: jumlah dokumen/data latih pada kategori *m.*
$maks\{n(c_m)|j = 1 \ldots . N_c\}$: jumlah dokumen/data latih terbanyak pada semua kategori yang ada.

Hasil *k-values* baru atau nilai *n* digunakan sebagai dokumen yang terpilih pada tiap kelas yang merupakan *top n* dokumen atau dokumen teratas yaitu dokumen yang mempunyai nilai atau bobot paling besar di setiap kategorinya. Perhitungan pada Persamaan 2.4 menyatakan perhitungan nilai probabilitas dan diambil nilai maksimum dari hasil dari perhitungan probabilitas tersebut pada masing-masing kelas (Baoli, Shiwen dan Qin, 2003).

$$p(x, c_m) = argmax_m \frac{\sum_{d_j \in top\ n\ kNN(c_m)} sim(x, d_j) y(d_j.c_m)}{\sum_{d_j \in top\ n\ kNN(c_m)} sim(x, d_j)} \quad (2.4)$$

Keterangan:

$p(x, c_m)$: probabilitas dokumen *X* menjadi anggota kategori c_m.
$sim(x, d_j)$: kemiripan antara dokumen *X* dengan dokumen latih d_j.
$top\ n\ kNN$: *top n* tetangga.
$y(d_j.c_m)$: fungsi atribut dari kategori tertentu yang memenuhi Persamaan 2.5.

$$y(d_j.c_m) = \begin{cases} 1, d_j \in c_m \\ 0, d_j \notin c_m \end{cases} \quad (2.5)$$

Berdasarkan perhitungan pada Persamaan 2.4 maka akan diketahui nilai probabilitas terbesar. Hasil perhitungan probabilitas tersebut yang akan menjadi penentuan terhadap dokumen uji.

2.8 Evaluasi

Hasil dari klasifikasi yang dilakukan oleh sistem dapat dilakukan dengan menggunakan metode pengujian yang diukur dari tingkat akurasi sistem yang telah dibuat. Untuk mengetahui tingkat akurasi tersebut dilakukan beberapa cara yaitu dengan menggunakan *k-fold cross validation,* perhitungan nilai *precision, recall, f-measure* dan *accuracy*.

2.8.1 *K-Fold Cross Validation*

K-fold cross validation merupakan metode yang paling sederhana dan paling banyak digunakan untuk memprediksi kesalahan pada sistem. Metode ini dapat digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem. Proses yang dilakukan yaitu dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa data *input* atau data uji yang acak (Pandie, 2012). Nilai *K* yang digunakan dapat bervariasi agar dapat digunakan dalam memperkirakan tingkat kesalahan yang terjadi (Hastie,

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tibshirani dan Friedman, 2008). Gambar 2.5 merupakan gambaran dari *5-fold cross validation*.

1	2	**3**	4	5
Train	Train	**Validation**	Train	Train

Gambar 2.5 *K-fold Cross Validation*

2.8.2 Evaluasi *Precision, Recall, F-Measure,* Dan *Accuracy*

Evaluasi *precision, recall, f-measure* dan *accuracy* disajikan dalam *confusion matrix* pada Tabel 2.1. Evaluasi tersebut dilakukan dengan cara membandingkan kategori hasil aktual atau *ground truth* dengan kategori hasil prediksi sistem (Manning, Raghavan dan Schutze, 2009).

Tabel 2.1 *Confusion Matrix*

	relevant	***nonrelevant***
Retrieved	*True positives (tp)*	*False positives (fp)*
Non retrieved	*False negatives (fn)*	*True negatives (tn)*

Persamaan 2.6 merupakan rumus untuk menghitung nilai *precision. Precision* merupakan tingkat ketepatan antara data latih yang telah ditentukan kelasnya dengan hasil yang ditampilkan oleh sistem.

$$precision = \frac{tp}{tp+fp} \tag{2.6}$$

Persamaan 2.7 merupakan rumus untuk menghitung nilai *recall. Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali atau menampilkan sebuah informasi.

$$recall = \frac{tp}{tp+fn} \tag{2.7}$$

Persamaan 2.8 merupakan rumus untuk menghitung nilai *f-measure. F-measure* merupakan kombinasi dari perhitungan *precision* dan *recall.*

$$f - measure = 2 * \frac{precision*recall}{precision+recall} \tag{2.8}$$

Persamaan 2.9 merupakan rumus untuk menghitung nilai *accuracy. Accuracy* merupakan kedekatan antara nilai prediksi dengan nilai aktual dari sistem.

$$accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{2.9}$$

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Keterangan untuk Persamaan 2.6, Persamaan 2.7, Persamaan 2.8, dan Persamaan 2.9 adalah sebagai berikut.

tp : jumlah data positif yang terklasifikasi dengan benar oleh sistem.
tn : jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
fp : jumlah data positif namun terklasifikasi dengan salah oleh sistem.
fn : jumlah data negatif namun terklasifikasi dengan salah oleh sistem.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 3 METODOLOGI

Pada bab ini berisi tentang tipe penelitian, strategi penelitian, lokasi penelitian, peralatan pendukung, metode pengumpulan data, serta teknik penerapan algoritme yang digunakan dalam pembuatan sistem analisis sentimen pada ulasan aplikasi BCA Mobile dengan menggunakan metode BM25 dan *Improved K-Nearest Neighbor*.

3.1 Tipe Penelitian

Penelitian analisis sentimen pada ulasan aplikasi BCA Mobile dengan menggunakan metode BM25 dan *Improved K-Nearest Neighbor* bersifat non implementatif. Penelitian ini merupakan pelaksanaan tinjauan dari sebuah ilmu yang sudah ada sebelumnya. Sedangkan pendekatan yang digunakan pada penelitian ini adalah pendekatan analitik. Pendekatan analitik menjelaskan hubungan antar elemen dalam objek penelitian yaitu aplikasi BCA Mobile dengan sebuah analisis sentimen. Hasil dari penelitian ini akan menjawab pertanyaan-pertanyaan dalam rumusan masalah yang sudah dijabarkan pada bab sebelumnya.

3.2 Strategi Penelitian

Strategi penelitian yang dilakukan dalam analisis sentimen pada ulasan aplikasi BCA Mobile adalah dengan mengambil secara langsung ulasan suatu aplikasi. Studi kasus pada penelitian ini yaitu aplikasi BCA Mobile. Data yang digunakan berupa ulasan yang terdapat pada Play Store. Data ulasan yang digunakan merupakan 500 data ulasan sekunder.

3.3 Lokasi Penelitian

Lokasi penelitian analisis sentimen pada ulasan aplikasi BCA Mobile dengan menggunakan metode BM25 dan *Improved K-Nearest Neighbor,* dilakukan di Laboratorium Riset Fakultas Ilmu Komputer (FILKOM). Penelitian ini dilakukan di lingkungan Universitas Brawijaya. Penelitian dilakukan selama lima bulan.

3.4 Peralatan Pendukung

Untuk penerapan metode BM25 dan *Improved K-Nearest Neighbor* dalam analisis sentimen terhadap ulasan aplikasi BCA Mobile, diperlukan beberapa perangkat pendukung dengan spesifikasi sebagai berikut.

1. Perangkat lunak.
    1. Sistem Operasi Microsoft Office 2016
    2. *Python* versi 3.6.3
    3. Aplikasi editor *Spyder*
2. Perangkat keras.
    1. Processor Intel(R) Core(TM) i3-2370M CPU @ 2.40GHz

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

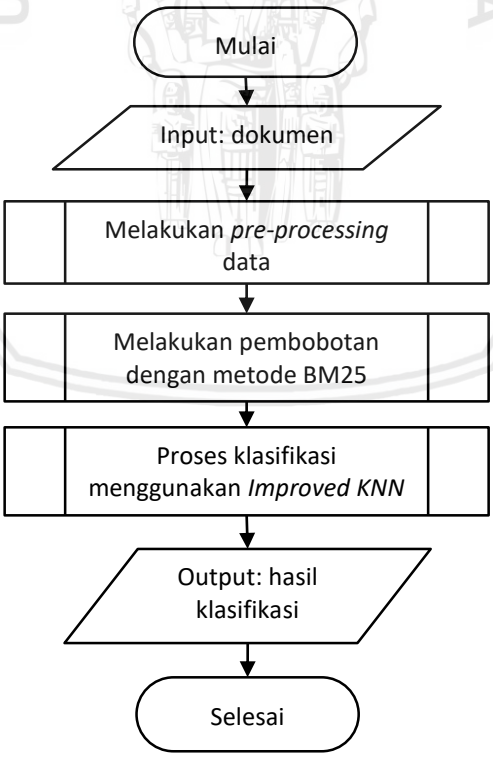
2. NVIDIA GEFORCE 610M 2GB
3. RAM 4GB

3.5 Pengumpulan Data

Pengumpulan data merupakan tahapan dimana peneliti mengumpulkan data berupa ulasan yang didapatkan dari ulasan yang diberikan pada aplikasi *mobile*. Aplikasi *mobile* yang digunakan sebagai objek dalam penelitian ini adalah aplikasi BCA Mobile. Data ulasan merupakan data sekunder sebanyak 500 data yang terdiri dari 200 data dengan kelas positif dan 300 data dengan kelas negatif yang disimpan dalam suatu dokumen teks. Data yang didapat kemudian dilakukan proses *labelling* oleh seorang pakar. Hal ini berguna dalam proses evaluasi yang akan dilakukan.

3.6 Teknik Penerapan Algoritme

Dalam proses perancangan pada penelitian analisis sentimen terhadap ulasan aplikasi BCA Mobile menggunakan metode BM25 dan *Improved KNN* terdapat beberapa tahapan yang harus dilakukan. Proses awal yang dilakukan yaitu dengan *pre-processing* data, selanjutnya dilakukan proses pembobotan kata menggunakan metode BM25, dan proses klasifikasi menggunakan metode *Improved KNN*. Proses tersebut dijelaskan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Teknik Penerapan Algoritme

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Diagram alir perancangan algoritme pada Gambar 3.1 merupakan gambaran umum sistem analisis sentimen. Dalam penerapannya, sistem akan terbagi menjadi dua proses. Proses pertama yaitu proses *training*. Proses ini dilakukan dengan menggunakan data *training* sebagai *input* proses. Langkah selanjutnya yaitu dengan melakukan proses *pre-processing* data. Proses tersebut terdiri dari beberapa sub proses, diantaranya *case folding, tokenizing, filtering,* dan *stemming.* Hasil dari proses tersebut disimpan ke dalam file berupa hasil *training* yang akan digunakan sebagai *input* dalam proses *testing*. Proses kedua setelah *training* dilakukan yaitu dengan melakukan proses *training.* Langkah-langkah yang dilakukan yaitu dengan menggunakan *input* dari hasil *training,* dokumen *testing,* dan dokumen *training.* Langkah kedua yaitu dengan melakukan *pre-processing* dengan alur yang sama dengan proses *training.* Langkah ketiga yaitu dengan melakukan proses pembobotan menggunakan metode BM25, dan langkah terakhir yaitu dengan melakukan klasifikasi dokumen menggunakan metode *Improved KNN.*

3.7 Penarikan Kesimpulan

Proses penarikan kesimpulan merupakan tahapan terakhir dalam menentukan analisis sentimen pada ulasan aplikasi BCA Mobile. Proses ini menunjukkan hasil berupa klasifikasi terhadap dokumen yang menunjukkan kelas dari dokumen tersebut. Kelas dari dokumen terdiri dari dua kelas yaitu positif dan negatif. Hasil tersebut diperoleh dari hasil perhitungan probabilitas dengan menggunakan perhitungan dari algoritme *Improved K-Nearest Neighbor.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini dijelaskan perancangan secara menyeluruh tentang sistem yang dibuat dan berisi tentang cara implementasi yang dilakukan dalam analisis sentimen ulasan aplikasi BCA Mobile menggunakan metode BM25 dan *Improved K-Nearest Neighbor*. Perancangan ini didasari oleh teori yang ada sebelumnya, dan akan dijadikan dasar dalam implementasi sistem.

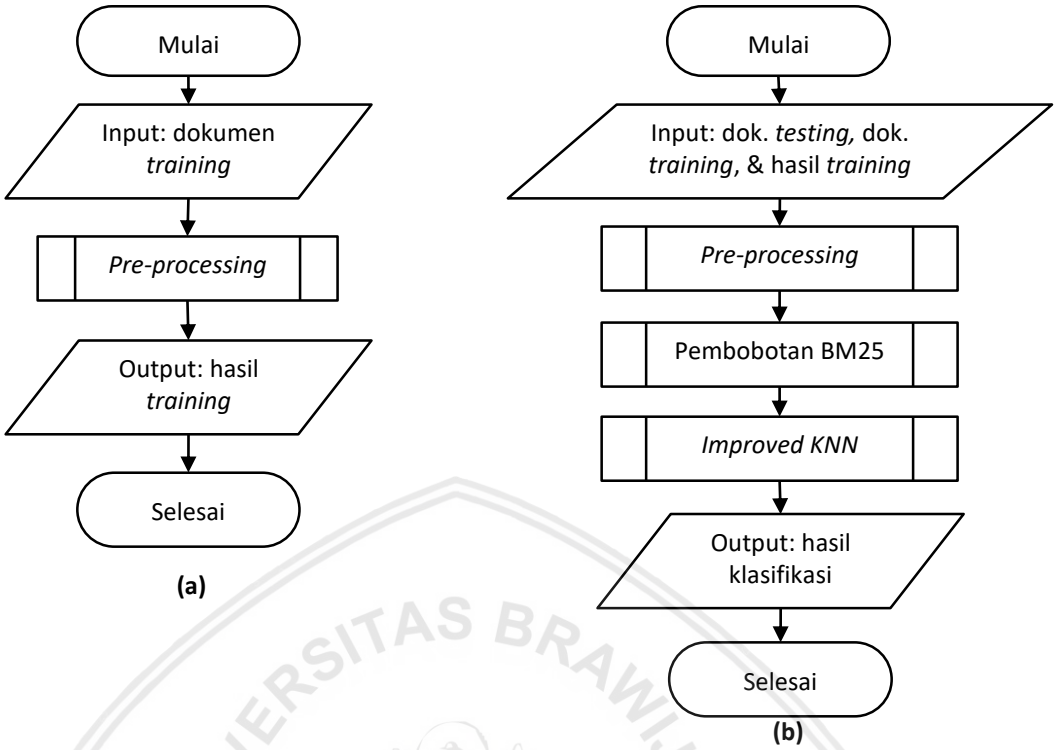
4.1 Deskripsi Permasalahan

Saat ini sangat banyak ulasan yang kita jumpai dalam penilaian suatu produk. Produk yang dimaksud dapat berupa suatu aplikasi, salah satunya yaitu aplikasi *mobile.* Aplikasi *mobile* banyak digunakan karena dianggap dapat mempermudah kegiatan sehari-hari. Maka dari itu, tak sedikit pula yang memberikan ulasan terhadap aplikasi yang sedang marak saat ini. Ulasan yang diberikan terhadap suatu produk tentunya sangat berguna bagi pengguna dalam menentukan bahwa aplikasi tersebut sudah cukup layak atau tidak untuk digunakan sehari-hari dalam memenuhi kebutuhan. Selain itu, pengembang juga dapat diuntungkan dengan mengetahui ulasan mana yang dianggap positif dan negatif. Apabila ulasan bersifat negatif maka pengembang dapat mengembangkan atau memperbaiki kembali fitur yang dimiliki oleh aplikasi *mobile* tersebut. Analisis sentimen sangat diperlukan dalam menangani hal tersebut dengan cara melakukan klasifikasi menggunakan metode *Improved K-Nearest Neighbor.* Cara pemeringkatan data dilakukan dengan menggunakan BM25. Hasil dari klasifikasi nantinya menghasilkan dua kelas atau kategori yaitu positif dan negatif.

4.2 Deskripsi Umum Sistem

Pada sub bab ini akan membahas alur sistem yang akan dibangun guna menyelesaikan permasalahan analisis sentimen yang terdapat pada ulasan aplikasi BCA Mobile dengan menggunakan metode BM25 sebagai metode pemeringkatan kata dan metode *Improved K-Nearest Neighbor* sebagai metode klasifikasi. Analisis sentimen ini mengklasifikasikan ulasan menjadi dua kelas yaitu positif dan negatif. Sebelum *input* dimasukkan untuk proses klasifikasi, harus melewati tahap *pre-processing* terlebih dahulu. Setelah tahap *pre-processing* selesai, dilakukan pemeringkatan kata menggunakan metode BM25. Setelah didapatkan peringkat berdasarkan dokumen data latih, maka proses selanjutnya yaitu menentukan *k-values* baru. Setelah diketahui nilai *k-values* baru, lalu dihitung nilai similaritas dokumen data latih dan data uji. Langkah selanjutnya yaitu menghitung nilai maksimum perbandingan antara kemiripan dokumen uji dengan dokumen latih. Kelas dari nilai maksimum tersebut akan menjadi kelas dokumen uji yang telah dimasukkan sebagai *input*. Alur perancangan sistem secara umum ditunjukkan pada Gambar 4.1.

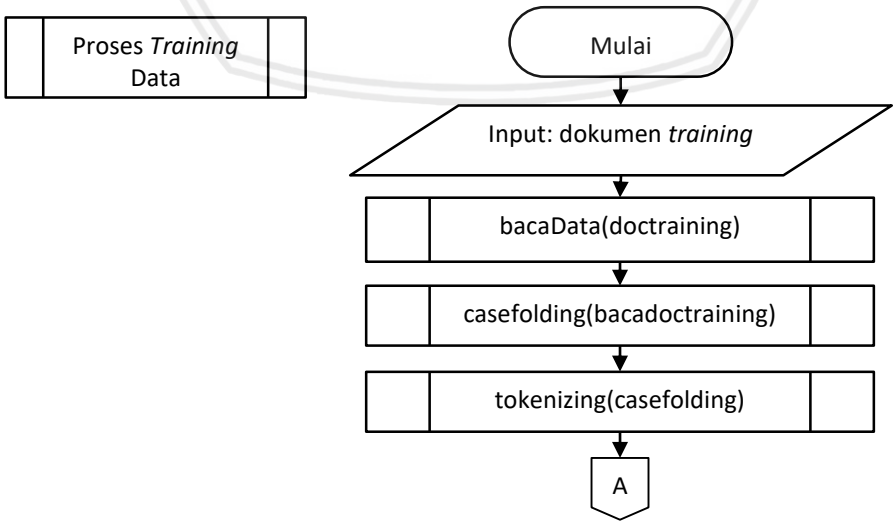
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



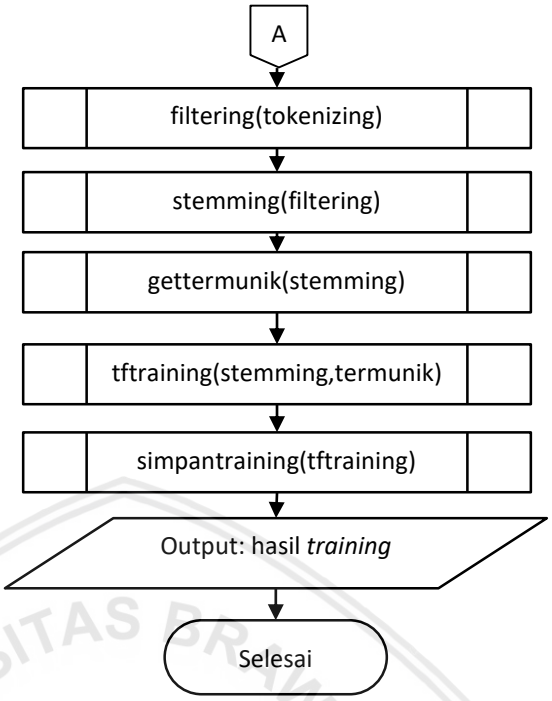
Gambar 4.1 Diagram Alir Perancangan Algoritme, (a) Proses *Training*, (b) Proses *Testing*

4.3 Proses *Training* Data

Proses *training* data merupakan proses yang digunakan untuk mengolah data *training.* Data yang digunakan yaitu berupa ulasan yang didapatkan dari Playstore. Tahapan yang dilakukan dalam proses ini yaitu dengan melakukan *pre-processing* data. Proses ini terdiri dari *case folding, tokenizing, filtering,* dan *stemming.* Tahapan proses tersebut ditunjukkan pada Gambar 4.2.



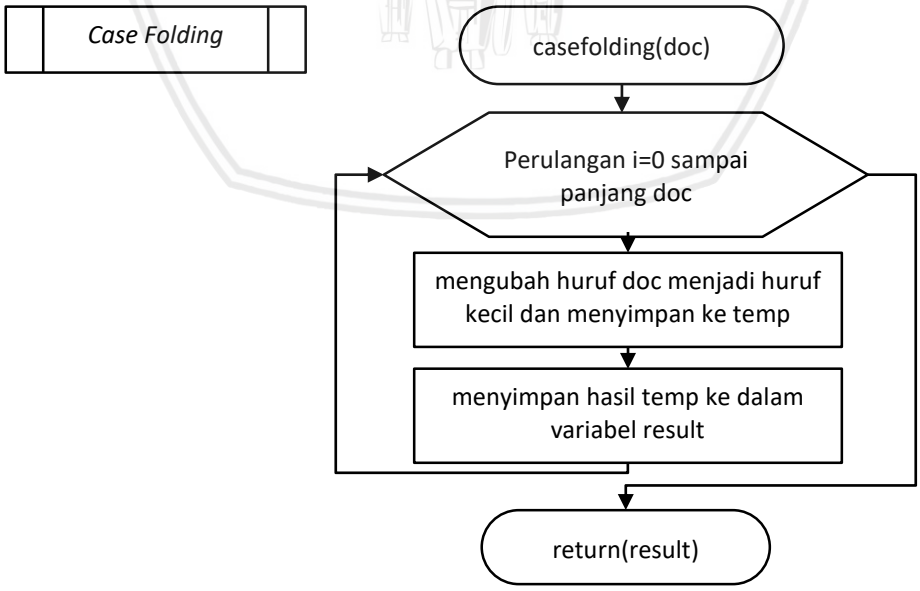
Gambar 4.2 Proses *Training* Data



Gambar 4.2 Proses *Training* Data (Lanjutan)

4.3.1 *Case folding*

Berdasarkan Gambar 4.2, tahapan yang dilakukan yaitu dimulai melalui proses *case folding*. Tahap ini berfungsi mengubah masukan yang berupa huruf awal pada suatu kata menjadi huruf kecil (*lower case*). Alur tahap *case folding* ditunjukkan pada Gambar 4.3.

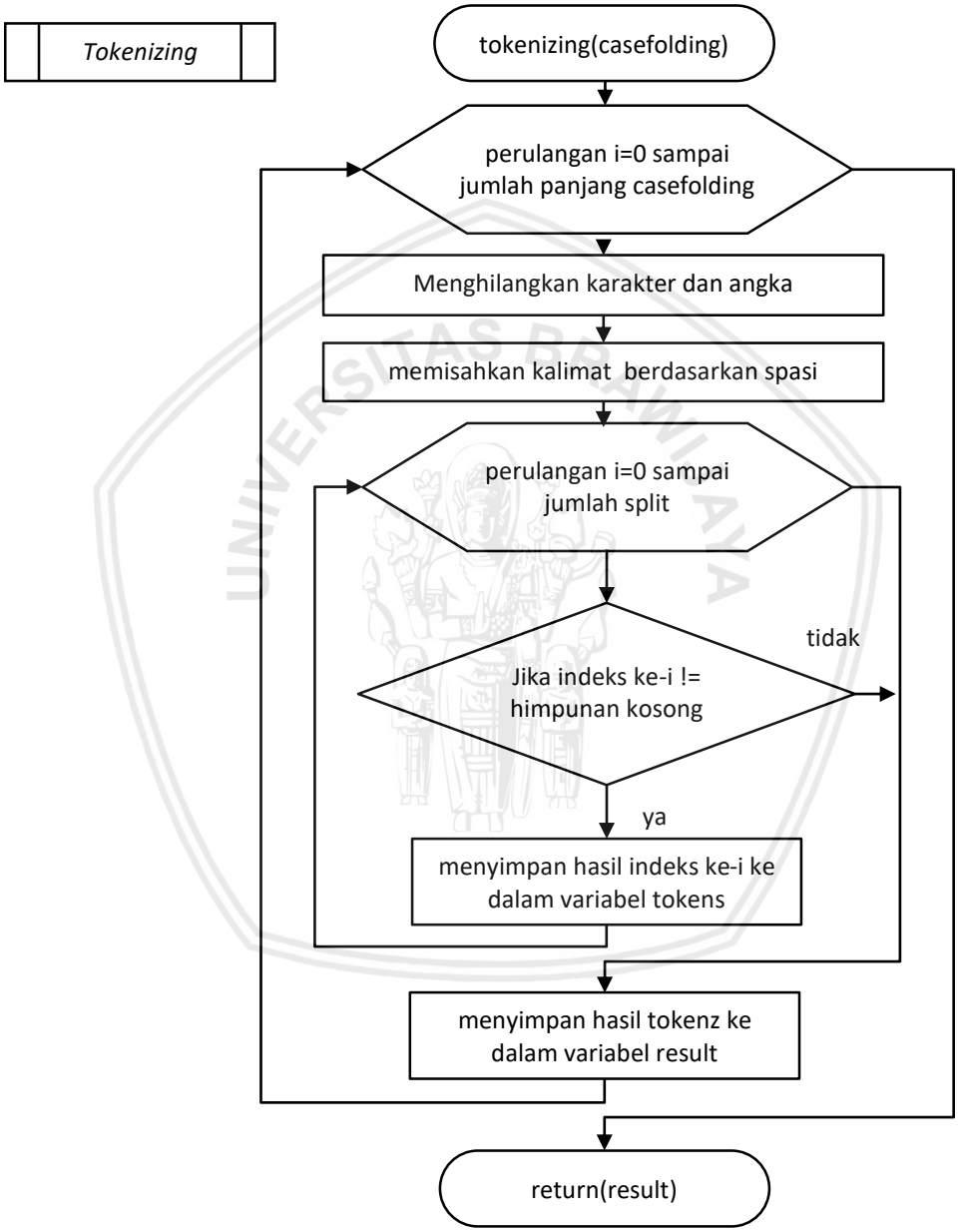


Gambar 4.3 Alur tahapan *Case Folding*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.3.2 *Tokenizing*

Berdasarkan alur tahapan *pre-prosessing*, langkah selanjutnya yaitu melakukan pemisahan kumpulan kalimat menjadi kumpulan kata. Tahapan ini disebut dengan tahap *tokenizing.* Tahapan ini menggunakan hasil dari proses *case folding* yang telah dilakukan sebelumnya. Alur tahapan proses *tokenizing* dapat dilihat pada Gambar 4.4.

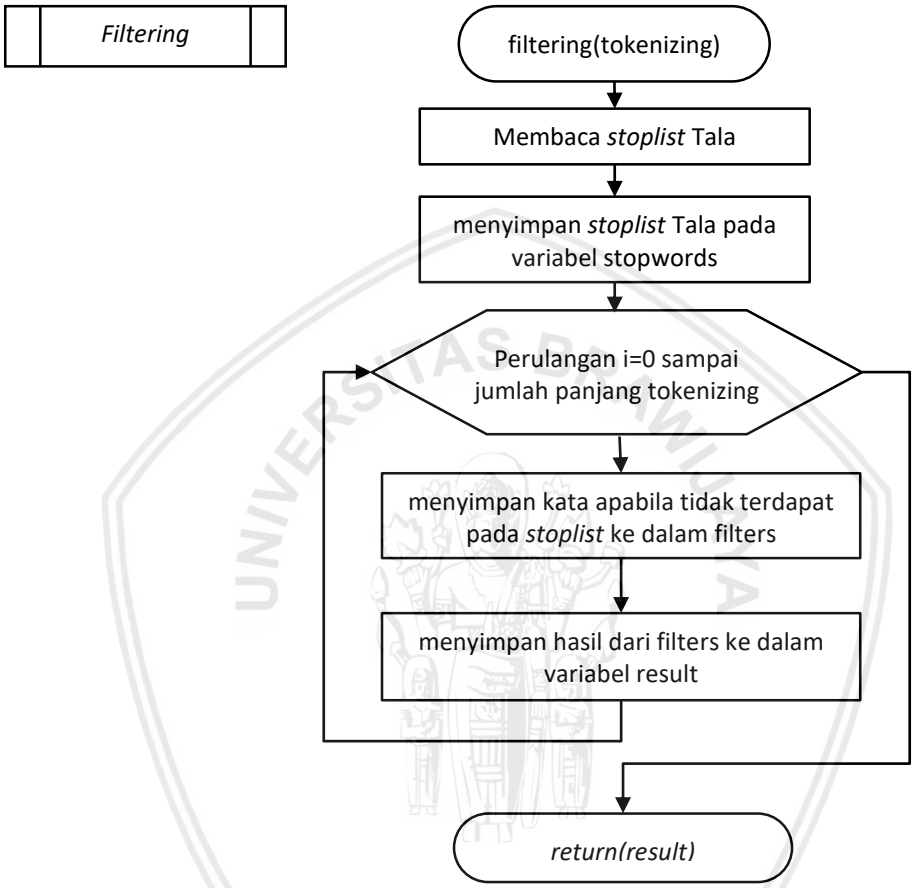


Gambar 4.4 Alur Tahapan *Tokenizing*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.3.3 *Filtering*

Filtering merupakan tahapan selanjutnya setelah dilakukan proses *tokenizing.* Proses ini dilakukan untuk menghapus kata yang sering muncul, namun tidak memiliki pengaruh dalam proses klasifikasi dalam analisis sentimen. Proses *filtering* dijalankan dengan *input* dari hasil *tokenizing.* Alur tahapan *filtering* dijelaskan pada Gambar 4.5.

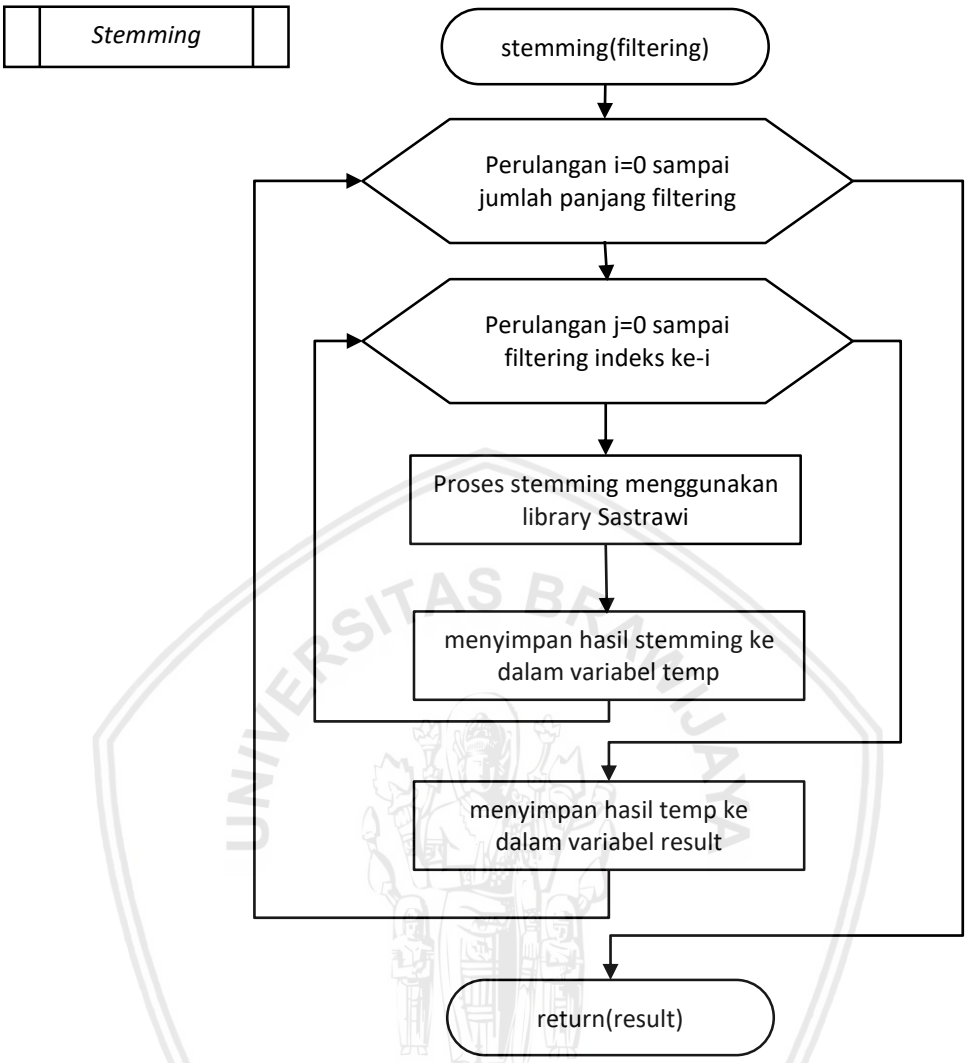


Gambar 4.5 Alur Tahapan *Filtering*

4.3.4 *Stemming*

Proses terakhir dalam *pre-processing* adalah *stemming.* Proses *stemming* menggunakan masukan berupa hasil dari proses sebelumnya, yaitu *filtering.* Proses ini akan mengubah kata yang memiliki imbuhan dan menguraikan bentuk kata tersebut menjadi kata dasar. Alur tahapan *stemming* dijelaskan pada Gambar 4.6.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

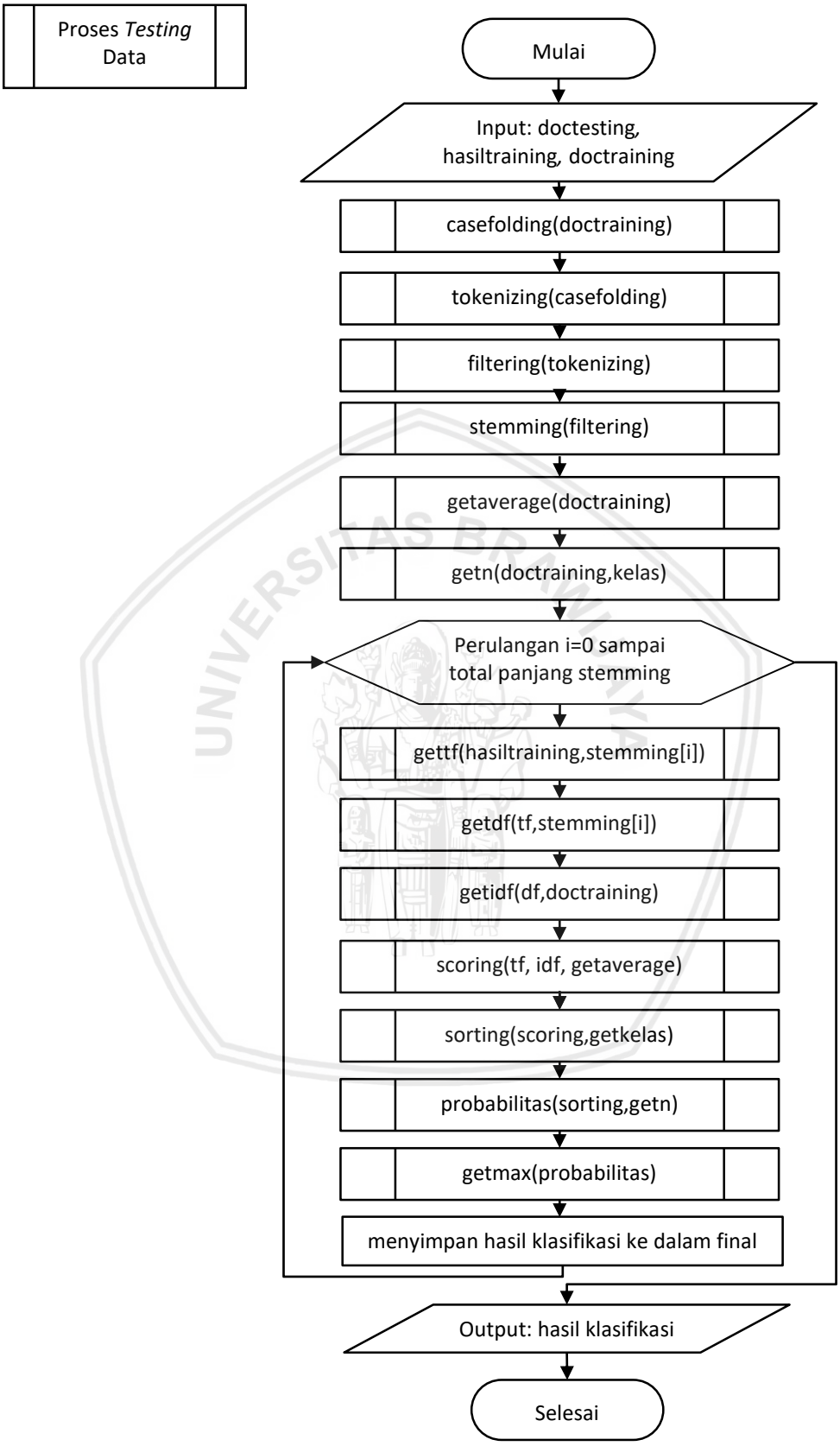


Gambar 4.6 Alur Tahapan *Stemming*

4.4 Proses *Testing* Data

Proses *testing* data merupakan proses yang digunakan untuk mengolah data uji atau data *testing.* Proses ini menggunakan hasil dari proses *training* sebagai acuan dalam proses pengerjaannya. Tahapan yang dilakukan yaitu *pre-processing* data, pembobotan menggunakan metode BM25, dan klasifikasi menggunakan metode *Improved KNN.* Proses *pre-processing* data pada *testing* sama dengan *pre-processing* data pada *training.* Proses tersebut terdiri dari *case folding, tokenizing, filtering,* dan *stemming.* Tahapan pembobotan kata meggunakan BM25 terdiri dari perhitungan TF, DF, IDF, dan *scoring* BM25 pada dokumen *training.* Lalu tahapan terakhir yaitu dengan melakukan klasifikasi dokumen *testing* menggunakan metode *Improved KNN.* Diagram alur proses *testing* data ditunjukkan pada Gambar 4.7.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.7 Alur Proses *Testing* Data

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.4.1 *Case Folding*

Tahap *case folding* merupakan sub proses pertama yang dilakukan dalam *pre-processing* data. Pada tahap ini, seluruh huruf pada awal kata diubah menjadi huruf kecil. Alur proses *case folding* dapat dilihat pada Gambar 4.3.

4.4.2 *Tokenizing*

Tahap *tokenizing* merupakan sub proses kedua setelah *case folding*. Tahap ini dilakukan untuk memisahkan kalimat menjadi suatu term atau token. Selain memisahkan kalimat menjadi token, dilakukan proses penghapusan angka dan tanda baca pada dokumen. Alur proses *tokenizing* dapat dilihat pada Gambar 4.4.

4.4.3 *Filtering*

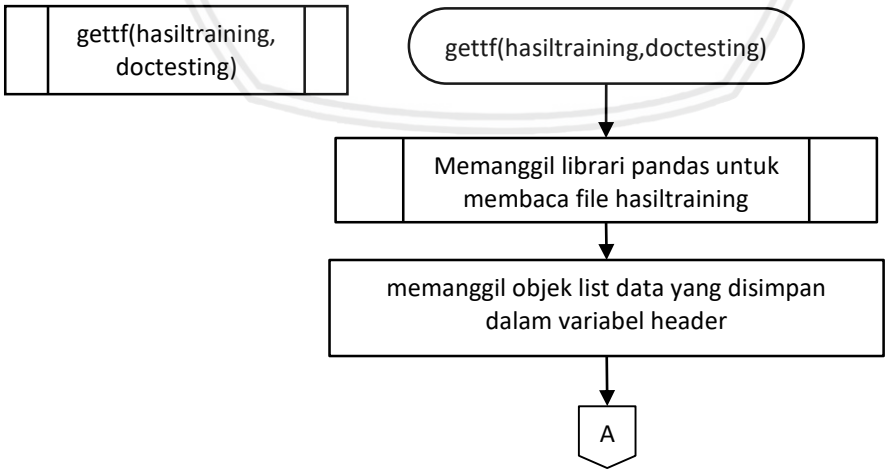
Tahap *filtering* merupakan sub proses ketiga dalam *pre-processing* data. Tahap *filtering* dilakukan untuk menghilangkan kata yang tidak berguna dalam pemrosesan teks. Kata yang tidak berguna tersebut disimpan dalam *stopword list.* Alur proses *filtering* dapat dilihat pada Gambar 4.5.

4.4.4 *Stemming*

Tahap *stemming* merupakan sub proses terakhit dalam *pre-processing* data. Tahap *stemming* dilakukan untuk mengubah suatu kata pada dokumen *testing* ke dalam kata dasar. *Library* yang digunakan dalam proses ini adalah *library* Sastrawi. Alur proses *stemming* dapat dilihat pada Gambar 4.6.

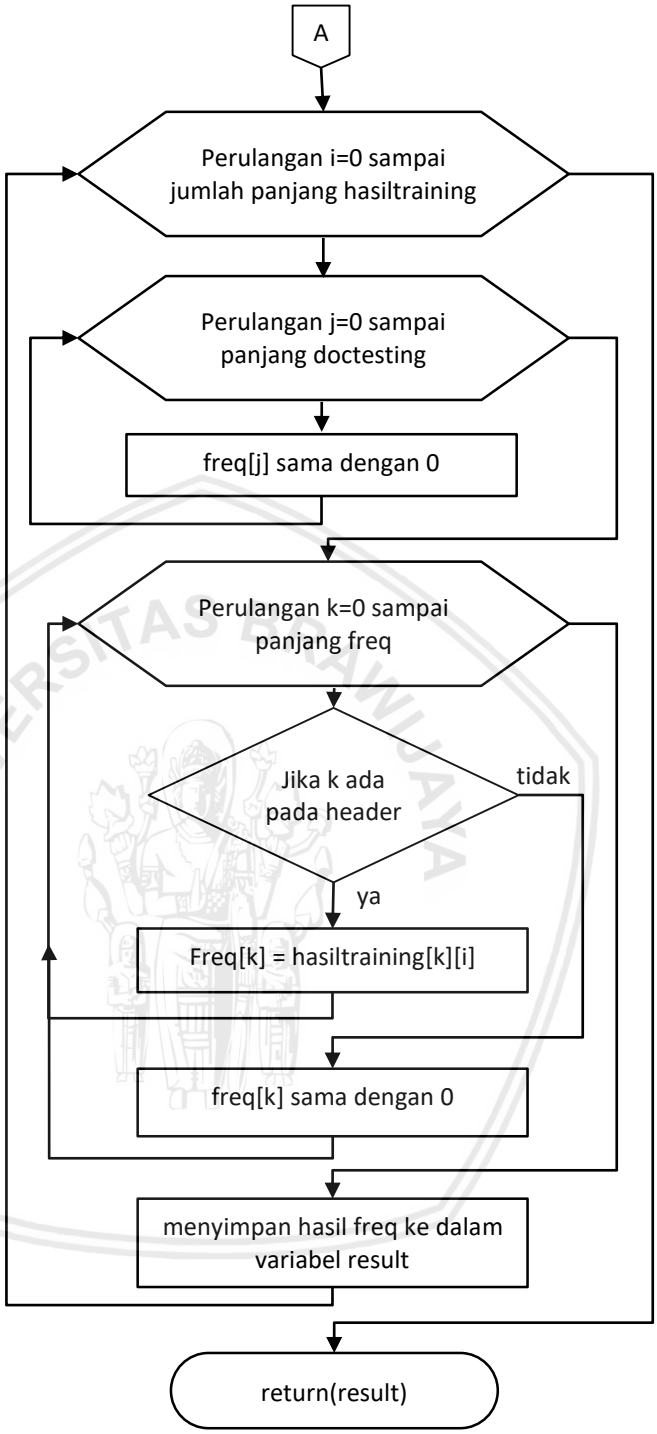
4.4.5 Perhitungan Nilai TF

Menghitung nilai TF merupakan tahapan awal dalam melakukan metode BM25. Dalam sub proses ini dilakukan perhitungan perhitungan term pada seluruh dokumen. Alur perancangan perhitungan nilai TF dijelaskan pada Gambar 4.8.



Gambar 4.8 Alur Perhitungan Nilai TF

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

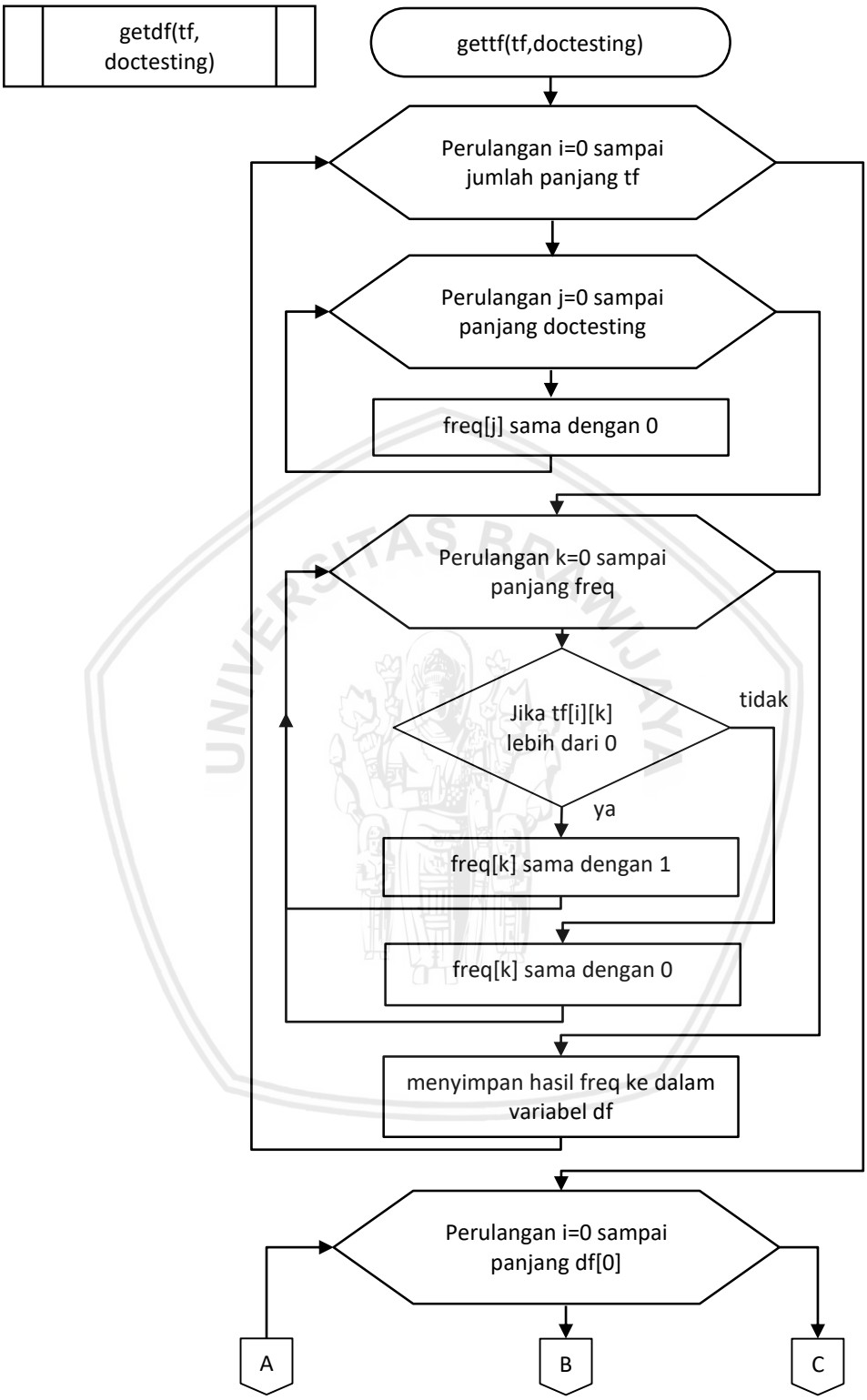


Gambar 4.8 Alur Perhitungan Nilai TF (Lanjutan)

4.4.6 Perhitungan Nilai DF

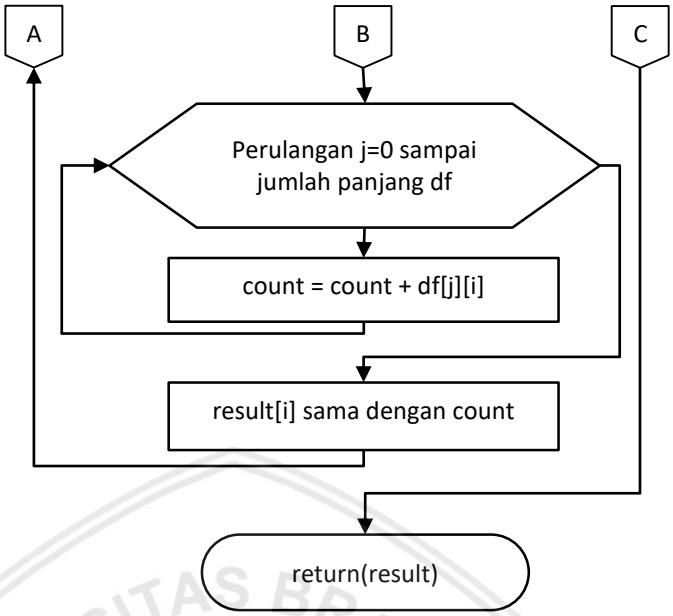
Sub proses setelah menghitung nilai TF adalah menghitung nilai DF. Proses ini merupakan proses menghitung kemunculan suatu term dalam dokumen data uji. Tahap ini dilakukan untuk mengetahui berapa jumlah suatu kata dalam tiap dokumen yang ada. Alur perhitungan nilai DF dijelaskan pada Gambar 4.9.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.9 Alur Perhitungan Nilai DF

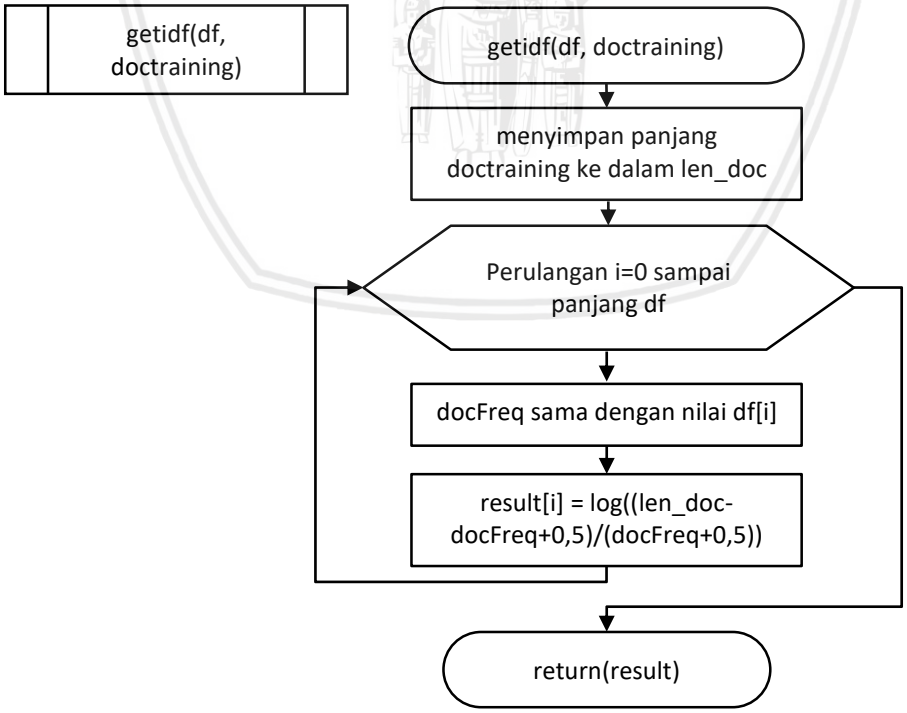
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.9 Alur Perhitungan Nilai DF (Lanjutan)

4.4.7 Perhitungan Nilai IDF

Sub proses selanjutnya setelah menemukan nilai TF dan DF adalah dengan menghitung nilai IDF. Perhitungan ini dilakkan dengan menggunakan Persamaan 2.2. Alur perhitungan nilai IDF dijelaskan pada Gambar 4.10.

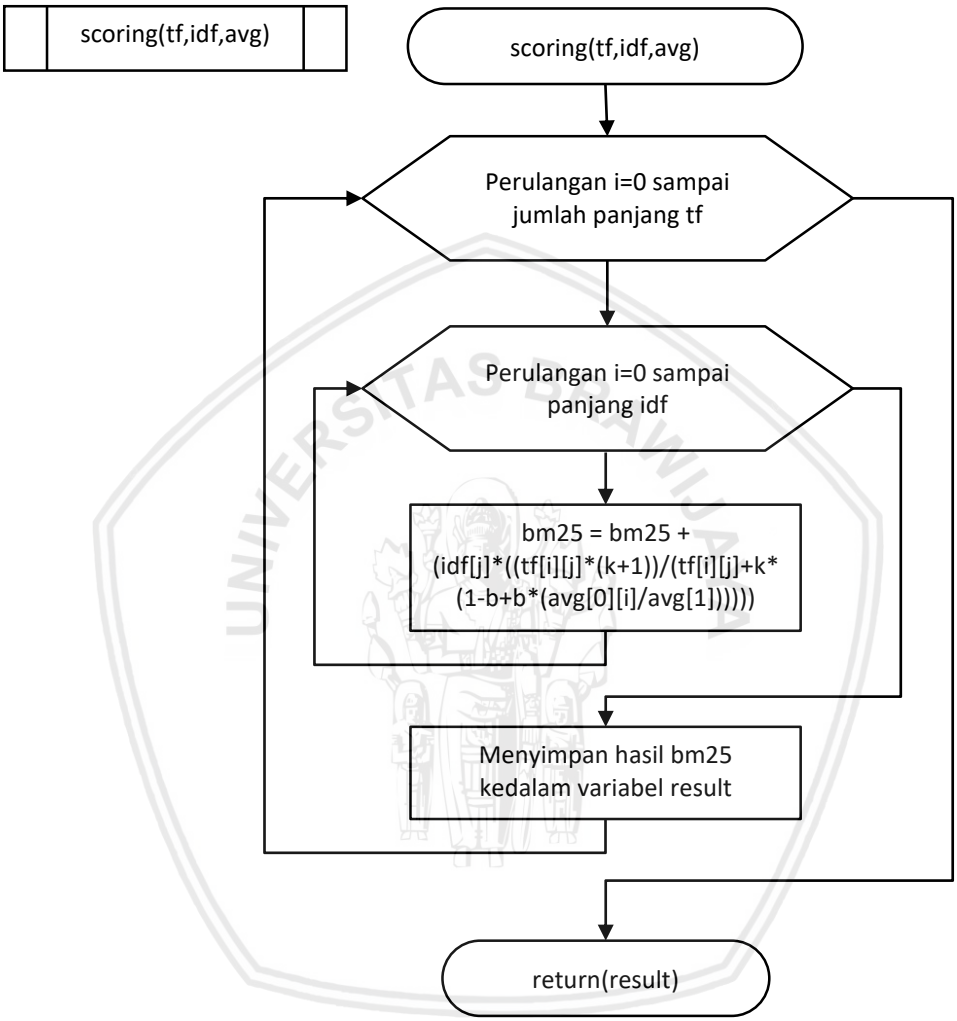


Gambar 4.10 Alur Perhitungan IDF

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.4.8 Perhitungan *Scoring* BM25

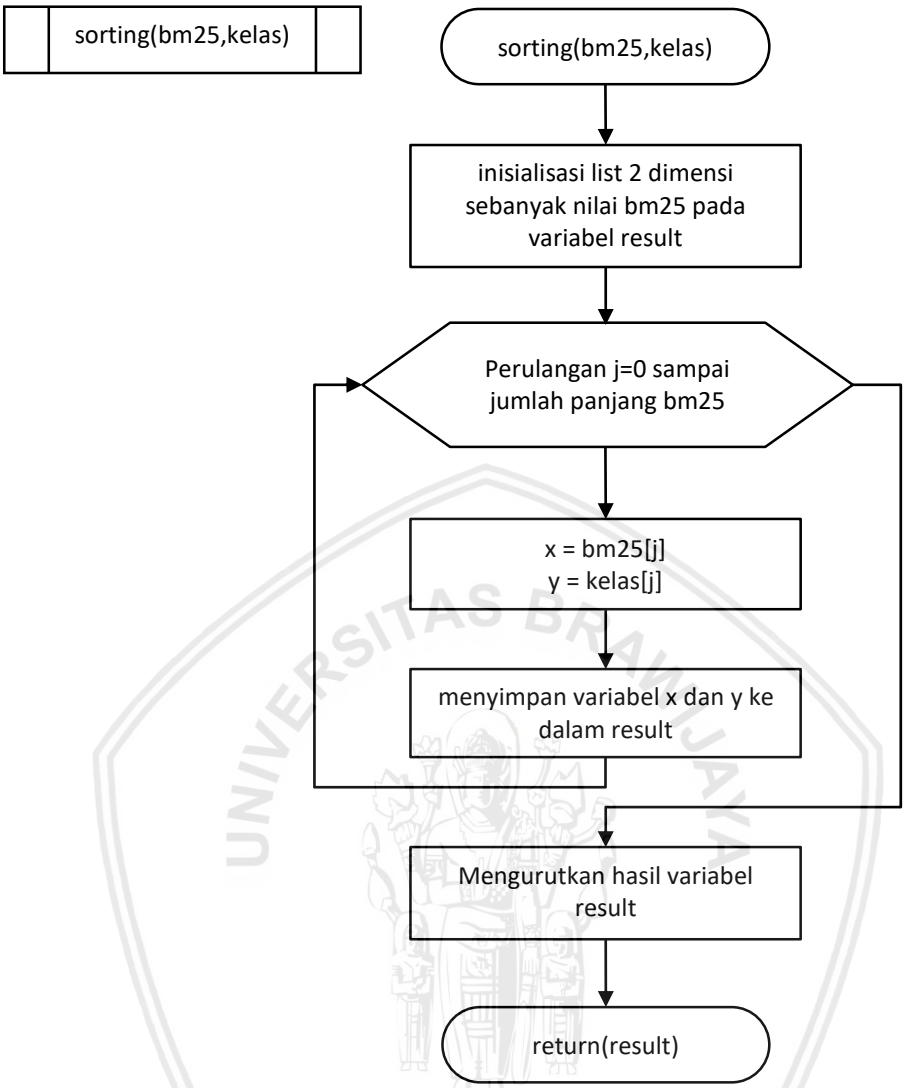
Langkah terakhir dalam pembobotan dalam algoritme BM25 yaitu menghitung *scoring* BM25. Proses perhitungan *scoring* BM25 dilakukan dengan menggunakan Persamaan 2.3. Alur perhitungan *scoring* BM25 dijelaskan pada Gambar 4.11.



Gambar 4.11 Alur Perhitungan *Scoring* BM25

4.4.9 Pemeringkatan Hasil *Scoring* BM25

Langkah awal yang dilakukan pada proses klasifikasi adalah dengan pemeringkatan hasil perhitungan dari metode BM25. Hasil perhitungan tersebut berupa bobot per-kata. Pada sub proses ini akan dilakukan pemeringkatan hasil perhitungan BM25 sebelumnya. Alur tahapan proses pemeringkatan hasil *scoring* BM25 ditunjukkan pada Gambar 4.12.

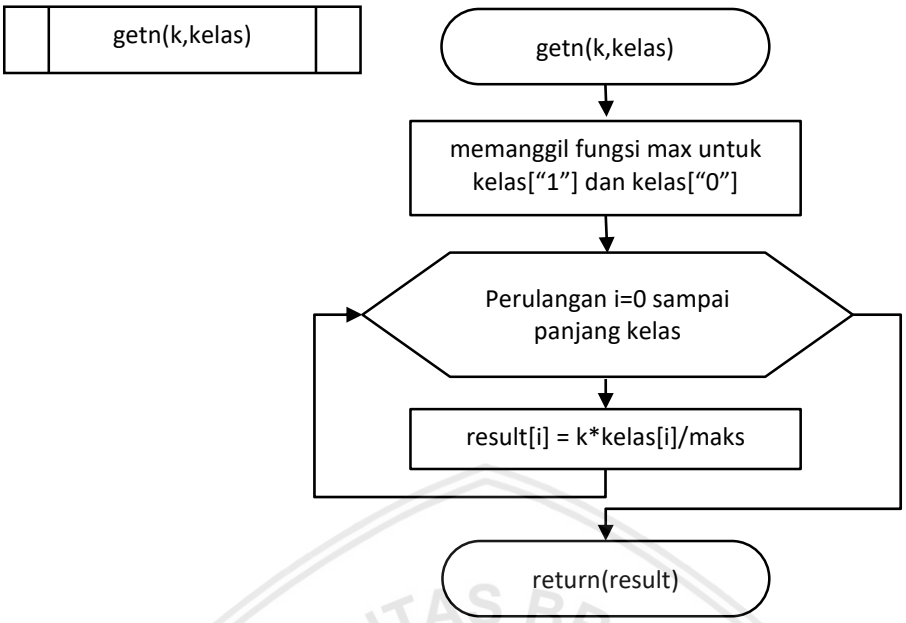


Gambar 4.12 Alur Tahapan Pemeringkatan Hasil *Scoring* BM25

4.4.10 Perhitungan Nilai *n* (*k-values* baru)

Proses selanjutnya akan dilakukan proses perhitungan nilai *n,* yang merupakan nilai *k-values* yang baru. Dalam proses ini, nilai *k* ditetapkan terlebih dahulu kemudian dilakukan perhitungan nilai *k* baru tiap kelas. Hal ini dilakukan untuk mendapatkan nilai *k-values* yang berbeda pada tiap kelas. Alur tahapan perhitungan nilai *n* dijelaskan pada Gambar 4.13.

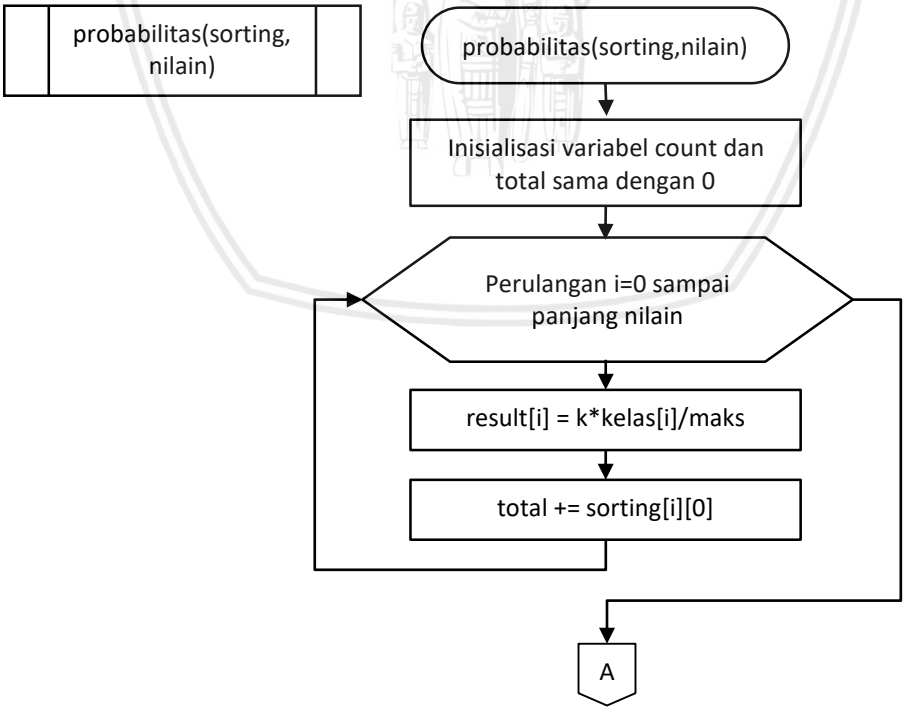
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.13 Alur Tahapan Perhitungan Nilai *n*

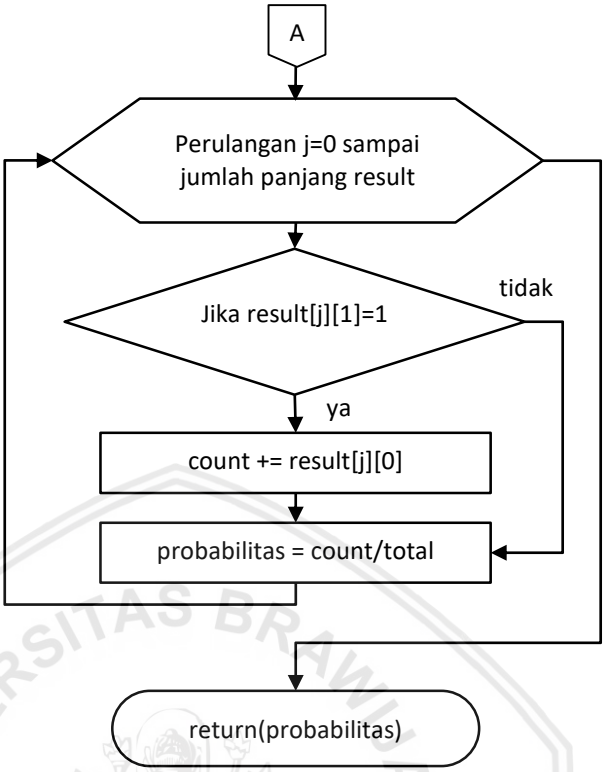
4.4.11 Perhitungan Probabilitas

Proses selanjutnya yaitu proses hitung probabilitas. Pada tahap ini dilakukan perhitungan probabilitas tiap kelas, yaitu positif dan negatif. Alur tahapan proses tersebut dijelaskan pada Gambar 4.14.



Gambar 4.14 Alur Tahapan Perhitungan Probabilitas

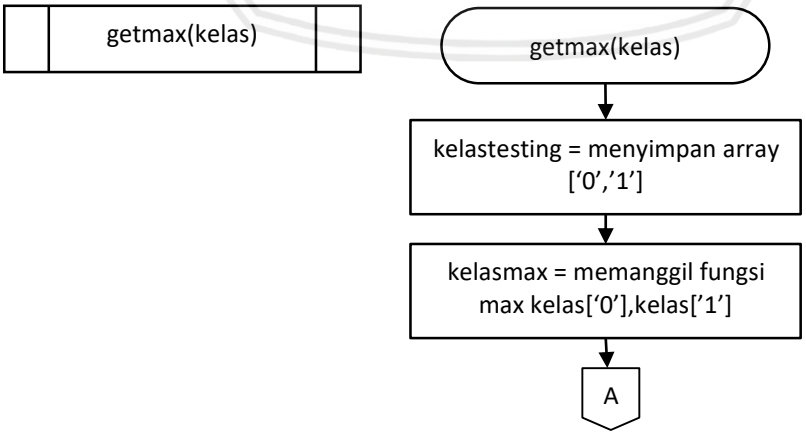
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.14 Alur Tahapan Perhitungan Probabilitas (Lanjutan)

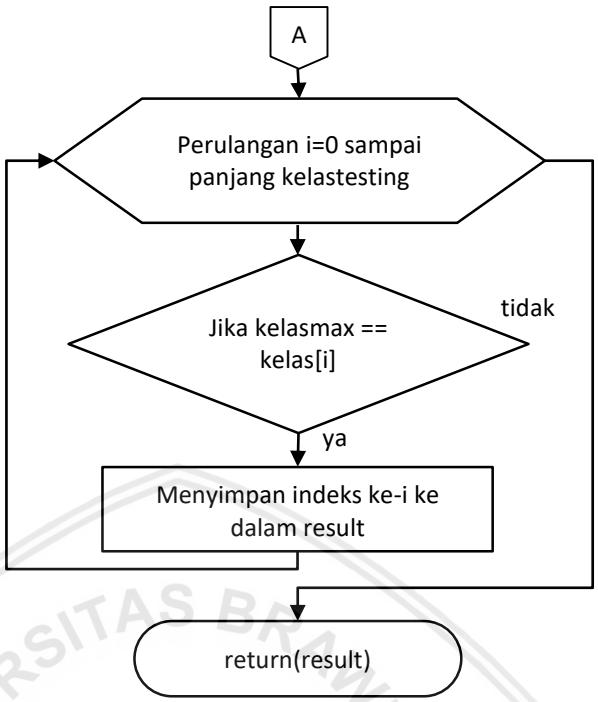
4.4.12 Penentuan Kelas

Proses setelah menemukan nilai probabilitas adalah dengan mengurutkannya. Setelah dirutkan maka akan didapatkan probabilitas terbesar yang digunakan untuk menentukan kategori suatu data uji. Apabila didapatkan nilai yang lebih besar dari nilai pada kelas yang lain, maka dapat dipastikan data uji tersebut merupakan kelas dari suatu dokumen tersebut. Alur tahapan proses tersebut dijelaskan pada Gambar 4.15.



Gambar 4.15 Alur Tahapan Penentuan Kelas

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.15 Alur Tahapan Penentuan Kelas (Lanjutan)

4.5 Manualisasi

Manualisasi merupakan tahap perhitungan manual terhadap data latih dan data uji. Proses manualisasi terdiri dari beberapa tahapan yaitu *pre-processing*, pemeringkatan kata menggunakan metode BM25, dan proses klasifikasi menggunakan metode *Improved K-Nearest Neighbor.* Terdapat 10 dokumen latih dan 4 dokumen uji yang akan dijadikan sampel dalam perhitungan manual. Data yang digunakan dalam proses manualisasi dijelaskan pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Data Latih

No.	Ulasan	Kelas
1	Saya dari thn kemarin pakai app ini, ga pernah ada masalah, walau beberapa x ganti2 hape. The problem is yours!	Positif
2	Aplikasi sgt membantu, tapi klo bisa ada pembelian token PLN buat update slnjtny	Positif
3	aplikasi yang sangat bermanfaat membantu kelancaran aktifitas perbankan	Positif
4	Aplikasi bca mobie Sangat membantu dan mudah utk diaplikasikan....mantap	Positif
5	Sangat berguna Yg pake andro 2 sim biar gak error, sms harus diset otomatis pakai sim yang terdaftar.	Positif
6	Udah berkali2 send sms buat registrasi tetep aja ga bs.... Please fix	Negatif
7	Error Saya Sudah daftar m-banking di ATM, tp pas buka m-bca untuk proses setelah masukin 16digit selalu gagal. Itu kenapa	Negatif
8	Lemot Lemot dan buang2 pulsa sms juga data juga.	Negatif

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.3 Data Latih (Lanjutan)

No.	Ulasan	Kelas
9	Gagal Maning Son! Gagal teruuus.. ketika memasukkan no atm trus submit, ga da respon. Tolong diperbaiki! Terutama utk yg dual sim.	Negatif

Tabel 4.2 Data Uji

Ulasan	Kelas
Error tolong diperbaiki aplikasi ini..	?

4.5.1 Manualisasi *Case Folding*

Pada tahap *case folding* akan dilakukan perubahan pada awalan kata seluruh dokumen menjadi huruf kecil. Manuallisasi *case folding* dokumen latih dan dokumen uji dijabarkan pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Manualisasi *Case Folding* (Data Latih)

No.	Hasil *Case Folding*
1	saya dari thn kemarin pakai app ini, ga pernah ada masalah, walau beberapa x ganti2 hape. the problem is yours!
2	aplikasi sgt membantu, tapi klo bisa ada pembelian token PLN buat update slnjtny
3	aplikasi yang sangat bermanfaat membantu kelancaran aktifitas perbankan
4	aplikasi bca mobie sangat membantu dan mudah utk diaplikasikan....mantap
5	sangat berguna yg pake andro 2 sim biar gak error, sms harus diset otomatis pakai sim yang terdaftar.
6	udah berkali2 send sms buat registrasi tetep aja ga bs.... lease fix
7	error saya sudah daftar m-banking di ATM, tp pas buka m-bca untuk proses setelah masukin 16digit selalu gagal. itu kenapa
8	lemot lemot dan buang2 pulsa sms juga data juga.
9	gagal maning son! gagal teruuus.. ketika memasukkan no atm trus submit, ga da respom. tolong diperbaiki! terutama utk yg dual sim.

Tabel 4.4 Manualisasi *Case Folding* (Data Uji)

Hasil *Case Folding*
error tolong diperbaiki aplikasi ini..

4.5.2 Manualisasi *Tokenizing*

Pada tahap manualisasi *tokenizing,* akan dilakukan proses pemisahan kumpulan kalimat menjadi kata-kata. Proses *tokenizing* tidak mempertimbangkan adanya duplikasi data, sehingga hanya bila terdapat data yang sama akan terhitung satu tidak lebih. Manualisasi *tokenizing* pada data latih dan data uji dapat dilihat pada Tabel 4.5 dan Tabel 4.6.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.5 Manualisasi *Tokenizing* (Data Latih)

No.	Hasil *Case Folding*
1	‘saya’, ‘dari’, ‘thn’, ‘kemarin’, ‘pakai’, ‘app’, ‘ini’, ‘ga’, ‘pernah’, ‘ada’, ‘masalah’, ‘walau’, ‘beberapa’, ‘x’, ‘ganti’, ‘hape’, ‘the’, ‘problem’, ‘is’, ‘yours!’
2	‘aplikasi’, ‘sgt’, ‘membantu’, ‘tapi’, ‘klo’, ‘bisa’, ‘ada’, ‘pembelian’, ‘token’, ‘PLN’, ‘buat’, ‘update’, ‘slnjtny’
3	‘aplikasi’, ‘yang’, ‘sangat’, ‘bermanfaat’, ‘membantu’, ‘kelancaran’, ‘aktifitas’, ‘perbankan’
4	‘aplikasi’, ‘bca’, ‘mobie’, ‘sangat’, ‘membantu’, ‘dan’, ‘mudah’, ‘utk’, ‘diaplikasikan’, ‘mantap’
5	‘sangat’, ‘berguna’, ‘yg’, ‘pake’, ‘andro’, ‘sim’, ‘biar’, ‘gak’, ‘error’, ‘sms’, ‘harus’, ‘diset’, ‘otomatis’, ‘pakai’, ‘sim’, ‘yang’, ‘terdaftar’,
6	‘udah’, ‘berkali’, ‘send’, ‘sms’, ‘buat’, ‘registrasi’, ‘tetep’, ‘aja’, ‘ga’, ‘bs’, ‘please’, ‘fix’
7	‘error’, ‘saya’, ‘sudah’, ‘daftar’, ‘m’, ‘banking’, ‘di’, ‘ATM’, ‘tp’, ‘pas’, ‘buka’, ‘m’, ‘bca’, ‘untuk’, ‘proses’, ‘setelah’, ‘masukin’, ‘digit’, ‘selalu’, ‘gagal’, ‘itu’, ‘kenapa’
8	‘lemot’, ‘lemot’, ‘dan’, ‘buang’, ‘pulsa’, ‘sms’, ‘juga’, ‘data’, ‘juga’
9	‘gagal’, ‘maning’, ‘son’, ‘gagal’, ‘teruuus’, ‘ketika’, ‘memasukkan’, ‘no’, ‘atm’, ‘trus’, ‘submit’, ‘ga’, ‘da’, ‘respom’, ‘tolong’, ‘diperbaiki’, ‘terutama’, ‘utk’, ‘yg’, ‘dual’, ‘sim’

Tabel 4.6 Manualisasi *Tokenizing* (Data Uji)

Hasil *Case Folding*
‘error’, ‘tolong’, ‘diperbaiki’, ‘aplikasi’, ‘ini’

4.5.3 Manualisasi *Filtering*

Pada tahapan manualisasi *filtering* akan dilakukan proses penghapusan kata yang sering muncul namun tidak memiliki makna yang berarti atau *stopword* berdasarkan *stoplist* Tala. Manualisasi *filtering* pada dokumen latih dan dokumen uji dapat dilihat pada Tabel 4.7 dan Tabel 4.8.

Tabel 4.7 Manualisasi *Filtering* (Data Latih)

No.	Hasil *Filtering*
1	‘thn’, ‘kemarin’, ‘pakai’, ‘app’, ‘ga’, ‘x’, ‘ganti’, ‘hape’, ‘the’, ‘problem’, ‘is’, ‘yours’
2	‘aplikasi’, ‘sgt’, ‘membantu’, ‘klo’, ‘pembelian’, ‘token’, ‘PLN’, ‘update’, ‘slnjtny’
3	‘aplikasi’, ‘bermanfaat’, ‘membantu’, ‘kelancaran’, ‘aktifitas’, ‘perbankan’
4	‘aplikasi’, ‘bca’, ‘mobie’, ‘membantu’, ‘mudah’, ‘utk’, ‘diaplikasikan’, ‘mantap’
5	‘berguna’, ‘yg’, ‘pake’, ‘andro’, ‘sim’, ‘biar’, ‘gak’, ‘error’, ‘sms’, ‘diset’, ‘otomatis’, ‘pakai’, ‘sim’, ‘terdaftar’
6	‘udah’, ‘berkali’, ‘send’, ‘sms’, ‘buat’, ‘registrasi’, ‘tetep’, ‘aja’, ‘ga’, ‘bs’, ‘please’, ‘fix’

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.7 Manualisasi *Filtering* (Data Latih) (Lanjutan)

No.	Hasil *Filtering*
7	‘error’, ‘daftar’, ‘m’, ‘banking’, ‘ATM’, ‘tp’, ‘pas’, ‘buka’, ‘m’, ‘bca’, ‘proses’, ‘masukin’, ‘digit’, ‘gagal’
8	‘lemot’, ‘lemot’, ‘buang’, ‘pulsa’, ‘sms’, ‘data’
9	‘gagal’, ‘maning’, ‘son’, ‘gagal’, ‘teruuus’, ‘memasukkan’, ‘no’, ‘atm’, ‘trus’, ‘submit’, ‘ga’, ‘da’, ‘respom’, ‘tolong’, ‘diperbaiki’, ‘terutama’, ‘utk’, ‘yg’, ‘dual’, ‘sim’

Tabel 4.8 Manualisasi *Filtering* (Data Uji)

Hasil *Filtering*
‘error’, ‘tolong’, ‘diperbaiki’, ‘aplikasi’

4.5.4 Manualisasi *Stemming*

Pada tahapan manualisasi *stemming,* akan dilakukan proses pengubahan kata yang memiliki imbuhan menjadi bentuk kata dasar. Penelitian ini menggunakan *library stemmer* Sastrawi. Manualisasi *stemming* pada dokumen latih dan dokumen uji dapat dilihat pada Tabel 4.9 dan Tabel 4.10.

Tabel 4.9 Manualisasi *Stemming* (Data Latih)

No.	Hasil *Stemming*
1	‘thn’, ‘kemarin’, ‘pakai’, ‘app’, ‘ga’, ‘x’, ‘ganti’, ‘hape’, ‘the’, ‘problem’, ‘is’, ‘yours’
2	‘aplikasi’, ‘sgt’, ‘bantu, ‘klo’, ‘beli, ‘token’, ‘PLN’, ‘buat’, ‘update’, ‘slnjtny’
3	‘aplikasi’, ‘manfaat, ‘bantu’, ‘lancar, ‘aktifitas’, ‘bank’
4	‘aplikasi’, ‘bca’, ‘mobile’, ‘bantu’, ‘mudah’, ‘utk’, ‘aplikasi, ‘mantap’
5	‘guna’, ‘yg’, ‘pake’, ‘andro’, ‘sim’, ‘biar’, ‘gak’, ‘error’, ‘sms’, ‘set, ‘otomatis’, ‘pakai’, ‘sim’, ‘daftar’
6	‘udah’, ‘kali’, ‘send’, ‘sms’, ‘buat’, ‘registrasi’, ‘tetep’, ‘aja’, ‘ga’, ‘bs’, ‘please’, ‘fix’
7	‘error’, ‘daftar’, ‘m’, ‘banking’, ‘ATM’, ‘tp’, ‘pas’, ‘buka’, ‘m’, ‘bca’, ‘proses’, ‘masuk’, ‘digit’, ‘gagal’
8	‘lemot’, ‘lemot’, ‘buang’, ‘pulsa’, ‘sms’, ‘data’
9	‘gagal’, ‘maning’, ‘son’, ‘gagal’, ‘teruuus’, ‘masuk, ‘no’, ‘atm’, ‘trus’, ‘submit’, ‘ga’, ‘da’, ‘respon’, ‘tolong’, ‘baik, ‘utama’, ‘utk’, ‘yg’, ‘dual’, ‘sim’

Tabel 4.10 Manualisasi *Stemming* (Data Uji)

Hasil *Stemming*
‘error’, ‘tolong’, ‘baik, ‘aplikasi’

4.5.5 Manualisasi Metode BM25

Pada manualisasi dengan menggunakan metode BM25 merupakan tahapan dalam pemeringkatan kata. Pada tahap ini, terbagi menjadi beberapa tahapan. Tahapan yang pertama yaitu menghitung TF, lalu menghitung IDF, dan yang terakhir yaitu *scoring* BM25. Tahapan yang pertama yaitu menghitung frekuensi suatu kata pada data uji berdasarkan data latih. Perhitungan TF dijelaskan pada

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

tabel sebelumnya. Dimisalkan untuk dokumen latih adalah DL dan dokumen uji adalah DU. Data latih yang digunakan merupakan data latih yang terdapat pada Tabel 4.11.

Tabel 4.11 Perhitungan TF

Term	DL1	DL2	DL3	DL4	DL5	DL6	DL7	DL8	DL9	DU1
error	0	0	0	0	1	0	1	0	0	1
tolong	0	0	0	0	0	0	0	0	1	1
baik	0	0	0	0	0	0	0	0	1	1
aplikasi	0	1	1	2	0	0	0	0	1	1

Langkah selanjutnya setelah dilakukan perhitungan TF yaitu menghitung DF dan IDF. DF merupakan frekuensi kemunculan kata pada seluruh dokumen. Perhitungan IDF dihitung menggunakan Persamaan 2.2 yang merupakan rumus dari metode BM25. Dalam proses perhitungannya, perlu diketahui nilai dari panjang dokumen, dan rata-rata panjang dokumen. Kedua nilai tersebut dijelaskan pada Tabel 4.12.

Tabel 4.12 Panjang Dokumen Dan Rata-Rata Panjang Dokumen

Ket	DL1	DL2	DL3	DL4	DL5	DL6	DL7	DL8	DL9
pjg dok	20	13	8	10	18	12	22	9	21
rata" dok	14,77778								

Berikut merupakan contoh perhitungan nilai IDF dari data uji. Perhitungan data uji berlaku untuk seluruh term pada data uji. Jumlah data latih sebanyak 9.

IDF(“error”) = $\log \frac{9-2+0,5}{2+0,5}$ = 0,477121255

IDF(“tolong”) = $\log \frac{9-1+0,5}{1+0,5}$ = 0,753327667

IDF(“baik”) = $\log \frac{9-1+0,5}{1+0,5}$ = 0,753327667

IDF(“aplikasi”) = $\log \frac{9-3+0,5}{3+0,5}$ = 0,268845312

Hasil perhitungan manualisasi dari DF dan IDF dijabarkan pada Tabel 4.13.

Tabel 4.13 Nilai DF dan IDF

Term	DF	IDF
error	2	0,477121255
tolong	1	0,753327667
baik	1	0,753327667
aplikasi	3	0,268845312

Langkah selanjutnya yaitu menghitung *scoring* BM25 menggunakan Persamaan 2.1. Pada tahap ini, menggunakan nilai *k1* sama dengan 2, nilai *b* sama

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

dengan 0,75 sesuai dengan nilai yang telah ditetapkan, dan nilai dari rata-rata panjang dokumen sebesar 14,333. Salah satu contoh perhitungan *scoring* BM25 adalah pada dokumen data latih pertama sebagai berikut.

BM25("error") = 0,477121255. $\frac{0.(2+1)}{0+2.(1-0,75+0,75.\frac{20}{14,33})}$ = 0

BM25("tolong") = 0,753327667. $\frac{0.(2+1)}{0+2.(1-0,75+0,75.\frac{20}{14,33})}$ = 0

BM25("baik") = 0,753327667. $\frac{0.(2+1)}{0+2.(1-0,75+0,75.\frac{20}{14,33})}$ = 0

BM25("aplikasi") = 0,268845312. $\frac{0.(2+1)}{0+2.(1-0,75+0,75.\frac{20}{14,33})}$ = 0

Hasil dari perhitungan seluruh data uji dijumlahkan, sehingga mendapatkan hasil sama dengan 0. Hasil dari perhitungan *scoring* BM25 dari data uji akan dijabarkan pada Tabel 4.14.

Tabel 4.14 Hasil *Scoring* BM25

Kelas	DOC	Score
Positif	DL1	0
Positif	DL2	0,286051412
Positif	DL3	0,348843186
Positif	DL4	0,458906009
Positif	DL5	0,430217809
Negatif	DL6	0
Negatif	DL7	0,383426749
Negatif	DL8	0
Negatif	DL9	1,244628319

Setelah diketahui hasil *scoring* BM25 maka proses selanjutnya yaitu dengan melakukan proses klasifikasi menggunakan metode *Improved K-Nearest Neighbor*.

4.5.6 Manualisasi *Improved K-Nearest Neighbor*

Tahapan manualisasi *Improved K-Nearest Neighbor* merupakan tahapan yang digunakan dalam proses klasifikasi. Proses ini mengklasifikasikan dokumen menjadi dua kelas, yaitu positif dan negatif. Tahapan awal dalam manualisasi ini adalah dengan mengurutkan hasil *scoring* BM25. Hasil pemeringkatan tersebut dijabarkan pada Tabel 4.15.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.15 Hasil Pemeringkatan BM25

Kelas	DOC	Score
Negatif	D9	2,292367202
Negatif	D7	1,810264125
Positif	D5	1,358543453
Positif	D3	0,619093921
Positif	D1	0,405476849
Positif	D2	0
Positif	D4	0
Negatif	D6	0
Negatif	D8	0

Proses ini menggunakan hasil dari pemeringkatan BM25 dalam menghitung kemiripan dokumen data latih dan data uji. Perbandingan kemiripan ini dihitung menggunakan Persamaan 2.3. Pada Tabel 4.16 akan dijabarkan jumlah data latih yang digunakan dalam proses ini.

Tabel 4.16 Jumlah Data Latih

Data Latih		
Positif	Negatif	Jumlah
5	4	9

Berdasarkan pada Tabel 4.16, maka didapatkan hasil *n* (*k values* baru) dengan nilai *k* awal sama dengan 3, dan didapatkan perhitungan sebagai berikut.

Kelas positif.

$n = \frac{3 \times 5}{5} = 3$

Kelas negatif.

$n = \frac{3 \times 4}{5} = 2{,}4 = 2$

Hasil nilai *n* untuk masing-masing kelas bernilai 3. Tabel 4.17 menunjukkan hasil perhitungan dari nilai *k-values* awal dan nilai *n*.

Tabel 4.17 Hasil Perhitungan Nilai *n* (*k-values* baru)

k-values	*n* (*k-values* baru)	
	Positif	Negatif
3	3	2

Setelah didapatkan hasil dari nilai *n,* maka dilakukan perhitungan peluang dokumen uji yang termasuk dokumen latih. Berikut merupakan contoh perhitungan manualisasi proses tersebut.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

$$P(x,c_{positif}) = \frac{(2,292367202*0)+(1,810264125*0)+(1,358543453*1)}{2,292367202 + 1,810264125 + 1,358543453} = \frac{1,358543453}{5,46117478} = 0,248763958$$

$$P(x,c_{negatif}) = \frac{(2,292367202*1)+(1,810264125*1)}{2,292367202 + 1,810264125} = \frac{4,102631}{4,102631} = 1$$

Dari hasil perhitungan tersebut dapat disimpulkan bahwa data uji yang dimasukkan termasuk ke dalam kelas negatif. Hasil perhitungan tersebut menunjukkan bahwa nilai probabilitas kelas negatif yaitu 0,248 lebih besar daripada kelas positif yaitu sebesar 1.

4.6 Perancangan Pengujian Sistem

Pada sub bab ini akan dilakukan pengujian yang bertujuan untuk mengetahui hasil dari sistem yang telah dibuat. Pengujian yang dilakukan yaitu dengan menggunakan variasi nilai *k* yang bertujuan untuk mengetahui hasil akurasi terbaik. Pengujian pengaruh nilai *k* juga dilakukan dengan menggunakan pengujian *k-fold,* yaitu dengan menggunakan dokumen latih dan dokumen uji yang berbeda tiap pengujian. Perhitungan yang digunakan yaitu meliputi *precision, recall, f-measure,* dan *accuracy* berdasarkan keluaran yang dihasilkan oleh sistem.

4.6.1 Perancangan *Confusion Matrix*

Perancangan ini dilakukan untuk mengetahui nilai *true positive* (TP), *false negative* (FN), *false positive* (FP), dan *true negative* (TN) yang nantinya akan digunakan dalam perhitungan evaluasi *precision, recall, f-measure,* dan *accuracy.* Perancangan evaluasi *confusion matrix* untuk kelas positif dan kelas negatig ditunjukkan pada Tabel 4.18 dan Tabel 4.19.

Tabel 4.18 Perancangan *Confusion Matrix* Positif

		Nilai Prediksi POSITIF	
		POSITIF	**Bukan POSITIF**
Nilai Sebenarnya POSITIF	**POSITIF**	.. (TP)	.. (FN)
	Bukan POSITIF	.. (FP)	.. (TN)

Tabel 4.19 Perancangan *Confusion Matrix* Negatif

		Nilai Prediksi NEGATIF	
		NEGATIF	**Bukan NEGATIF**
Nilai Sebenarnya NEGATIF	**NEGATIF**	.. (TP)	.. (FN)
	Bukan NEGATIF	.. (FP)	.. (TN)

4.6.2 Perancangan Pengujian Pengaruh Nilai *k-values*

Proses pengujian ini dilakukan untuk mengetahui nilai ketetanggaan paling baik atau optimal melalui nilai *k* yang digunakan dalam proses klasifikasi. Hasil ini nantinya akan memengaruhi hasil akurasi dari sistem. Nilai *k* yang akan digunakan pada pengujian ini berbeda-beda. Pada pengujian ini, dilakukan dengan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

menggunakan *5-fold cross validation.* Perancangan pengujian pengaruh nilai *k-values* ditunjukkan pada Tabel 4.20.

Tabel 4.20 Perancangan Pengujian Pengaruh Nilai *k-values*

k-values	***n* (*k-values* baru)**		***Precision***	***Recall***	***F-Measure***	***Accuracy***
	Positif	**Negatif**				
2						
3						
5						
10						
....						
150						

4.7 Penarikan Kesimpulan

Proses penarikan kesimpulan dilakukan untuk menyimpulkan hasil yang diperoleh setelah seluruh tahapan mulai dari perancangan, implementasi, dan pengujian sistem selesai dilakukan. Setelah mendapatkan suatu kesimpulan, maka dapat dibuat suatu saran terhadap sistem tersebut. Saran dapat berupa perbaikan atau masukan agar sistem yang telah dibuat sebelumnya dapat dikembangkan ataupun diperbaiki.

4.8 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penerapan perancangan sistem ke dalam sistem yang ingin dibuat. Implementasi sistem ini juga dibagi menjadi beberapa proses yaitu *pre-processing,* pembobotan BM25, dan klasifikasi menggunakan *Improved KNN.* Masing-masing proses tersebut terdiri dari beberapa sub proses.

4.8.1 Implementasi *Pre-processing*

Tahap implementasi *pre-processing* terbagi menjadi beberapa sub proses yaitu *case folding, tokenizing, filtering,* dan *stemming.*

4.8.1.1 Implementasi *Case Folding*

Proses *case folding* merupakan proses mengubah huruf awal pada suatu kata menjadi huruf kecil. Kode program untuk proses *case folding* dijabarkan pada Kode Program 4.1.

```
Algoritme: Case Folding
1   def casefolding(doc):
2       result = []
3       for i in doc:
4           temp = i.lower()
5           result.append(temp)
6       return result
```

Kode Program 4.1 *Case Folding*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Penjelasan untuk Kode Program 4.1 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method casefolding*.

Baris 2 merupakan inisialisasi variabel *result berupa* array 1 dimensi.

Baris 3-6 merupakan proses mengubah seluruh kata pada *doc* menjadi *lowercase.*

4.8.1.2 Implementasi *Tokenizing*

Proses *tokenizing* merupakan proses memisahkan kata pada tiap kalimat menjadi token. Pada proses ini juga dilakukan proses menghilangkan karakter yang tidak diperlukan. Kode program untuk proses *tokenizing* dijabarkan pada Kode Program 4.2.

```
Algoritme: Tokenizing
1   def tokenizing(casefolding):
2       result = []
3       for i in range(len(casefolding)):
4           temp = re.sub('[^a-z0-9]',' ',casefolding[i])
5           split = temp.split(' ')
6           a = ''
7           tokens = []
8           for i in split:
9               if i != a:
10                  tokens.append(i)
11          result.append(tokens)
12      return result
```

Kode Program 4.2 *Tokenizing*

Penjelasan untuk Kode Program 4.2 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method tokenizing*.

Baris 2 merupakan inisialisasi variabel *result* berupa *array* 1 dimensi.

Baris 3-7 merupakan proses memisah kalimat menjadi token.

Baris 8-12 merupakan proses menghilangkan himpunan kosong.

4.8.1.3 Implementasi *Filtering*

Proses *filtering* merupakan proses yang digunakan untuk menghilangkan kata yang tidak penting serta menghapus karakter dan angka. Proses ini menggunakan *stopword list* Tala sebagai acuan untuk menghilangkan kata yang tidak diperlukan untuk dilakukan proses selanjutnya. Kode program untuk proses *filtering* dijabarkan pada Kode Program 4.3.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
Algoritme: Filtering
1   def filtering(tokenizing):
2       result = []
3       with open('stopwords.txt') as f:
4           content = f.readlines()
5       stopwords = [x.strip() for x in content]
6       for i in range(len(tokenizing)):
7           filters = [w for w in tokenizing[i] if w not in
    stopwords]
8           result.append(filters)
9       return result
```

Kode Program 4.3 *Filtering*

Penjelasan untuk Kode Program 4.3 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method filtering*.

Baris 2 merupakan inisialisasi variabel *result* berupa *array* 1 dimensi.

Baris 3-5 merupakan membaca dokumen *stowords* berdasarkan baris.

Baris 6-9 merupakan proses menghilangkan menghilangkan kata yang termasuk *stowpords*.

4.8.1.4 Implementasi *Stemming*

Proses *stemming* merupakan proses mengubah suatu bentuk kata menjadi kata dasar. Pada proses ini digunakan *library* Sastrawi yang digunakan sebagai *stemmer* dalam proses *Stemming.* Kode program untuk proses *stemming* dijabarkan pada Kode Program 4.4.

```
Algoritme: Stemming
1   def stemming(filtering):
2       result = []
3       for i in range(len(filtering)):
4           temp = []
5           for j in filtering[i]:
6               factory = StemmerFactory()
7               stemmer = factory.create_stemmer()
8               a = stemmer.stem(j)
9               temp.append(a)
10          result.append(temp)
11      return result
```

Kode Program 4.4 *Stemming*

Penjelasan untuk Kode Program 4.4 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method stemming*.

Baris 2 merupakan inisialisasi variabel *result* berupa *array* 1 dimensi.

Baris 3-11 merupakan proses memanggil fungsi *stemmer* dan menyimpan hasil *stemming* ke dalam variabel *result.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.8.2 Implementasi Metode BM25

Tahap implementasi dalam proses pembobotan menggunakan metode BM25 terbagi menjadi beberapa sub proses. Proses tersebut yaitu perhitungan TF, DF, IDF, dan *scoring* BM25.

4.8.2.1 Implementasi Perhitungan TF

Proses perhitungan TF merupakan proses menghitung jumlah atau frekuensi kemunculan term pada satu dokumen. Implementasi perhitungan TF dihitung dengan menggunakan term unik dari data latih yang telah dilakukan proses *pre-processing.* Implementasi perhitungan TF dijelaskan pada Kode Program 4.5.

```
Algoritme: Perhitungan TF
1    def gettf(doc,termunik):
2        result = []
3        for i in range(len(doc)):
4            value = 0
5            termunik = dict.fromkeys(termunik,value)
6            for j in doc[i]:
7                if j in termunik:
8                    termunik[j] += 1
9            result.append(termunik)
10       return result
```

Kode Program 4.5 Perhitungan TF

Penjelasan untuk Kode Program 4.5 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method gettf*.

Baris 2 merupakan inisialisasi *variabel* result berupa *array* 1 dimensi.

Baris 3-5 merupakan proses memanggil *method* termunik.

Baris 6-8 merupakan proses menghitung apabila kata pada dokumen juga terdapat pada termunik maka nilainya bertambah 1.

Baris 9-10 merupakan menyimpan hasil pada *result*.

4.8.2.2 Implementasi Perhitungan DF

Proses perhitungan DF merupakan proses menghitung jumlah total kemunculan term pada suatu dokumen. Implementasi perhitungan DF dijelaskan pada Kode Program 4.6.

```
Algoritme: Perhitungan DF
1    def getdf(tf, doctesting):
2        df = []
3        result = {}
4        for i in range(len(tf)):
5            freq = {}
6            for j in doctesting:
7                freq[j] = 0
8            for k in freq:
9                if tf[i][k] > 0:
10                   freq[k] = 1
11               else:
```


repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
12                  freq[k] = 0
13          df.append(freq)
14      for i in df[0]:
15          count=0
16          for j in range(len(df)):
17              count = count + df[j][i]
18          result[i] = count
19      return result
```

Kode Program 4.6 Perhitungan DF

Penjelasan untuk Kode Program 4.6 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method getdf*.

Baris 2-3 merupakan inisialisasi variabel *df* berupa *array* 1 dimensi dan *result* berupa *list.*

Baris 4-12 merupakan proses jika *tf* lebih dari 0 maka frekuensi bertambah 1, selain itu bernilai 0.

Baris 13 merupakan menyimpan hasil frekuensi ke dalam variabel *df*.

Baris 14-19 merupakan proses menghitung nilai *df* dengan menghitung *count* dan memasukkan hasilnya ke dalam *result*.

4.8.2.3 Implementasi Perhitungan IDF

Proses perhitungan IDF dilakukan dengan menggunakan Persamaan 2.3 yang merupakan persamaan dari metode BM25. Implementasi perhitungan IDF dijelaskan pada Kode Program 4.7.

```
Algoritme: Perhitungan IDF
1   def getidf(df, doctraining):
2       result = {}
3       len_doc = len(doctraining)
4       for i in df:
5           docFreq = float(df[i])
6           result[i] = math.log10((len_doc - docFreq + 0.5) /
7   (docFreq + 0.5))
8       return result
```

Kode Program 4.7 Perhitungan IDF

Penjelasan untuk Kode Program 4.7 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method getidf*.

Baris 2-3 merupakan inisialisasi variabel *result* berupa *list*, dan *len_doc* yang merupakan panjang dari *doctraining*.

Baris 4-6 merupakan proses menghitung IDF menggunakan Persamaan 2.3 dan menyimpan ke dalam *result*.

4.8.2.4 Implementasi *Scoring* BM25

Proses perhitungan *scoring* BM25 dilakukan menggunakan nilai dari TF, IDF, dan rata-rata panjang dokumen *training*. Implementasi perhitungan *scoring* BM25 dijelaskan pada Kode Program 4.8.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
Algoritme: Scoring BM25
1    def scoring(tf,idf,avg):
2        result = []
3        k = 2
4        b = 0.75
5        for i in range(len(tf)):
6            bm25 =0
7            for j in idf:
8                bm25 += (idf[j] * ((tf[i][j] * (k+1)) / (tf[i]
     [j] + k *(1 - b+b * (avg[0][i]/avg[1])))))
9            result.append(bm25)
10       return result
```

Kode Program 4.8 *Scoring* BM25

Penjelasan untuk Kode Program 4.8 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method scoring*.

Baris 2-4 merupakan inisialisasi variabel *result* berupa *array* 1 dimensi, *k* sama dengan 2, dan *b* sama denga 0,75.

Baris 5-10 merupakan proses perhitungan BM25 menggunakan Persamaan 2.1 dan menyimpan hasil variabel *bm25* ke dalam *result.*

4.8.3 Implementasi *Improved K-Nearest Neighbor*

Tahap impementasi algoritme *Improved K-Nearest Neighbor* terbagi menjadi beberapa proses. Proses tersebut yaitu melakukan pemeringkatan hasil *scoring* BM25, perhitungan nilai *n* (*k-values* baru), perhitungan probabilitas tiap kelas, dan penentuan kelas dokumen uji.

4.8.3.1 Implementasi Pemeringkatan Hasil *Scoring* BM25

Proses pemeringkatan hasil dari *scoring* BM25 dilakukan dengan menggunakan hasil dari perhitungan *scoring* BM25. Implementasi pemeringkatan hasil *scoring* BM25 dijelaskan pada Kode Program 4.9.

```
Algoritme: Pemeringkatan Hasil Scoring BM25
1    def sorting(bm25, kelas):
2        result = [[] for i in range(len(bm25))]
3        for j in range(len(bm25)):
4            x = bm25[j]
5            y = kelas[j]
6            result[j].append(x)
7            result[j].append(y)
8        result = sorted(result, reverse=True)
9        return result
```

Kode Program 4.9 Pemeringkatan Hasil *Scoring* BM25

Penjelasan untuk Kode Program 4.9 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method sorting*.

Baris 2 merupakan inisialisasi variabel *result* berupa *array* 2 dimensi.

Baris 3-5 merupakan proses menyimpan nilai *bm25* dan kelas ke dalam variabel *x* dan *y*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baris 6-7 merupakan proses menyimpan variabel *x* dan *y* ke dalam *result*.

Baris 8-9 merupakan proses memanggil *method sorted* untuk mengurutkan nilai *bm25*.

4.8.3.2 Implementasi Perhitungan Nilai *n*

Proses perhitungan nilai *n* (*k-values* baru) dilakukan untuk mencari nilai dari *k* tiap kelas untuk digunakan pada proses selanjutnya. Implementasi perhitungan nilai *n* dijelaskan pada Kode Program 4.10.

```
Algiritme: Perhitungan Nilai n
1   def getn(k, kelas):
2       maks = max(kelas["1"],kelas["0"])
3       result = {}
4       for i in kelas:
5           result[i] = round(k*kelas[i]/maks)
6       return result
```

Kode Program 4.10 Perhitungan Nilai *n*

Penjelasan untuk Kode Program 4.10 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method getn*.

Baris 2 merupakan inisialisasi variabel *maks* dengan memanggil *method max*.

Baris 3 merupakan inisialisasi variabel *result* berupa *list.*

Baris 4-6 merupakan proses perhitungan nilai *n* dan menyimpan ke dalam *result*.

4.8.3.3 Implementasi Perhitungan Probabilitas

Proses perhitungan probabilitas dilakukan untuk mencari peluang suatu dokumen termasuk ke dalam kelas positif atau negatif. Implementasi perhitungan probabilitas terbagi menjadi dua yaitu probabilitas positif dan negatif, yang dijelaskan pada Kode Program 4.11.

```
Algoritme: Perhitungan Probabilitas
1   def probabilitas(sorting, nilain):
2       result = []
3       count = 0
4       total = 0
5       for i in range(nilain):
6           result.append(sorting[i])
7           total += sorting[i][0]
8       for j in range(len(result)):
9           if result[j][1] == '1':
10              count += result[j][0]
11      prob_positif = count/total
12      return prob_positif
```

Kode Program 4.11 Perhitungan Probabilitas

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Penjelasan untuk Kode Program 4.11 adalah sebagai berikut.

Baris 1 merupakan deklarasi method probabilitas.

Baris 2-4 merupakan inisialisasi variabel *df* berupa *array* 1 dimensi, *count*, dan total sama dengan 0.

Baris 5-7 merupakan proses penjumlahan nilai *bm25* sebanyak nilai *n.*

Baris 8-10 merupakan proses perhitungan nilai jika bernilai 1 maka akan dijumlahkan.

Baris 11-12 merupakan proses menyimpan hasil *count* dibagi total ke dalam *prob_positif*.

4.8.3.4 Implementasi Penentuan Kelas

Dari hasil perhitungan probabilitas, maka diketahui nilai probabilitas mana yang merupakan nilai terbesar. Maka dari itu, proses penentuan kelas dihitung berdasarkan nilai probabilitas terbesar tersebut. Implementasi penentuan kelas dijelaskan pada Kode Program 4.12.

```
Algoritme: Penentuan Kelas
1   def getmax(kelas):
2       kelastesting=['0','1']
3       result = []
4       kelasmax = max(kelas['0'],kelas['1'])
5       for i in kelastesting:
6           if kelasmax == kelas[i]:
7               result.append(i)
8       return result
```

Kode Program 4.12 Penentuan Kelas

Penjelasan untuk Kode Program 4.12 adalah sebagai berikut.

Baris 1 merupakan deklarasi *method getmax*.

Baris 2-4 merupakan inisialisasi variabel *kelastesting*, *result* berupa *array* 1 dimensi, dan *kelasmax* dengan memanggil fungsi *max* pada kelas.

Baris 5-8 merupakan proses mengurutkan kelastesting dengan nilai yang terbesar dan menyimpan ke dalam *result.*

4.9 Tampilan Sistem

Hasil sistem yang telah dibuat pada penelitian analisis sentimen ini ditunjukkan pada Gambar 4.16.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
Console 1/A
K =  3

JUMLAH KELAS
{'1': 160, '0': 240}

 Nilai K-Values Awal :  3

NILAI N (K-VALUES BARU)
{'1': 2, '0': 3}
Ulasan            Prediksi        Aktual
Ulasan  1  =      Positif         Positif
Ulasan  2  =      Positif         Positif
Ulasan  3  =      Positif         Positif
Ulasan  4  =      Positif         Positif
Ulasan  5  =      Positif         Positif
Ulasan  6  =      Positif         Positif
Ulasan  7  =      Positif         Positif
Ulasan  8  =      Positif         Positif
Ulasan  9  =      Positif         Positif
Ulasan  10  =     Positif         Positif
Ulasan  11  =     Positif         Positif
Ulasan  12  =     Positif         Positif
```

Gambar 4.16 Hasil Tampilan Sistem

Pada Gambar 4.16 menjelaskan bahwa keluaran yang dihasilkan sistem berupa kelas dokumen uji yang dimasukkan yaitu sebanyak 100 dokumen. Sistem akan menampilkan kelas dokumen uji satu per satu sebanyak 100 kali. Kelas 1 menunjukkan kelas positif dan kelas 0 menunjukkan kelas negatif.

BAB 5 HASIL DAN PEMBAHASAN

Pada bab hasil dan pembahasan, menjelaskan tentang hasil yang didapatkan dari pembuatan sistem serta pembahasan tentang sistem tersebut. Pembahasan yang dilakukan terdiri dari hasil pengujian berupa *precision, recall, f-measure,* dan *accuracy.*

5.1 Hasil Sistem

Sistem yang telah dibuat mendapatkan hasil berupa keluaran yang menunjukkan kelas dari 100 dokumen uji. Hasil yang didapatkan oleh sistem merupakan hasil evaluasi berupa pengujian *k-fold* yang berjumlah 5 atau bisa disebut dengan 5-*fold.* Hasil dari sistem merupakan hasil dari pengujian 100 data uji yang terdiri dari 60 data positif dan 0 data negatif, yang dijelaskan pada Tabel 5.1, hasil sistem selengkapnya dijelaskan pada Lampiran B.

Tabel 5.1 Hasil Klasifikasi Sistem

No	Ulasan	Kelas Aktual	Kelas Prediksi
1	Lebih praktis bertransaksi dengan bank bca ini mudah	Positif	Positif
2	Mantap banget woo oii bisa kirim uang dan transfer secepat kilat	Positif	Positif
3	Mempermudah Tingkatkan lagi pelayanan dan kualitas bagus	Positif	Positif
4	Keren All transaksi ada di tanganmu	Positif	Positif
5	Bagus sekali, sangat membantu dan aman.	Positif	Positif
6	aplikasi yang sangat membantu di dalam transaksi dimana pun kami berada	Positif	Positif
7	kemudahan bertransaksi ada pada gemgaman praktis efisiensi mudah	Positif	Positif
8	Suka banget Mudah untuk bertransaksi apapun	Positif	Positif
9	So simple Nyantai drmah, mw transfer tengah malam, beres	Positif	Positif
10	Lebih efisien Kita nggak hrs keluar rumah untuk melakukan transaksi	Positif	Positif
...			
96	Not for use Nomor hp saya sudah terdaftar utk mobile banking, tetapi untuk di gunakan di aplikasi ini selalu di minta untuk daftar di ATM terdekat! Ini aplikasi tdk jelas dan Customer service BCA pun tidak bisa memberikan solusi terhadap masalah masalah yg ada! Saran dari bca hanya uninstall dan install kembali! Satu jawaban dr sekian CS yg saya hubungi!kasihan CS nya pengetahuan produk nya hanya seperti itu, sangat tidak memuaskan!	Negatif	Negatif
97	Error 107 Bisa kasih bintang 1/4 gak. Aslii ancur bgt nih aplikasi 1 thn lebih gak ada update apa2.	Negatif	Negatif

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.1 Hasil Klasifikasi Sistem (Lanjutan)

No	Ulasan	Kelas Aktual	Kelas Prediksi
98	BCA kacau Bro. Sering muncul: 101 unable bla bla harus install ulang! Ngisi no kartu, dijawab "ndaftar di atm". Repot amat, pdhal sdh sering pakai. BCA koq gak pernah ngrespon semua keluhan. kasihan bagian IT nya, gak cerdas dan gak simpatik. Keluhan sama dan sering tp njawab aja gak! Apalagi memperbaiki! BCA persh besar tp kerjanya payah.	Negatif	Negatif
99	Error 107 Jumat kemaren baru buka rekening BCA + daftar mobile banking. Setelah download, harus masukin nomor kartu, udah masukin semua gag bisa bisa, akhirnya saya putusin untuk coba di rumah. Sampe dirumah ga bisa juga, besokannya saya coba lagi, ga bisa lagi, sampe hari ini gag bisa juga. Gimana nih? masak kalah sama bank-bank lain? kalo kayak gini mah males banget dong.	Negatif	Negatif
100	Gagal install Selalu muncul error 132(2).	Negatif	Negatif
True Positive (TP)			98

5.2 Hasil Pengujian

Hasil pengujian yang dilakukan yaitu untuk mendapatkan nilai dari *precision, recall, f-measure,* dan *accuracy.* Maka dari itu dilakukan pengujian menggunakan *k-fold* sebanyak 5 kali. Data uji yang digunakan tiap pengujian *5-fold* sebanyak 100 data, yang terdiri dari 20 persen data latih tiap kelas yaitu sebanyak 60 uji positif dan 40 data uji negatif. Setiap *k-fold* menggunakan data uji yang berbeda dari data lainnya.

5.2.1 Pengujian *5-Fold* Pada *Fold* Ke-1

Pengujian *5-fold* pada *fold* pertama dilakukan dengan menggunakan data latih sebanyak 400 dokumen, dan data uji sebanyak 100 dokumen. Hasil pengujian *5-fold* pada *fold* ke-1 dapat dilihat pada Tabel 5.2.

Tabel 5.2 Hasil Pengujian *Fold* 1

k-values	*k-values* baru		*Precision*	*Recall*	*F-Measure*	*Accuracy*
	Positif	Negatif				
2	1	2	0,888	0,863	0,871	0,880
3	2	3	0,932	0,904	0,914	0,920
5	**3**	**5**	**0,940**	**0,917**	**0,925**	**0,930**
10	7	10	0,932	0,904	0,914	0,920
20	13	20	0,932	0,904	0,914	0,920
30	20	30	0,948	0,913	0,924	0,930
40	27	40	0,935	0,888	0,902	0,910
50	33	50	0,941	0,900	0,913	0,920
100	67	100	0,911	0,838	0,854	0,870
150	100	150	0,905	0,825	0,842	0,860

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.2 menyatakan hasil *precision, recall, f-measure,* dan *accuracy* terhadap nilai *k-values* pada pengujian *fold* 1. Pada pengujian tersebut, nilai *f-measure* tertinggi diperoleh ketika nilai *k-values* sama dengan 5 dengan nilai *f-measure* sebesar 0,925, *precision* sebesar 0,940, dan *recall* sebesar 0,917, sedangkan nilai *f-measure* terendah diperoleh ketika nilai *k-values* sama dengan 150 dengan nilai *f-measure* sebesar 0,842, *precision* sebesar 0,905, dan *recall* sebesar 0,825.

5.2.2 Pengujian *5-Fold* Pada *Fold* Ke-2

Pengujian *fold* 2 dilakukan dengan menggunakan data latih sebanyak 400 dokumen, dan data uji sebanyak 100 dokumen. Hasil pengujian *fold* 2 dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil Pengujian *Fold* 2

k-values	*k-values* baru		*Precision*	*Recall*	*F-Measure*	*Accuracy*
	Positif	Negatif				
2	1	2	0,869	0,858	0,863	0,870
3	2	3	0,904	0,888	0,894	0,900
5	3	5	0,904	0,888	0,894	0,900
10	**7**	**10**	**0,918**	**0,896**	**0,904**	**0,910**
20	13	20	0,910	0,893	0,899	0,906
30	20	30	0,925	0,892	0,903	0,910
40	27	40	0,918	0,879	0,891	0,900
50	33	50	0,918	0,879	0,891	0,900
100	67	100	0,911	0,867	0,880	0,890
150	100	150	0,905	0,854	0,868	0,880

Tabel 5.3 menyatakan hasil *precision, recall, f-measure,* dan *accuracy* terhadap nilai *k* pada pengujian *fold* 2. Pada pengujian tersebut, nilai *f-measure* tertinggi diperoleh ketika nilai *k-values* sama dengan 10 dengan nilai *f-measure* sebesar 0,904, *precision* sebesar 0,918, dan *recall* sebesar 0,896, sedangkan nilai *f-measure* terendah diperoleh ketika nilai *k-values* sama dengan 2 dengan nilai *f-measure* sebesar 0,863, *precision* sebesar 0,869, dan *recall* sebesar 0,858.

5.2.3 Pengujian *5-Fold* Pada *Fold* Ke-3

Pengujian *fold* 3 dilakukan dengan menggunakan data latih sebanyak 400 dokumen, dan data uji sebanyak 100 dokumen. Hasil pengujian *fold* 3 dapat dilihat pada Tabel 5.4.

Tabel 5.4 Hasil Pengujian *Fold* 3

k-values	*k-values* baru		*Precision*	*Recall*	*F-Measure*	*Accuracy*
	Positif	Negatif				
2	1	2	0,894	0,904	0,897	0,900
3	2	3	0,915	0,921	0,917	0,920
5	3	5	0,915	0,921	0,917	0,920
10	**7**	**10**	**0,929**	**0,925**	**0,927**	**0,930**
20	13	20	0,921	0,913	0,916	0,920

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.4 Hasil Pengujian *Fold* 3 (Lanjutan)

k-values	*k-values* baru		*Precision*	*Recall*	*F-Measure*	*Accuracy*
	Positif	Negatif				
30	20	30	0,904	0,900	0,900	0,900
40	27	40	0,940	0,917	0,925	0,930
50	33	50	0,925	0,908	0,915	0,902
100	67	100	0,910	0,883	0,893	0,900
150	100	150	0,925	0,892	0,903	0,910

Tabel 5.4 menyatakan hasil *precision, recall, f-measure,* dan *accuracy* terhadap nilai *k-values* pada pengujian *fold* 3. Pada pengujian tersebut, nilai *f-measure* tertinggi diperoleh ketika nilai *k* sama dengan 10 dengan nilai *f-measure* sebesar 0,927, *precision* sebesar 0,929, dan *recall* sebesar 0,925, sedangkan nilai *f-measure* terendah diperoleh ketika nilai *k-values* sama dengan 100 dengan nilai *f-measure* sebesar 0,893, *precision* sebesar 0,910, dan *recall* sebesar 0,883.

5.2.4 Pengujian *5-Fold* Pada *Fold* Ke-4

Pengujian *fold* 4 dilakukan dengan menggunakan data latih sebanyak 400 dokumen, dan data uji sebanyak 100 dokumen. Hasil pengujian *fold* 4 dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil Pengujian *Fold* 4

k-values	*k-values* baru		*Precision*	*Recall*	*F-Measure*	*Accuracy*
	Positif	Negatif				
2	1	2	0,969	0,950	0,958	0,960
3	2	3	0,962	0,938	0,947	0,950
5	3	5	0,962	0,938	0,947	0,950
10	**7**	**10**	**0,984**	**0,975**	**0,979**	**0,980**
20	13	20	0,976	0,963	0,968	0,970
30	20	30	0,962	0,938	0,947	0,950
40	27	40	0,962	0,938	0,947	0,950
50	33	50	0,948	0,913	0,924	0,930
100	67	100	0,948	0,913	0,924	0,930
150	100	150	0,948	0,913	0,924	0,930

Tabel 5.5 menyatakan hasil *precision, recall, f-measure,* dan *accuracy* terhadap nilai *k-values* pada pengujian *fold* 4. Pada pengujian tersebut, nilai *f-measure* tertinggi diperoleh ketika nilai *k* sama dengan 10 dengan nilai *f-measure* sebesar 0,979, *precision* sebesar 0,984, dan *recall* sebesar 0,975, sedangkan nilai *f-measure* terendah diperoleh ketika nilai *k-values* sama dengan 50, 100, dan 150 dengan nilai *f-measure* sebesar 0,924, *precision* sebesar 0,948, dan *recall* sebesar 0,913.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.2.5 Pengujian *5-Fold* Pada *Fold* Ke-5

Pengujian *fold* 5 dilakukan dengan menggunakan data latih sebanyak 400 dokumen, dan data uji sebanyak 100 dokumen. Hasil pengujian *fold* 5 dapat dilihat pada Tabel 5.6.

Tabel 5.6 Hasil Pengujian *Fold* 5

k-values	*k-values* baru		*Precision*	*Recall*	*F-Measure*	*Accuracy*
	Positif	Negatif				
2	1	2	0,865	0,879	0,868	0,870
3	2	3	0,916	0,934	0,922	0,925
5	3	5	0,934	0,946	0,938	0,940
10	**7**	**10**	**0,967**	**0,971**	**0,969**	**0,970**
20	13	20	0,950	0,946	0,948	0,950
30	20	30	0,941	0,933	0,937	0,940
40	27	40	0,940	0,917	0,925	0,930
50	33	50	0,918	0,896	0,904	0,910
100	67	100	0,918	0,896	0,904	0,910
150	100	150	0,932	0,904	0,914	0,920

Tabel 5.6 menyatakan hasil *precision, recall, f-measure,* dan *accuracy* terhadap nilai *k-values* pada pengujian *fold* 5. Pada pengujian tersebut, nilai *f-measure* tertinggi diperoleh ketika nilai *k-values* sama dengan 10 dengan nilai *f-measure* sebesar 0,969, *precision* sebesar 0,967, dan *recall* sebesar 0,971, sedangkan nilai *f-measure* terendah diperoleh ketika nilai *k* sama dengan 2 dengan nilai *f-measure* sebesar 0,868, *precision* sebesar 0,865, dan *recall* sebesar 0,879.

5.2.6 Rata-Rata Pengujian Kelas Positif

Pengujian *5-fold* yang dilakukan menghasilkan nilai *precision, recall,* dan *f-measure* yang berbeda tiap kelas. Tabel 5.7 merupakan hasil rata-rata pengujian dari seluruh *5-fold* yang dilakukan terhadap kelas positif.

Tabel 5.7 Rata-Rata Pengujian Kelas Positif

k-values	*Precision*	*Recall*	*F-Measure*	*Accuracy*
2	0,881	0,865	0,870	0,896
3	0,923	0,885	0,901	0,923
5	0,931	0,890	0,908	0,928
10	**0,958**	**0,895**	**0,924**	**0,942**
20	0,957	0,875	0,913	0,934
30	0,972	0,850	0,906	0,926
40	0,982	0,825	0,896	0,924
50	0,971	0,815	0,886	0,916
100	0,976	0,775	0,862	0,902
150	0,981	0,765	0,858	0,900

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.7 menghasilkan nilai *f-measure* tertinggi ketika *k-values* sama dengan 10, dengan nilai *f-measure* sebesar 0,924, *precision* sebesar 0,958, dan *recall* sebesar 0,895, sedangkan nilai *f-measure* terendah adalah ketika *k-values* sama dengan 150 dengan nilai *f-measure* sebesar 0,870, *precision* sebesar 0,881, dan *recall* sebesar 0,865.

5.2.7 Rata-Rata Pengujian Kelas Negatif

Pengujian selanjutnya adalah pengujian terhadap kelas negatif. Pengujian ini dilakukan untuk mengetahui hasil akurasi yang dihasilkan oleh masing-masing kelas. Tabel 5.8 merupakan hasil rata-rata pengujian dari seluruh 5-*fold* yang dilakukan terhadap kelas negatif.

Tabel 5.8 Rata-Rata Pengujian Kelas Negatif

k-values	*Precision*	*Recall*	*F-Measure*	*Accuracy*
2	0,913	0,917	0,913	0,896
3	0,928	0,949	0,937	0,923
5	0,930	0,953	0,941	0,928
10	**0,934**	**0,973**	**0,953**	**0,942**
20	0,919	0,972	0,945	0,932
30	0,900	0,980	0,938	0,926
40	0,895	0,990	0,940	0,924
50	0,889	0,983	0,934	0,909
100	0,869	0,983	0,922	0,900
150	0,865	0,990	0,923	0,900

Tabel 5.8 menghasilkan nilai *f-measure* tertinggi ketika *k-values* sama dengan 10, dengan nilai *f-measure* sebesar 0,953, *precision* sebesar 0,934, dan *recall* sebesar 0,973, sedangkan nilai *f-measure* terendah adalah ketika *k* sama dengan 2 dengan nilai *f-measure* sebesar 0,913, *precision* sebesar 0,913, dan *recall* sebesar 0,917.

5.2.8 Rata-Rata Hasil Seluruh Pengujian

Rata-rata hasil pengujian merupakan rata-rata dari seluruh nilai *precision, recall,* dan *f-measure* pada pengujian *5-fold* yang dihitung untuk tiap nilai *k-values.* Hasil pengujian tersebut dijelaskan pada Tabel 5.9.

Tabel 5.9 Rata-Rata Hasil Seluruh Pengujian

k-values	*Precision*	*Recall*	*F-Measure*	*Accuracy*
2	0,897	0,891	0,891	0,896
3	0,926	0,917	0,919	0,923
5	0,931	0,922	0,924	0,928
10	**0,946**	**0,934**	**0,939**	**0,942**
20	0,938	0,924	0,929	0,933
30	0,936	0,915	0,922	0,926
40	0,939	0,908	0,918	0,924

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.9 Rata-Rata Hasil Seluruh Pengujian (Lanjutan)

k-values	*Precision*	*Recall*	*F-Measure*	*Accuracy*
50	0,930	0,899	0,910	0,912
100	0,920	0,879	0,891	0,901
150	0,923	0,878	0,890	0,900

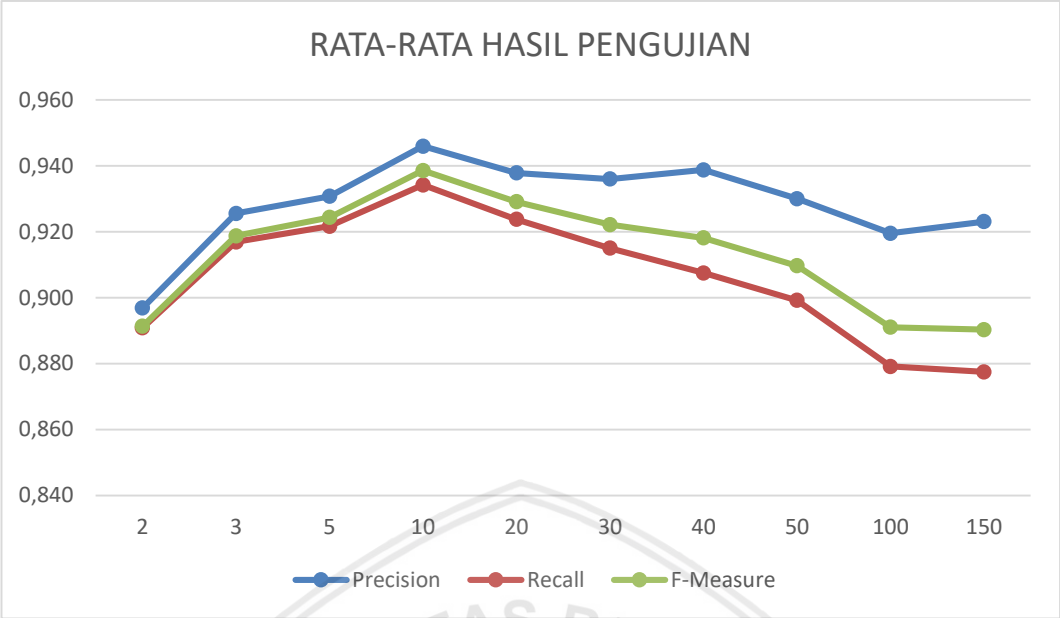
Pada Tabel 5.9 dapat disimpulkan bahwa nilai *f-measure* terbaik didapatkan ketika nilai *k* sama dengan 10 dengan nilai *f-measure* sebesar 0,939, nilai *precision* sebesar 0,946, dan nilai *recall* sebesar 0,934, sedangkan nilai *f-measure* terendah didapatkan ketika nilai *k* sama dengan 150 dengan nilai *f- measure* sebesar 0,890, nilai *precision* sebesar 0,923, dan nilai *recall* sebesar 0,878.

5.3 Analisis

Berdasarkan hasil evaluasi untuk masing-masing pengujian, dapat disimpulkan bahwa evaluasi yang memengaruhi klasifikasi dengan menggunakan metode *Improved K-Nearest Neighbor* adalah nilai *k-values*. Besar kecilnya nilai *k-values* dapat memengaruhi hasil klasifikasi yang dilakukan. Pada penelitian ini, didapatkan hasil bahwa *k*=10 menghasilkan nilai *f-measure* tertinggi yaitu sebesar 0,939, dan ketika *k*=150 menghasilkan nilai *f-measure* terendah yaitu sebesar 0,890. Ketika nilai *k-values* terlalu besar (*k-values*=150), atau terlalu kecil (*k-values*=2) akan menghasilkan nilai *f-measure* yang rendah. Nilai ketetanggaan atau nilai *n* yang rendah mengakibatkan proses klasifikasi hanya mengambil nilai ketetanggaan yang sedikit, sedangkan apabila nilai *n* terlalu besar juga akan mendapatkan hasil yang tidak optimal karena jumlah data dalam ketetanggaan yang terlalu banyak. Maka dari itu, dapat disimpulkan bahwa nilai *f-measure* pada penelitian ini cukup fluktuatif.

Selain itu, hasil evaluasi yang didapatkan pada masing-masing kelas tidak jauh berbeda dan menghasilkan nilai *f-measure* yang sama tinggi. Hal ini terjadi karena meskipun data jumlah atau proporsi data yang digunakan berbeda tetapi nilai *k-values* baru menyesuaikan dengan jumlah data tiap kelas. Maka dari itu hasil yang didapatkan masing-masing kelas tidak jauh berbeda. Rata-rata hasil seluruh pengujian ditunjukkan pada Gambar 5.1.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 5.1 Grafik Rata-Rata Hasil Pengujian

BAB 6 PENUTUP

Pada bab ini berisi kesimpulan yang didapatkan dari pembuatan sistem analisis sentimen para ulasan aplikasi BCA Mobile dan saran yang dapat diterapkan dalam penelitian selanjutnya yang menggunakan metode serupa.

6.1 Kesimpulan

Kesimpulan dari penelitian analisis sentimen pada ulasan aplikasi BCA Mobile yang telah dilakukan adalah sebagai berikut.

1. Metode BM25 dan *Improved K-Nearest Neighbor* dapat diterapkan pada sistem analisis sentimen pada ulasan aplikasi BCA M*obile,* sehingga dapat dengan mudah menentukan kelas suatu dokumen uji. Proses *pre-processing* data dilakukan untuk memudahkan data menjadi data yang terstruktur, sedangkan metode BM25 digunakan dalam proses pembobotan kata yang kemudian hasil dari pembobotan kata tersebut dilanjutkan dengan proses klasifikasi menggunakan metode *Improved K-Nearest Neighbor.*
2. Pengujian yang dilakukan pada analisis sentimen pada ulasan aplikasi BCA Mobile menghasilkan rata-rata nilai *f-measure* yang cukup tinggi yaitu sebesar 0,939, nilai *precision* sebesar 0,946, dan nilai *recall* sebesar 0,934. Hasil pengujian ini dipengaruhi oleh besarnya nilai *k-values* yang digunakan*,* namun jumlah dokumen atau perbandingan proporsi data latih yang digunakan dalam proses *training* tidak berpengaruh terhadap proses klasifikasi yang dilakukan. Hal ini dikarenakan pada proses klasifikasi menggunakan *Improved KNN* nilai *k-values* baru menyesuaikan dengan jumlah data latih tiap kelas.

6.2 Saran

Proses analisis sentimen ini menggunakan dokumen dengan beberapa kata yang tidak baku, sehingga kata yang berupa singkatan atau kata berbahasa asing yang seharusnya tidak diperlukan termasuk ke dalam term atau kata unik. Sehingga untuk mendapatkan hasil yang optimal seharusnya dilakukan normalisasi atau perbaikan kata berdasarkan kata yang baku atau ejaan yang disempurnakan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR REFERENSI

Baoli, L., Shiwen, Y. dan Qin, L., 2003. An Improved k-Nearest Neighbor Algorithm for Text Categorization. *Reading*, hal.678.

Febrianti, Y.M., Indriati dan Widodo, A.W., 2018. Analisis Sentimen Pada Ulasan " Lazada " Berbahasa Indonesia Menggunakan K-Nearest Neighbor (K-NN) Dengan Perbaikan Kata Menggunakan Jaro Winkler Distance. 2(10), hal.3689–3698.

Hardiyanto, E., Rahutomo, F. dan Puspitasari, D., 2016. Implementasi K Nearest Neighbor (KNN) pada Klasifikasi Artikel Wikipedia Indonesia.

Hastie, T., Tibshirani, R. dan Friedman, J., 2008. The Elements of Statistical Learning. Springer.

Herdiawan, 2015. Analisis Sentimen Terhadap TELKOM Indihome Berdasarkan Opini Publik Menggunakan Metode Improved K-Nearest Neighbor.

Indriati dan Ridok, A., 2016. Sentiment Analysis for Review Mobile Applications Using Neighbor Method Weighted K-Nearest Neighbor (Nwknn). *Journal of Environmental Engineering & Sustainable Technology 2016*, 03(01), hal.23–32.

Liu, B., 2016. Sentiment Analysis and Opinion Mining. *Encyclopedia of Machine Learning and Data Mining*, [daring] (May), hal.1–10. Tersedia pada: <http://link.springer.com/10.1007/978-1-4899-7502-7_907-1>.

Manning, C.D., Raghavan, P. dan Schutze, H., 2009. *Introduction to Information Retrieval*. [daring] *Computational Linguistics*, Tersedia pada: <http://www.mitpressjournals.org/doi/10.1162/coli.2009.35.2.307>.

Nugraha, P.D., Faraby, S. Al dan Adiwijaya, 2015. Klasifikasi Dokumen Menggunakan Metode k-Nearest Neighbor (kNN) dengan Information Gain. *Seminar Nasional Teknologi Informasi dan Multimedia. ISSN : 2302 - 3805*, 5(1), hal.1–14.

Pandie, E.S.Y., 2012. Sistem Informasi Pengambilan Keputuan Pengajuan Kredit Dengan Algoritma K-Nearest Neighbour (Studi Kasus: Koperasi Simpan Pinjam).

Pardede, J., Husada, M.G. dan Riansyah, R., 2015. Implementasi Dan Perbandingan Metode OKAPI BM25 Dan PLSA Pada Aplikasi Information Retrieval. hal.1–10.

Pramastuti, J.I.R., 2016. Sentimen Analisis Review Aplikasi Mobile Menggunakan Metode Ensemble Classifier Berbasis Average Of Probabilities.

Purwanto, A., Maulana, K. dan A, Y.F., 2011. Analisis dan implementasi penggabungan binary independence retrieval dan okapi bm25 untuk perankingan dokumen dalam sistem temu balik informasi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Putri, P.A., Ridok, A. dan Indriati, 2013. Implementasi Metode Improved K-Nearest Neighbor Pada Analisis Sentimen Twitter Berbahasa Indonesia.

Sianturi, M.H., Marji and Sutrisno, 2014. Perbandingan Kinerja Metode Naiive Bayesian Dengan Metode Improved K-Nearest Neighbor Dalam Implementasi Sistem Pengklasifikasian Spam Email

Talib, R., Hanif, M.K., Ayesha, S. dan Fatima, F., 2016. Text Mining: Techniques, Applications and Issues. *IJACSA) International Journal of Advanced Computer Science and Applications*, [daring] 7(11), hal.414–418. Tersedia pada: <www.ijacsa.thesai.org>.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA