IMPLEMENTASI METODE IMPROVED K-NEAREST NEIGHBOR PADA ANALISIS SENTIMEN TWITTER BERBAHASA INDONESIA

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
PRIMA ARFIANDA PUTRI
NIM. 0910680088

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
MALANG
2013

LEMBAR PERSETUJUAN

IMPLEMENTASI METODE IMPROVED K-NEAREST NEIGHBOR PADA ANALISIS SENTIMEN TWITTER BERBAHASA INDONESIA

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh:

PRIMA ARFIANDA PUTRI

NIM. 0910680088

Menyetujui :

Pembimbing I	**Pembimbing II**
Drs.Achmad Ridok, M.Kom	**Indriati, S.T, M.Kom**
NIP. 196808251994031002	NIK. 83101306120035

LEMBAR PENGESAHAN

IMPLEMENTASI METODE IMPROVED K-NEAREST NEIGHBOR PADA ANALISIS SENTIMEN TWITTER BERBAHASA INDONESIA

SKRIPSI

Laboratorium Komputasi Cerdas dan Visualisasi

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

PRIMA ARFIANDA PUTRI

NIM. 0910680088

Skripsi ini telah diuji dan dinyatakan lulus tanggal 5 Juli 2013

Penguji I	**Penguji II**
Lailil Muflikhah, S.Kom., M.Sc	**Dian Eka Ratnawati, S.Si, M.Kom**
NIP. 19741113 200501 2 001	**NIP. 19730619 200212 2 001**

Penguji III

Rekyan Regasari MP, S.T., M.T.
NIP. 77041406120253

Mengetahui
Ketua Program Studi Informatika

Drs. Marji., M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 5 Juli 2013

Mahasiswa,

Prima Arfianda Putri

NIM. 0910680088

KATA PENGANTAR

Alhamdulillahi rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul "Implementasi Metode Improved K-Nearest Neighbor pada Analisis Sentimen Twitter Berbahasa Indonesia".

Dalam pelaksanaan dan penulisan skripsi ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moral maupun material. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Bapak Drs. Achmad Ridok, M.Kom dan Ibu Indriati, S.T., M.Kom selaku dosen pembimbing selama pelaksanaan skripsi.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Ayahanda, Ibunda, dan seluruh keluarga atas segenap dukungan dan kasih sayang yang telah diberikan.
5. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
6. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
7. Teman-teman angkatan 2009 dan Konsentrasi Cerdas dan Visualisasi dan seluruh pihak yang telah membantu kelancaran penulisan skripsi yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik membangun senantiasa diharapkan guna perbaikan bagi penelitian selanjutnya. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak, Aamiin.

Malang, Maret 2013

Penulis

ABSTRAK

Twitter merupakan mikroblog yang sedang digemari dan berubah menjadi penyebar informasi yang sangat cepat saat ini. Informasi yang dihasilkan dan beredar melalui media ini sangat bebas dan beragam seperti berita, pertanyaan, opini, komentar, kritik baik yang bersifat positif maupun negatif. Analisis sentimen merupakan salah satu cabang penelitian pada domain *Text Mining* atau penggalian data berupa teks, yang diantaranya terdapat proses mengolah dan mengekstrak data tekstual secara otomatis untuk mendapatkan informasi. Manfaat analisis sentimen dalam dunia usaha antara lain untuk melakukan pemantauan terhadap sebuah produk. Analisis sentimen dapat digunakan sebagai alat bantu untuk melihat respon konsumen atau masyarakat terhadap suatu produk tertentu, sehingga dapat segera diambil langkah-langkah strategis berikutnya. Proses pada analisis sentimen diawali dengan *preprocessing*, dilanjutkan dengan pembobotan kata, kemudian pengkategorian yang terdiri dari penghitungan *cosine similarity* dan klasifikasi sentimen.

Preprocessing terdiri dari beberapa tahap yaitu pembersihan dokumen, *tokenizing, stopword removal,* dan *stemming*. Metode pembobotan kata yang digunakan pada skripsi ini adalah *Term Frequecny – Inverse Document Frequency* (TF-IDF) dan menggunakan *Improved K-Nearest Neighbor* (KNN) sebagai metode klasifikasinya. Metode *Improved* KNN memiliki kelebihan berupa kestabilan pada berapapun variasi nilai k.

Hasil yang telah diperoleh melalui implementasi dan pengujian sistem adalah jumlah data latih, keseimbangan proporsi kategori data latih, dan nilai k berpengaruh terhadap ketepatan hasil analisis sentimen. Rata-rata *precision* yang diperoleh sistem sebesar 82%, rata-rata *recall* sebesar 87%, dan rata-rata *F-measure* sebesar 84%, sehingga dapat disimpulkan efektivitas sistem sudah berjalan dengan relatif baik.

Kata kunci: Analisis Sentimen, *Text Mining, Improved K-Nearest Neighbor* (KNN)

ABSTRACT

Twitter is a popular microblog and turned into quickly information spreader nowadays. Information circulated through this media is very free and diverse such as news, questions, opinions, comments, critisism both positive or negative. Sentiment analysis is a branch of research in text mining or data mining domain, which contains process and extract textual data automatically to obtain information. One of the benefit of sentiment analysis in the business world is for monitoring a product. Sentiment analysis can be used as a tool to see customer response or a particular community to a product, so the next strategic steps can be taken. The process of sentiment analysis begins with the preprocessing, followed by a weighting word, then categorization consist of cosine similarity calculation and sentiment classification.

Preprocessing consist of several steps, document cleaning, tokenizing, stopword removal, and stemming. Term weighting method used in this research is Term Frequency – Inverse Document Frequency (TF-IDF) and using Improved K-Nearest Neighbor (KNN) as the classification method. Improved KNN's advantage is on the stability on regardless variations of k-values.

The result obtained through implementation and testing are: the amounts of data training set, the balance of the proportion the data training set cateogry, and k-values are influence to the accuracy of the sentiment analysis' results. The average of the precision obtained by system is 82%, where the average of the recall is 87%, and the average of F-measure is 84% so it can be conclude that system's effectiveness has been running relatively well.

Keyword: *Sentiment Analysis, Text Mining, Improved K-Nearest Neighbor (KNN)*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR *SOURCE CODE*

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Salah satu jejaring sosial berupa mikroblog yang saat ini sedang diminati oleh banyak orang adalah *Twitter*, dimana memungkinkan penggunanya untuk mengirim dan membaca pesan singkat yang disebut *tweets*. *Tweets* dapat dibaca secara bebas, namun dapat pula diatur hanya dapat dilihat oleh pengguna lain yang mengikuti *Twitter*-nya atau yang disebut *follower*. *Twitter* bertindak sebagai media penyebar informasi yang sangat cepat seiring bertambahnya pengguna *Twitter*. Informasi yang dihasilkan dan beredar melalui media ini sangat bebas dan beragam seperti berita, pertanyaan, opini, komentar, kritik baik yang bersifat positif maupun negatif. Kita dapat melihat bagaimana pendapat orang lain terhadap suatu permasalahan melalui analisis sentimen, sehingga dapat membantu kita dalam mengambil sebuah keputusan dengan lebih cepat dan tepat.

Analisis sentimen merupakan salah satu cabang penelitian pada domain *Text Mining* atau penggalian data berupa teks, yang diantaranya terdapat proses mengolah dan mengekstrak data tekstual secara otomatis untuk mendapatkan informasi [BPL-08]. Analisis sentimen dapat digunakan sebagai alat bantu dalam mengidentifikasi kecenderungan sesuatu hal yang ada di pasar [BPL-02]. Jonathon Read melakukan penelitian klasifikasi sentimen terhadap teks berbahasa Inggris pada *UseNet Newsgroup* dengan menggunakan *emoticon* seperti ":-)" dan ":-(" sebagai *classifier*nya dan membagi sentimen menjadi dua (positif dan negatif) dan menghasilkan akurasi 70% dengan menggunakan metode Naive Bayes dan SVM [JNR-05]. Manfaat analisis sentimen dalam dunia usaha antara lain untuk melakukan pemantauan terhadap sebuah produk. Analisis sentimen dapat digunakan sebagai alat bantu untuk melihat respon konsumen atau masyarakat terhadap suatu produk tertentu. Sehingga dapat segera diambil langkah-langkah strategis berikutnya.

Dari hasil survey terhadap lebih dari 2000 orang Amerika dewasa diketahui 81% pengguna internet melakukan penelitian terhadap suatu produk

secara online minimal satu kali, 20% melakukan hal tersebut setiap hari, *review* terhadap rumah makan, hotel, agen perjalanan wisata, dan dokter di internet dapat meningkatkan penjualan antara 73% sampai dengan 87%, pelanggan bersedia membayar lebih sebesar 20% sampai 99% terhadap *review* di internet yang mendapatkan bintang 5 daripada bintang 4 [BPL-08]. Pengaruh dan manfaat dari sentimen sedemikian besar sehingga penelitian ataupun aplikasi mengenai analisis sentimen berkembang sangat pesat. Terdapat kurang lebih 20-30 perusahaan di Amerika yang fokus pada layanan analisis sentimen [BGL-10]. Faktor keuntungan tersebut mendorong perlunya dilakukan penelitian analisis sentimen terhadap dokumen berbahasa Indonesia.

K-Nearest Neighbor (KNN) merupakan salah satu metode *machine learning* yang melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Metode ini sangat sederhana, mudah direpresentasikan, memiliki ketangguhan terhadap *training data* yang memiliki banyak *noise*, dan cukup efektif untuk proses pengelompokan [TSK-06]. Algoritma ini memiliki kelemahan pada akurasi, karena nilai k ditetapkan sama pada semua kategori tanpa memperhitungkan jumlah dokumen latih yang dimiliki masing-masing kategori, sedangkan distribusi dokumen latih dalam *training set* tidak sama. Namun kelemahan ini dapat diatasi dengan menggunakan algoritma *Improved K-Nearest Neighbor*. Penelitian oleh Baoli,Shiwen,dan Qin untuk dokumen teks berbahasa Cina menunjukkan bahwa dengan algoritma *Improved k-Nearest Neighbor*, didapatkan kestabilan pada proses kategorisasi dengan berapapun variasi k-values [BSQ-03].

Berdasarkan fenomena di atas, penulis tertarik untuk membuat suatu sistem yang dapat mengelompokkan *tweets* berbahasa Indonesia ke dalam dua sentimen yaitu positif, dan negatif pada *Twitter* berbahasa Indonesia yang bertema penyedia layanan GSM menggunakan metode *Improved K-Nearest Neighbor*. Implementasi dari sistem ini adalah dengan menggunakan bahasa pemrograman PHP.

1.2 Rumusan Masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka rumusan masalah dapat disusun sebagai berikut:

a. Bagaimana penerapan metode *Improved K-Nearest Neighbor* pada analisis sentimen *Twitter* berbahasa Indonesia.
b. Bagaimana akurasi pada perangkat lunak yang akan dibangun dengan menggunakan *Precision, Recall, dan F-measure*.

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal:

1. Obyek yang diteliti pada perancangan sistem ini adalah *Twitter*, dengan tema *tweets* yang digunakan sebagai dataset adalah mengenai salah satu penyedia layanan GSM di Indonesia.
2. Bahasa yang diteliti pada sistem ini adalah Bahasa Indonesia yang sesuai dengan ejaan yang disempurnakan (EYD).
3. Solusi yang dihasilkan dibatasi pada perancangan sistem ini adalah pengelompokan *tweet* tertentu sebagai *tweet* positif atau negatif.
4. Penelitian ini hanya membahas metode *Improved K-Nearest Neighbor* sebagai algoritma untuk pengklasifikasian.
5. Analisis sentimen pada *Twitter* berbahasa Indonesia ini berdasarkan pada perbandingan kemunculan kata dokumen, bukan berdasarkan analisis semantic.
6. Pengujian sistem dilakukan dengan menggunakan *Precision*, *Recall*, dan *F-measure*.

1.4 Tujuan

Tujuan yang ingin dicapai dari pembuatan skripsi ini adalah menerapkan metode klasifikasi *Improved K-Nearest Neighbor* pada analisis sentimen *Twitter* berbahasa Indonesia.

1.5 Manfaat

Penulisan skripsi ini diharapkan mempunyai manfaat yang baik dan berguna bagi pembaca dan penulis. Adapun manfaat yang diharapkan adalah sebagai berikut:

1. Bagi Penulis
   a. Menerapkan ilmu yang telah diperoleh dari Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
   b. Mendapatkan pemahaman tentang perancangan dan pengembangan Sistem Analisis Sentimen pada *Twitter* berbahasa Indonesia dengan Metode *Improved K-Nearest Neighbor*.
2. Bagi Pengguna

   Manfaat skripsi ini adalah memudahkan pengguna dalam melakukan pengelompokan *tweets* ke dalam kategori positif dan negatif sehingga dapat digunakan untuk mengamati kecenderungan sentimen atas suatu isu agar selanjutnya dapat dilakukan tindakan secara cepat dan tepat.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, metodologi pembahasan, dan sistematika penulisan.

BAB II Kajian Pustaka dan Dasar teori

Menguraikan tentang dasar teori dan referensi yang mendasari pembuatan sistem analisis sentimen pada Twitter berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor*.

BAB III Metode Penelitian dan Perancangan

Metode penelitian menguraikan tentang metode dan langkah kerja yang dilakukan dalam penulisan tugas akhir yang terdiri dari studi literatur, perancangan sistem perangkat lunak, implementasi sistem perangkat lunak, pengujian dan analisis, serta penulisan laporan. Sedangkan perancangan berisi tentang perencanaan aplikasi yang dibuat, meliputi deskripsi aplikasi, spesifikasi kebutuhan, dan

perancangan analisis sentimen pada Twitter berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor*.

BAB IV Implementasi

Membahas implementasi dari sistem analisis sentimen pada Twitter berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor* yang sesuai dengan perancangan sistem yang telah dibuat.

BAB V Pengujian dan Analisis

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VI Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi kajian pustaka dan pembahasan tentang teori dasar yang berhubungan dengan sistem analisis sentimen pada *Twitter* berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor*. Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya yang berjudul '*Twitter as a Corpus for Sentiment Analysis and Opinion Mining*'. Teori dasar yang akan dibahas pada bab ini yaitu konsep dasar *Text Mining* dan *preprocessing* yang terdiri dari *pembersihan dokumen, parsing, tokenizing, filtering/stopword removal*, dan *stemming*. Analisis Sentimen yang terdiri dari *term weighting* dan *classification.*

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini adalah membahas penelitian sebelumnya yang berjudul '*Twitter as a Corpus for Sentiment Analysis and Opinion Mining*' [APP-10]. Penelitian ini membahas tentang analisis sentimen pada mikroblog *Twitter* berbahasa inggris. Penelitian ini menggunakan unigram, bigram, dan trigram sebagai metode untuk pembobotannya. Metode klasifikasi yang digunakan pada penelitian ini adalah Naïve Bayes dan *Support Vector Machine* (SVM). Penelitian ini secara otomatis mengumpulkan korpus dan menggunakan korpus untuk membangun *classifier* sentimen yang mampu menentukan sentimen positif, negatif, dan netral.

Alec Go, Lei Huang, dan Richa Bayani melakukan penelitian pada *microblog Twitter* berbahasa Inggris dengan pendekatan yang serupa. Penelitiannya menghasilkan akurasi 81% dengan menggunakan Naive Bayes *Classifier* untuk klasifikasi ke dalam dua kelas sentimen yaitu positif dan negatif, sedangkan menunjukkan hasil yang kurang baik untuk tiga kelas sentimen (positif, negatif, dan netral) [ALG-09].

Perbedaan yang dibuat penulis pada skripsi ini adalah pada penggunaan implementasi analisis sentimen dikhususkan untuk teks berbahasa Indonesia. Perbedaan ini menyebabkan proses *stemming* pada *preprocessing* yang dilakukan akan berbeda dengan penelitian sebelumnya. Hal tersebut dikarenakan bahasa Indonesia memiliki morfologi yang berbeda dengan bahasa Inggris. Proses *stemming* pada penelitian yang akan dibuat ini akan meggunakan *stemming* arifin-setiono, yang dikenal cukup efektif untuk proses *stemming* pada teks berbahasa Indonesia. Skripsi ini akan menggunakan metode TF-IDF untuk pembobotan *term*. Metode klasifikasi yang digunakan pada skripsi ini adalah *Improved K-Nearest Neighbor* dengan *classifier* sentimen yang mampu menentukan sentimen positif dan negatif. Pengujian pada skripsi ini menggunakan *precision, recall,* dan *F-measure*.

2.2 Text Mining

Text mining dapat didefinisikan secara luas sebagai proses pengetahuan intensif di mana pengguna berinteraksi dengan koleksi dokumen dari waktu ke waktu dengan menggunakan berbagai macam analisis. Dalam cara yang sejalan dengan *data mining*, *text mining* berusaha mengekstrak informasi yang berguna dari sumber data melalui identifikasi dan eksplorasi *patterns*. *Text mining* menjadi menarik karena sumber data koleksi dokumen dan pola yang menarik tidak ditemukan dari *database* formal namun ditemukan dalam data tekstual yang tidak terstruktur pada dokumen dalam koleksi [FRS-07].

Algoritma yang digunakan pada *text mining*, tidak hanya melakukan perhitungan hanya pada dokumen, tetapi juga fitur. Terdapat empat macam fitur yang sering digunakan [FRS-07]:

a. *Character*

Character merupakan komponen individual yang dapat berupa huruf, angka, karakter spesial, dan spasi. *Character* juga merupakan blok pembangun pada level paling tinggi pembentuk semantik fitur, seperti kata, *term*, dan *concept*. Pada umumnya, representasi *character-based* ini jarang digunakan pada beberapa teknik pemrosesan teks.

b. *Words*

Kata-kata tertentu dipilih langsung dari sebuah dokumen "asli" berada pada apa yang mungkin digambarkan sebagai tingkat dasar semantik. Untuk alasan ini, fitur *word* kadang kala terdapat di dalam dokumen asli itu sendiri.

c. *Terms*

Merupakan *single word* dan frasa *multi word* yang terpilih secara langsung dari korpus. Representasi *term-based* dari dokumen tersusun dari subset *term* dalam dokumen.

d. *Concept*

Merupakan fitur yang digenerate dari sebuah dokumen secara manual, *rule-based*, atau metodologi lain.

2.2.1 Pembersihan Dokumen

Pembersihan dokumen adalah proses membersihkan dokumen dari karakter-karakter yang tidak diperlukan untuk mengurangi *noise* seperti emotikon dan simbol-simbol.

2.2.2 *Parsing*

Parsing adalah proses untuk memecah teks bebas yang besar menjadi bagian-bagian yang disebut kalimat [CDF-08]. Kalimat-kalimat yang dihasilkan kemudian dipecah lagi menjadi kata-kata melalui proses Tokenizing.

2.2.3 *Tokenizing*

Proses *Tokenizing* memotong setiap kata dalam teks, dan mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai ‘z’ yang diterima, sedangkan karakter selain huruf dihilangkan. Hasil dari proses *tokenizing* adalah kata-kata yang merupakan penyusun kalimat [AHA-10].

2.2.4 *Filtering*/*Stopword Removal*

Sebuah proses penyaringan untuk menghilangkan kata yang tidak relevan pada hasil *Tokenizing* sebuah dokumen teks dengan cara membandingkannya dengan *Stoplist (Stopword list)* yang ada [AHA-10]. Contoh dari *Stopword* misalnya, kata sambung, artikel dan preposisi.

2.2.5 *Stemming*

Stemming merupakan suatu proses untuk menemukan kata dasar dari sebuah kata dengan menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi Bahasa Indonesia yang baik dan benar [DGT-09]. *Stemming* yang digunakan pada penelitian ini adalah *stemming* arifin-setiono, yang sudah banyak digunakan untuk proses *stemming* pada teks berbahasa Indonesia.

2.2.5.1 Stemming Arifin Setiono

Algoritma ini didahului dengan pembacaan tiap kata dari file sampel. Sehingga input dari algoritma ini adalah sebuah kata yang kemudian dilakukan [ASA-01]:

1. Pemeriksaan semua kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 Awalan (*prefiks*) dan 3 Akhiran (*sufiks*). Sehingga bentuknya menjadi:

   Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

   Seandainya kata tersebut tidak memiliki imbuhan sebanyak imbuhan di atas, maka imbuhan yang kosong diberi tanda ‘x’ untuk *prefiks* dan diberi tanda ‘xx’ untuk *sufiks*.

   Pemotongan dilakukan secara berurutan sebagai berikut :

   AW : AW (Awalan)

   AK : AK (Akhiran)

   KD : KD (Kata Dasar)

   a. AW I, hasilnya disimpan pada p1

b. AW II, hasilnya disimpan pada p2
c. AK I, hasilnya disimpan pada s1
d. AK II, hasilnya disimpan pada s2
e. AK III, hasilnya disimpan pada s3

2. Setiap tahap pemotongan di atas diikuti dengan pemeriksaan di kamus apakah hasil pemotongan itu sudah berada dalam bentuk dasar. Jika pemeriksaan ini berhasil, proses dinyatakan selesai dan tidak perlu melanjutkan proses pemotongan imbuhan lainnya [ASA-01]. Contoh pemenggalan kata "mempermainkannya"

- Langkah 1 :
  Cek apakah kata ada dalam kamus
  Ya : Success
  Tidak : lakukan pemotongan AW I
  Kata = permainkannya
- Langkah 2 :
  Cek apakah kata ada dalam kamus
  Ya : Success
  Tidak : lakukan pemotongan AW II
  Kata = mainkannya
- Langkah 3 :
  Cek apakah kata ada dalam kamus
  Ya : Success
  Tidak : lakukan pemotongan AK I
  Kata = mainkan
- Langkah 4 :
  Cek apakah kata ada dalam kamus
  Ya : Success
  Tidak : lakukan pemotongan AK II
  Kata = main

- Langkah 5 :

  Cek apakah kata ada dalam kamus

  Ya : Success

  Tidak : lakukan pemotongan AK III. Dalam hal ini AK III tidak ada, sehingga kata tidak diubah.

  Kata = main

- Langkah 6

  Cek apakah kata ada dalam kamus

  Ya : Success

  Tidak : "Kata tidak ditemukan"

3. Jika sampai pada pemotongan AK III belum juga ditemukan di kamus, dilakukan proses kombinasi. KD yang dihasilkan dikombinasikan dengan imbuhan-imbuhannya dalam dua belas konfigurasi berikut:

a. KD
b. KD + AK III
c. KD + AK III + AK II
d. KD + AK III + AK II + AK I
e. AW I + AW II + KD
f. AW I + AW II + KD + AK III
g. AW I + AW II + KD + AK III + AK II
h. AW I + AW II + KD + AK III + AKII + AKI
i. AW II + KD
j. AW II + KD + AK III
k. AW II + KD + AK III + AK II
l. AW II + KD + AK III + AK II + AK I

Kombinasi a, b, c, d, h, dan l sudah diperiksa pada tahap sebelumnya karena kombinasi ini adalah hasil pemotongan bertahap tersebut. Dengan demikian, kombinasi yang masih perlu dilakukan tinggal 6 yakni pada kombinasi-kombinasi yang belum dilakukan (e, f, g, i, j, dan k). Jika hasil pemeriksaan suatu kombinasi adalah ‘ada’, pemeriksaan pada kombinasi lainnya sudah tidak diperlukan lagi.

Pemeriksaan dua belas kombinasi ini diperlukan karena adanya fenomena *overstemming* pada algoritma pemotongan imbuhan. Kelemahan ini berakibat pada pemotongan bagian kata yang sebenarnya adalah milik kata dasar itu sendiri yang kebetulan mirip dengan salah satu jenis imbuhan yang ada. Pemotongan yang sudah terlanjur tersebut dapat dikembalikan sesuai posisinya dengan dua belas kombinasi itu.

2.3 Analisis Sentimen

Analisis sentimen yang merupakan bagian dari *opinion mining*, adalah proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi [BPL-08]. Dilakukan untuk melihat pendapat terhadap sebuah masalah, atau dapat juga digunakan untuk identifikasi kecenderungan hal di pasar [BPL-02]. Analisis sentimen dalam penelitian ini adalah proses klasifikasi dokumen tekstual ke dalam dua kelas, yaitu kelas sentimen positif dan negatif. Besarnya pengaruh dan manfaat dari analisis sentimen, menyebabkan penelitian ataupun aplikasi mengenai analisis sentimen berkembang pesat, bahkan di Amerika kurang lebih 20-30 perusahaan yang memfokuskan pada layanan analisis sentimen [BGL-10]. Pada dasarnya analisis sentimen merupakan klasifikasi, tetapi kenyataannya tidak semudah proses klasifikasi biasa karena terkait penggunaan bahasa. Dimana terdapat ambigu dalam penggunaan kata, tidak adanya intonasi dalam sebuah teks, dan perkembangan dari bahasa itu sendiri [BGL-10].

Manfaat analisis sentimen dalam dunia usaha antara lain untuk melakukan pemantauan terhadap sebuah produk. Analisis sentimen dapat digunakan sebagai alat bantu untuk melihat respon konsumen atau masyarakat terhadap suatu produk tertentu. Sehingga dapat segera diambil langkah-langkah strategis berikutnya. Analisis sentimen dapat juga digunakan sebagai media untuk mengamati tanggapan masyarakat terhadap isu tertentu, sehingga dapat pula digunakan untuk melihat respon politik. Faktor-faktor keuntungan tersebut mendorong perlunya dilakukan penelitian analisis sentimen terhadap dokumen berbahasa Indonesia. Penelitian analisis sentimen dalam skripsi ini dilakukan dengan menggunakan

pendekatan dalam *machine learning* yang dikenal dengan Metode *Improved K-Nearest Neighbor* dan dikhususkan pada dokumen teks berbahasa Indonesia.

2.3.1 Term Weighting

Pembobotan dilakukan untuk mendapatkan nilai dari kata/*term* yang berhasil diekstrak. Pada skripsi ini, penulis menggunakan metode TF-IDF sebagai proses pembobotan. Pada tahap ini, setiap dokumen diwujudkan sebagai sebuah vektor dengan elemen sebanyak term yang berhasil dikenali dari tahap ekstraksi dokumen di atas. Vektor tersebut beranggotakan bobot dari setiap *term* yang dihitung berdasarkan metode TF-IDF [YYJ-99].

2.3.1.1 *Term Frequency* (TF)

Term Frequency (TF) adalah jumlah kemunculan sebuah *term* pada sebuah dokumen. Jika sebuah *term* *t* sering muncul pada sebuah dokumen, maka *query* yang mengandung t harus mendapatkan dokumen tersebut. *Term frequency* ini didasari pada aspek lokal pada TF-IDF *monotonicity*.

Local weight, atau yang biasa disebut denganTF (*term frequency*) berfungsi untuk menentukan bobot dari *term* t pada dokumen tertentu, yang pada dasarnya menghasilkan estimasi berdasarkan frekuensi atau *relative frequency* dari term t pada dokumen tersebut [KAO-07].

Ada empat cara yang bisa digunakan untuk mendapatkan nilai TF yaitu [SAL-12]:

1. *Raw* TF

   Pada *Raw* TF ini, nilai TF sebuah *term* dihitung berdasarkan kemunculan *term* tersebut dalam dokumen.

2. *Logarithmic* TF

   Dalam memperoleh nilai TF, cara ini menggunakan fungsi logaritmik pada matematika.

$$tf = 1 + \log(tf), \qquad (2\text{-}1)$$

dimana tf adalah kemunculan kata pada dokumen.

3. *Binary* TF

   Cara ini akan menghasilkan nilai boolean berdasarkan kemunculan *term* pada dokumen tersebut. Bernilai 0 apabila *term* tidak ada pada sebuah dokumen, dan bernilai 1 apabila *term* tersebut ada dalam dokumen. Sehingga banyaknya kemunculan *term* pada sebuah dokumen tidak berpengaruh.

4. *Augmented* TF

$$tf = 0{,}5 + 0{,}5\frac{tf}{\max(tf)}, \qquad (2\text{-}2)$$

   Dimana nilai *tf* adalah jumlah kemunculan *term* pada sebuah dokumen dan nilai max(*tf*) adalah jumlah kemunculan terbanyak *term* pada dokumen yang sama.

Pada implementasi skripsi ini, cara yang dipakai untuk mendapatkan nilai TF adalah dengan menggunakan *raw* TF.

2.3.1.2 *Inverse Document Frequency* (IDF)

Inverse Document Frequency (IDF) adalah jumlah dokumen yang mengandung sebuah *term* yang dicari dari kumpulan dokumen yang ada. IDF ini didasari pada aspek global pada TF-IDF *monotoncity* [SAL-12].

Global weight atau biasa yang disebut dengan istilah IDF mendefinisikan kontribusidari term t ke dokumen tertentu dalam sebuah *global sense* [KAO-07].

$$idf = \log(\frac{N}{df}), \qquad (2\text{-}3)$$

dimana:

N: jumlah dokumen yang terdapat pada kumpulan dokumen

df: jumlah dokumen yang mengandung *term*

Pembobotan TF-IDF untuk sebuah *term* t dari dokumen D didapat dari perkalian TF dan IDF.

2.3.1.3 TF-IDF

Metode TF-IDF ini merupakan metode pembobotan dalam bentuk sebuah metode yang merupakan integrasi antar *term frequency* (TF), dan *inverse document frequency* (IDF) [YYJ-99]. Metode TF-IDF dapat dirumuskan sebagai berikut:

$$w(t,d) = tf(t,d) * idf \quad (2\text{-}4)$$

Fungsi metode ini adalah untuk mencari representasi nilai dari tiap-tiap dokumen dari suatu kumpulan data *training (training set)* dimana nantinya akan dibentuk suatu vektor antara dokumen dengan kata (*documents with terms*) yang kemudian untuk kesamaan antar dokumen dengan *cluster* akan ditentukan oleh sebuah *prototype* vektor yang disebut juga dengan *cluster centroid* [YYJ-99].

2.3.2 Klasifikasi (*Classification*)

Metode klasifikasi K-Nearest Neighbor (KNN) merupakan salah satu metode yang umum digunakan sebagai algoritma pengelompokan. Algoritma ini tidak membangun model dalam proses pelatihannya dan hanya mengandalkan memori. Algoritma ini akan mengelompokkan data-data dengan mengidentifikasi sejumlah k tetangga terdekat kemudian mengambil mayoritas kelas terbanyak dalam menentukan jenis kelas dari data tersebut [HEL-09].

Cara identifikasi sejumlah k tetangga terdekat adalah dengan menghitung jarak ke seluruh data training kemudian ditentukan sejumlah k data yang mempunyai jarak terdekat dengan data test tersebut. Nilai k di sini adalah bilangan integer positif yang dimulai dari satu [HEL-09].

2.3.2.1 *Cosine Similarity*

Metode *Cosine Similarity* adalah metode untuk menghitung kesamaan dari dua dokumen. Untuk menyamakan frekuensi jangka setiap kata pada kalimat yang ada digunakan pembobotan, proses pembobotan mengekstrak dokumen menjadi proses yang terdiri dari kumpulan kata perkalimat. Tujuannya adalah menyamakan kedua kalimat pada suatu dokumen yang nantinya akan

dibandingkan, sehingga kita dapat melangkah ke tahap selanjutnya yaitu tahapan *similarity* [RIZ-12].

Penentuan kesesuaian dokumen dengan query dipandang sebagai pengukuran (*similarity measure*) antara vector dokumen (D) dengan vector query (Q). Semakin sama suatu vector dokumen dengan vector query maka dokumen dapat dipandang semakin sesuai dengan query [RIZ-12].

Rumus yang digunakan untuk menghitung *cosine similarity* adalah sebagai berikut [YYX-09]:

$$CosSim(X, d_j) = \frac{\sum_{i=1}^{m} x_i . d_{ji}}{\sqrt{\left(\sum_{i=1}^{m} x_i\right)^2} . \sqrt{\left(\sum_{i=1}^{m} d_{ji}\right)^2}}, \quad (2\text{-}5)$$

dimana:

X : dokumen uji

d_j : dokumen *training*

x_i dan d_{ji} : nilai bobot yang diberikan pada setiap *term* pada dokumen

Kedekatan *query* dan dokumen diindikasikan dengan sudut yang dibentuk. Nilai *cosinus* yang cenderung besar mengindikasikan bahwa dokumen cenderung sesuai *query* [RIZ-12]. Dalam proses membandingkan dokumen yang sesuai dengan dokumen yang telah ada atau dokumen lainnya, maka digunakan perhitungan dengan rumus pada persamaan (2-5) untuk mengetahui angka similaritas dari dokumen tersebut.

2.3.2.2 KNN Decision Rule

Pada proses pengambilan keputusan diperlukan suatu nilai *k* yang akan digunakan untuk memilih kategori yang sesuai. Konsep yang digunakan untuk pengambilan keputusan yaitu dengan mengurutkan hasil perhitungan kemiripan *cosSim*(X,d_j) dimulai dari yang besar. Untuk k=1, dipilih nilai *cosSim*(X,d_j) pada urutan paling atas, dan untuk k>1 dipilih sebanyak k urutan teratas [YYX-09].

Sebagai contoh, terdapat sebuah dokumen X yang akan dikategorikan berdasarkan pada sekumpulan dokumen yang ada pada dokumen latih di d∈T. Misalnya k=1, nilai kemiripan antara X dengan dokumen latih d telah ditentukan, maka dipilih d yang memiliki nilai kemiripan yang paling tinggi. Proses ini dijelaskan pada persamaan (2-6).

$SIM_{max}(X) = \max_{d \in T} SIM(X, d_j)$. (2-6)

dimana $SIM_{max}(X)$ adalah nilai kemiripan dokumen X yang paling tinggi. $SIM(X,d_j)$ adalah nilai kemiripan antara dokumen X dengan dokumen latih d. Sedangkan $\max_{d \in T}$ $SIM(X,d_j)$ adalah nilai maksimum kemiripan dokumen X dengan dokumen d yang merupakan bagian dari dokumen latih T [AXB-01].

Jika digunakan k>1, maka penentuan kategorinya adalah dengan menjumlahkan semua nilai kemiripan $SIM(X,d_j)$ yang termasuk dalam suatu kategori. Perhitungan dilakukan dengan persamaan (2-7). Dokumen X masuk ke dalam kategori yang memiliki nilai P(X, C_m) paling besar [YYX-09].

$p(x, c_m) = \sum_{dj \in KNN\ of\ X} SIM(X, d_j).\, y(d_j.\, c_m)$, (2-7)

dimana:

$p(x,c_m)$: probabilitas dokumen X menjadi anggota kategori c_m

$SIM(x,d_j)$: kemiripan antara dokumen X dengan dokumen latih d_j

$y(d_j.c_m)$: fungsi atribut dari sebuah kategori yang memenuhi

$$y(d_j.\, c_m) = \begin{cases} 1, d_j \in c_m \\ 0, d_j \notin c_m \end{cases} \quad (2\text{-}8)$$

Secara umum, langkah-langkah dari metode *K-Nearest Neighbor* (KNN) adalah sebagai berikut [IGA-11]:

1. Menentukan parameter K (jumlah tetangga paling dekat)
2. Menghitung kuadrat jarak euclidean (*query instance*) masing-masing objek terhadap data sampel yang diberikan
3. Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak euclidean terkecil
4. Mengumpulkan kategori klasifikasi *nearest neighbor*
5. Dengan menggunakan kategori *nearest neighbor* yang paling mayoritas maka dapat diprediksikan nilai *query instance* yang telah dihitung.

2.3.3 Improved K-Nearest Neighbor

Penentuan *k-values* yang tepat diperlukan agar didapatkan akurasi yang tinggi dalam proses kategorisasi dokumen uji. Algoritma *Improved k-Nearest Neighbor* melakukan modifikasi dalam penentuan *k-values*. Dimana penetapan *k-values* tetap dilakukan, hanya saja tiap-tiap kategori memiliki *k-values* yang

berbeda. Perbedaan *k-values* yang dimiliki pada setiap kategori disesuaikan dengan besar-kecilnya jumlah dokumen latih yang dimiliki kategori tersebut. Sehingga ketika *k-values* semakin tinggi, hasil kategori tidak terpengaruh pada kategori yang memiliki jumlah dokumen latih yang lebih besar.

Perhitungan penetapan *k-values* pada algoritma *Improved k-Nearest Neighbor* dilakukan dengan menggunakan persamaan (2-9), dengan terlebih dahulu mengurutkan secara menurun hasil perhitungan similaritas pada setiap kategori. Penelitian oleh Baoli,Shiwen,dan Qin untuk dokumen teks berbahasa Cina menunjukkan bahwa dengan algoritma *Improved k-Nearest Neighbor*, didapatkan kestabilan pada proses kategorisasi dengan berapapun variasi *k-values* [BSQ-03].

Selanjutnya pada algoritma *Improved k-Nearest Neighbor*, *k-values* yang baru disebut dengan n. Persamaan (2-9) menjelaskan mengenai proporsi penetapan *k-values* (n) pada setiap kategori [BSQ-03].

$$n = \left[\frac{k * N(c_m)}{\text{maks}\{N(c_m)|j=1.....N_c\}}\right], \qquad (2\text{-}9)$$

dimana:

n : *k-values* baru

k : *k-values* yang ditetapkan

N(C$_m$) : Jumlah dokumen latih di kategori/kategori m

Maks{N(C$_m$)|j=1...N$_c$}: jumlah dokumen latih terbanyak pada semua kategori

Sejumlah n dokumen yang dipilih pada tiap kategori adalah top n dokumen atau dokumen teratas yaitu dokumen yang mempunyai similaritas paling besar di setiap kategorinya.

Dalam menentukan kategori untuk dokumen uji menggunakan algoritma *Improved k-Nearest Neighbor*, maka dilakukan perbandingan similaritas pada setiap kategori. Perbandingan ini berdasarkan proporsi antara dua persamaan di bawah ini. Persamaan (2-10) menyatakan penjumlahan nilai similaritas sejumlah top n tetangga yang termasuk dalam suatu kategori [BSQ-03].

$$\sum_{d_j \in top\ n\ k\ NN(c_m)} sim(x, d_j) y(d_j . c_m). \qquad (2\text{-}10)$$

Persamaan (2.11) menyatakan penjumlahan nilai similaritas sejumlah top n tetangga pada training set.

$$\sum_{d_j \in top\ n\ k\ NN(c_m)} sim(x, d_j). \quad (2\text{-}11)$$

Perhitungan yang dilakukan dengan persamaan (2-12) menyatakan nilai maksimum perbandingan antara kemiripan dokumen X dengan dokumen latih d_j sejumlah top n tetangga pada suatu kategori dengan kemiripan dokumen X dengan dokumen latih d_j sejumlah top n tetangga pada *training set*.

$$p(x, c_m) = argmax_m \frac{\sum_{d_j \in top\ n\ k\ NN(c_m)} sim(x,d_j) y(d_j.c_m)}{\sum_{d_j \in top\ n\ k\ NN(c_m)} sim(x,d_j)}, \quad (2\text{-}12)$$

dimana:

$p(x,c_m)$: 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 yang memenuhi persamaan (2-8)

Nantinya dokumen uji X masuk ke dalam kategori yang memiliki nilai $P(x,c_m)$ paling besar.

Proses dalam mengkategorikan dokumen uji X menggunakan algoritma *Improved K-Nearest Neighbor* adalah sebagai berikut [BSQ-03]:

1. Melakukan tahapan *preprosesing* sehingga didapatkan representasi dari dokumen uji X dan semua dokumen latih
2. Setelah terbentuk vektor, hitung nilai *cosine similarity* dan dokumen uji X dengan semua dokumen latih
3. Selanjutnya mengurutkan hasil dari perhitungan nilai *cosine similarity* secara menurun. Nilai yang lebih tinggi menunjukkan bahwa di antara dokumen uji dan dokumen latih tersebut memiliki kemiripan
4. Mengelompokkan hasil dari perhitungan nilai *cosine similarity* berdasarkan kategorinya
5. Menentukan *k-values* kemudian melakukan perhitungan penetapan *k-values* baru (n) pada masing-masing kategori c_m menggunakan persamaan (2-9). Pemilihan n dokumen pada setiap kategori berdasarkan dokumen latih yang memiliki similaritas terbesar dengan dokumen uji (top n tetangga)

6. Setelah didapatkan nilai n yang menyatakan sebagai top tetangga dari langkah 5, maka langkah selanjutnya adalah menentukan kategori dokumen uji X berdasarkan hasil perhitungan menggunakan persamaan (2-12).
7. Berdasarkan perhitungan pada persamaan (2-12), maka dokumen X akan dikategorikan ke dalam kategori yang memiliki $P(x,c_m)$ terbesar.

2.4 Evaluasi

Evaluasi merupakan kegiatan yang membandingkan antara hasil implementasi dengan kriteria dan standar yang telah ditetapkan untuk melihat keberhasilannya. Dari evaluasi kemudian akan tersedia informasi mengenai sejauh mana suatu kegiatan tertentu telah dicapai sehingga bisa diketahui bila terdapat selisih antara standar yang telah ditetapkan dengan hasil yang bisa dicapai. Evaluasi yang digunakan pada penelitian ini adalah dengan menghitung *precision*, *recall*, dan *F-measure* dari hasil ringkasan yang dihasilkan oleh aplikasi.

2.4.1 Precision, Recall, dan F-measure

Sistem temu kembali informasi mengembalikan sekumpulan dokumen sebagai jawaban dari *query* pengguna. Terdapat dua kategori dokumen yang dihasilkan oleh sistem temu kembali informasi terkait pemrosesan *query*, yaitu *relevant documents* (dokumen yang relevan dengan *query*) dan *retrieved documents* (dokumen yang diterima pengguna) [RSY-10]. Ukuran umum yang digunakan untuk mengukur kualitas dari data retrieval adalah kombinasi *precision* dan *recall*.

Precision mengevaluasi kemampuan sistem temu kembali informasi untuk menemukan kembali data *top-ranked* yang paling relevan, dan didefinisikan sebagai persentase data yang dikembalikan yang benar-benar relevan terhadap query pengguna. *Precision* merupakan proporsi dari suatu set yang diperoleh yang relevan. *Precision* dapat dirumuskan persamaan 2.13.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} . \quad (2\text{-}13)$$

Relevant adalah jumlah dokumen yang relevan. *Retrieved* adalah jumlah dokumen yang dikembalikan atau diperoleh oleh/dari sistem kepada pengguna. *Recall* mengevaluasi kemampuan sistem temu kembali informasi untuk menemukan semua item yang relevan dari dalam koleksi data dan didefinisikan sebagai persentase data yang relevan terhadap query pengguna dan yang diterima. *Recall* merupakan proporsi dari semua hasil yang relevan di koleksi termasuk hasil yang diperoleh atau dikembalikan. *Recall* dapat dirumuskan menjadi persamaan 2.14.

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} . \quad (2\text{-}14)$$

Agar dapat lebih mudah dimengerti pengertian precision dan recall dapat di jabarkan pada tabel 2.1.

Tabel 2. 1 Tabel Kontingensi

	***Actual Class* (*expectation*)**		
***Predicted Class* (*Observation*)**		+	-
	+	TP	FP
	-	FN	TN

Sumber: [PDM-11]

$$precision = \frac{TP}{(TP + FP)} . \quad (2\text{-}15)$$

$$recall = \frac{TP}{(TP + FN)} . \quad (2\text{-}16)$$

Dengan menjabarkan tabel 2.1 di atas maka kita bisa mendapatkan persamaan (2.15) dan (2.16) untuk mendapatkan nilai *precision* dan *recall.* Dengan TP adalah *true positive* yaitu jumlah dokumen yang di hasilkan aplikasi sesuai dengan jumlah dokumen yang diberi oleh pakar. FP adalah *false positive* yaitu jumlah dokumen yang bagi pakar dianggap salah akan tetapi oleh aplikasi

dianggap benar (hasil yang tidak diinginkan). FN adalah *false negative* yaitu jumlah dokumen yang bagi pakar dianggap benar akan tetapi oleh aplikasi dianggap salah (*missing result*) [PDM-11].

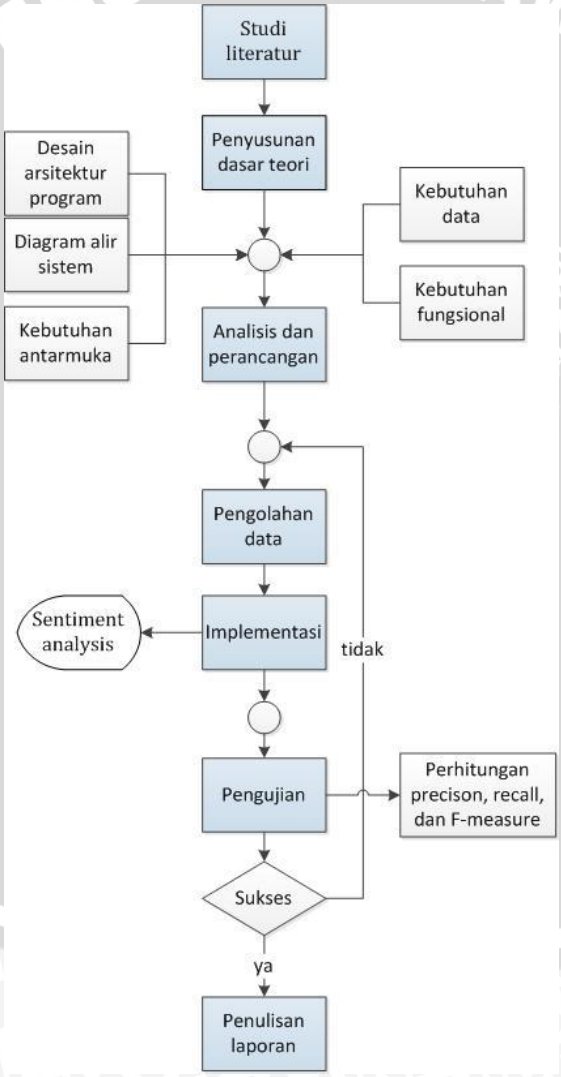
Kombinasi *precision* dan *recall* biasa dikombinasikan sebagai *harmonic mean*, biasa disebut *F-measure* yang mana dapat di formulasikan seperti persamaan (2.17).

$$F - measure = 2 * \frac{precision * recall}{precision + recall}. \quad (2\text{-}17)$$

F-measure biasa digunakan pada bidang sistem temu kembali informasi untuk mengukur klasifikasi pencarian dokumen dan performa *query classification*. Pada penelitian terdahulu *F-measure* lebih difokuskan untuk menghitung nilai, namun seiring dengan perkembangan mesin pencari dengan skala besar, kini *F-measure* lebih menekankan pada kinerja *precision* dan *recall* itu sendiri. Sehingga lebih bisa dilihat pada aplikasi secara keseluruhan [GHR-05].

BAB III
METODE PENELITIAN DAN PERANCANGAN

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam pengerjaan skripsi, yaitu studi literatur, penyusunan dasar teori, analisa dan perancangan, implementasi, analisis dan pengujian dari aplikasi perangkat lunak yang akan dibuat, hingga penulisan laporan. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya. Gambar 3.1 menunjukkan desain penelitian secara umum.



Gambar 3. 1 Desain Penelitian

Sumber: Perancangan

3.1 Studi Literatur

Studi literatur mempelajari mengenai penjelasan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut diperoleh dari buku, jurnal, *e-book*, penelitian sebelumnya, dan dokumentasi project. Referensi yang digunakan untuk mendukung penulisan skripsi ini meliputi:

1. Pemahaman tentang struktur dokumen penelitian
2. Pemahaman tentang *text mining*
3. Pemahaman tentang *sentiment analysis*
4. Pemahaman tentang metode *K-Nearest Neighbor* (KNN)
5. Pemahaman tentang metode *Improved K-Nearest Neighbor*
6. Bahasa pemrograman PHP
7. Basis data MySQL

3.2 Penyusunan Dasar Teori

Penyusunan dasar teori dilakukan setelah mendapatkan referensi yang tepat untuk mendukung penulisan penelitian ini. Teori-teori pendukung tersebut meliputi:

1. *Text Mining*

   Meliputi tahap-tahap pembersihan dokumen, *parsing, tokenizing, filtering/stopword removal*, dan *stemming.* untuk menghasilkan term yang nantinya akan di proses oleh program.

2. Analisis Sentimen (*Sentiment Analysis*)

   Meliputi tahap-tahap *term weighting* dan *classification* untuk menghasilkan analisis sentimen sesuai dengan metode yang telah dipilih oleh peneliti.

3.3 Metode Pengumpulan Data

Data yang dibutuhkan untuk penelitian ini adalah data *tweets* berbahasa Indonesia sesuai EYD yang berasal dari mikroblog Twitter dengan tema penyedia layanan GSM. Metode penentuan kelas sentimen dari data *tweets* dilakukan melalui wawancara dengan pakar bahasa Indonesia untuk mengetahui klasifikasi sentimen *tweets* secara tepat.

3.4 Analisis dan Perancangan

Analisis kebutuhan bertujuan untuk menganalisis dan mendapatkan kebutuhan – kebutuhan yang diperlukan dalam perancangan analisis sentimen *Twitter* berbahasa Indonesia menggunakan metode *improved k-nearest neighbor*.

Metode analisa yang digunakan adalah analisis prosedural dengan menggunakan bahasa pemodelan prosedural. Pemrograman berbasis prosedur merupakan teknik pemrograman yang dikembangkan berdasarkan algoritma untuk memecahkan suatu masalah. Algoritma merupakan cara-cara yang ditempuh dalam memanipulasi data sehingga masalah yang dihadapi bisa dipecahkan.

Kebutuhan yang digunakan dalam pembuatan analisis sentimen:

1. Kebutuhan *hardware*, meliputi:
   a. Laptop atau PC
2. Kebutuhan *software*, meliputi:
   a. Mac OS X 10.6.8 (10K540) sebagai sistem operasi
   b. MySQL sebagai *server Database Management System*
   c. Aptana Studio 3 versi 3.3.2.201302081546-08022013154827 sebagai aplikasi untuk implementasi analisis sentimen menggunakan bahasa pemrograman PHP.
3. Data yang dibutuhkan yaitu dokumen penelitian meliputi *tweets* dari domain http://www.twitter.com/ yang bertema penyedia layanan GSM dengan kata kunci pencarian“sinyal indosat”.

3.4.1 Kebutuhan Antar Muka

Kebutuhan-kebutuhan untuk pengembangan perangkat lunak ini sebagai berikut:

1. Program yang akan dibangun harus mempunyai tampilan yang familiar bagi pemakai.
2. Program yang akan dibangun harus mempunyai tampilan yang memungkinkan user untuk menginput nilai k.
3. Program yang akan dibangun harus mampu menampilkan hasil analisis sentimen yang telah diproses sebelumnya.

3.4.2 Kebutuhan Data

Data yang diolah oleh perangkat lunak ini adalah:

1. Data kata dasar bahasa indonesia yang berfungsi pada saat proses *stemming.*
2. Data berupa *tweet* yang menggunakan bahasa Indonesia yang sesuai dengan EYD.
3. Data hasil analisis sentimen yang akan disimpan di dalam basis data.

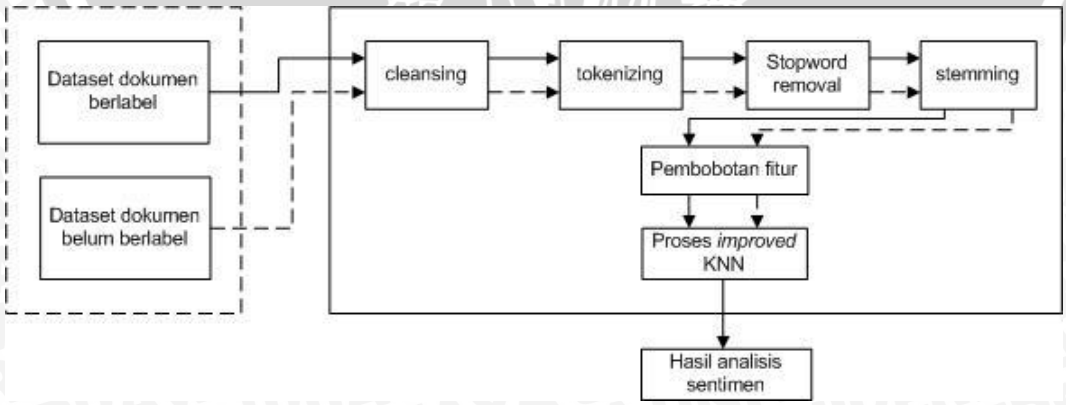
3.4.3 Kebutuhan Fungsional

Fungsi-fungsi yang dimiliki oleh perangkat lunak ini adalah:

1. Perangkat lunak harus mampu melakukan proses *preprocessing*, yakni berupa *tokenizing*, penghilangan kata *stopwords* dan *stemming*.
2. Perangkat lunak harus mampu melakukan proses pembobotan berdasarkan metode yang akan digunakan.
3. Perangkat lunak harus mampu menghasilkan hasil analisis sentimen *tweets* berdasarkan klasifikasi dari proses yang telah dilakukan.

3.4.4 Arsitektur Sistem

Rancangan arsitektur sistem menggambarkan kerangka dasar dari sistem yang akan dikembangkan. Tahapan proses analisis sentimen pada *Twitter* berbahasa Indonesia dapat dilihat lebih jelas pada gambar 3.2. Garis putus-putus menunjukkan proses klasifikasi dokumen uji, sedangkan garis tebal menunjukkan proses pembentukan *classifier*.

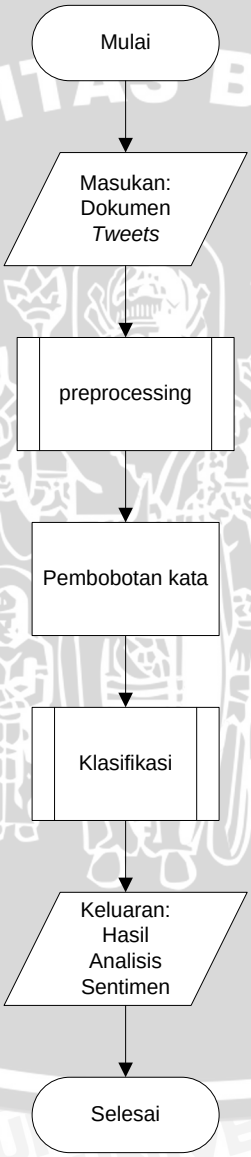


Gambar 3. 2 Pereancangan Arsitektur Sistem

Sumber: Perancangan

3.4.5 Diagram Alir

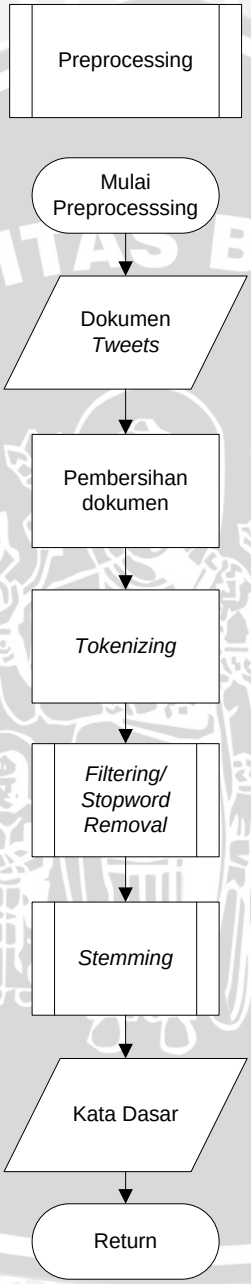
Diagram alir menggunakan notasi-notasi untuk menggambarkan arus data yang membantu dalam proses memahami jalannya aplikasi. Secara umum sistem ini dimulai dengan masukan berupa dokumen *tweets*. Dokumen *tweets* akan melalui tahapan *preprocessing*, yang dilanjutkan dengan pembobotan *term*, proses klasifikasi sentimen *tweets*, dan menghasilkan keluaran hasil analisis sentimen. Gambar 3.3 menunjukkan diagram alir sistem.



Gambar 3. 3 Diagram Alir Sistem

Sumber: Perancangan

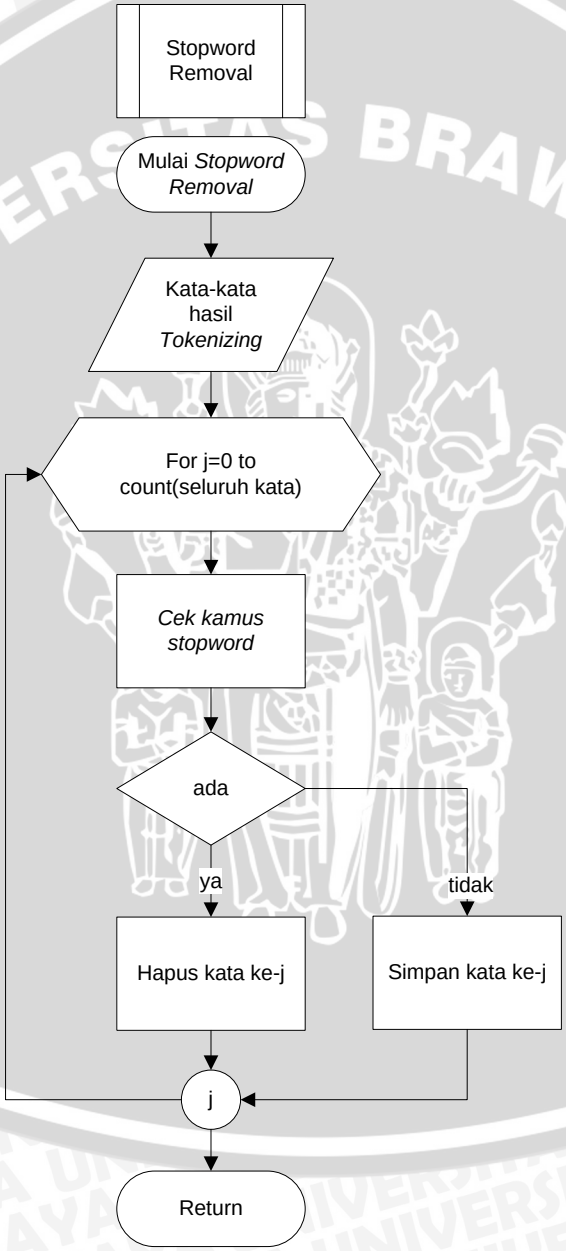
Proses *preprocessing* terdiri dari pembersihan dokumen yaitu penghilangan simbol dan karakter selain huruf alfabet, *tokenizing* yaitu pemecahan dokumen menjadi kata atau *term*, *stopword removal*, dan *stemming*. Gambar 3.4 menunjukkan diagram alir *preprocessing*.



Gambar 3. 4 Diagram Alir *Preprocessing*

Sumber: Perancangan

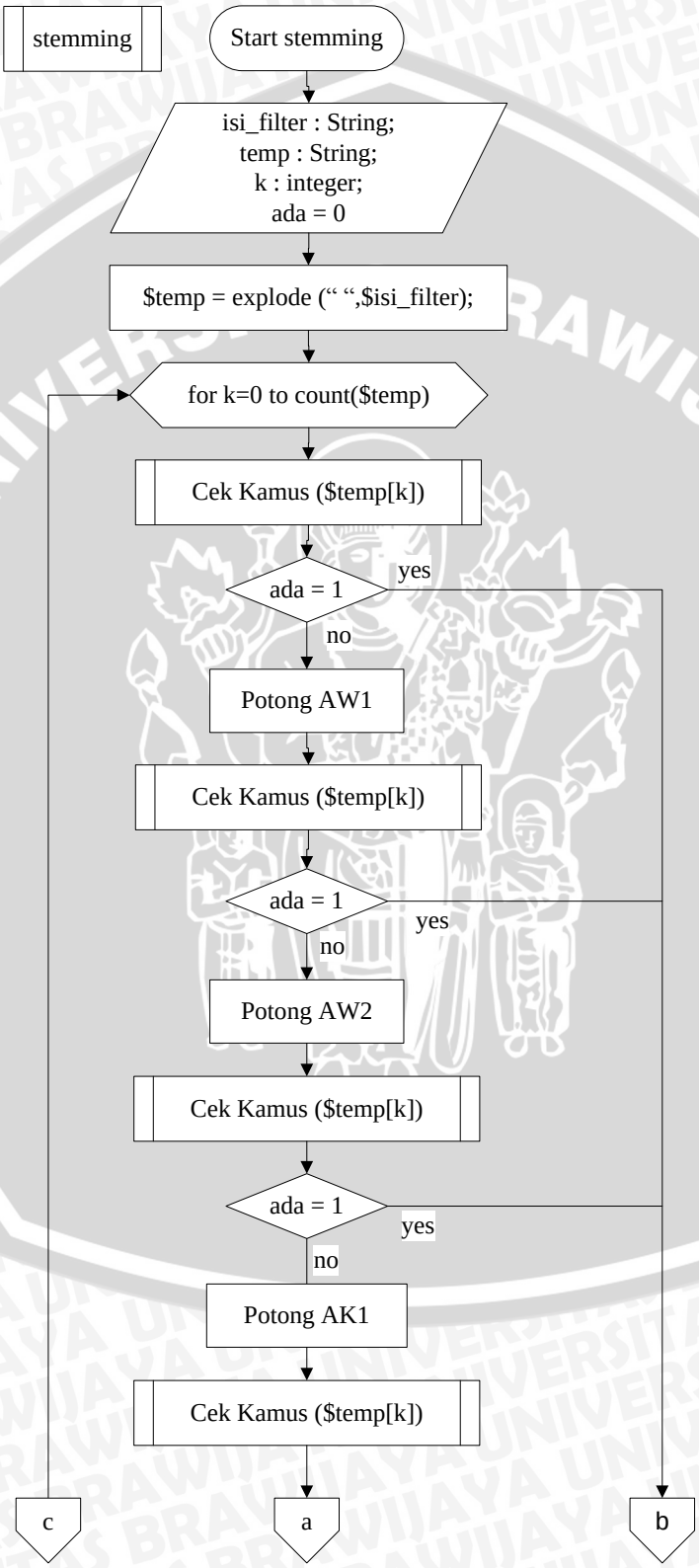
Stopword removal kemudian dilakukan untuk menghilangkan kata-kata yang tidak relevan yang terdapat pada dokumen yang telah di-*tokenize* sebelumnya. Kata-kata pada dokumen dicocokkan dengan kamus *stopword*, jika kata tersebut ada, maka dihilangkan dari dokumen, sedangkan jika tidak ada, maka kata tersebut disimpan. Gambar 3.5 menunjukkan diagram alir *stopword removal*.

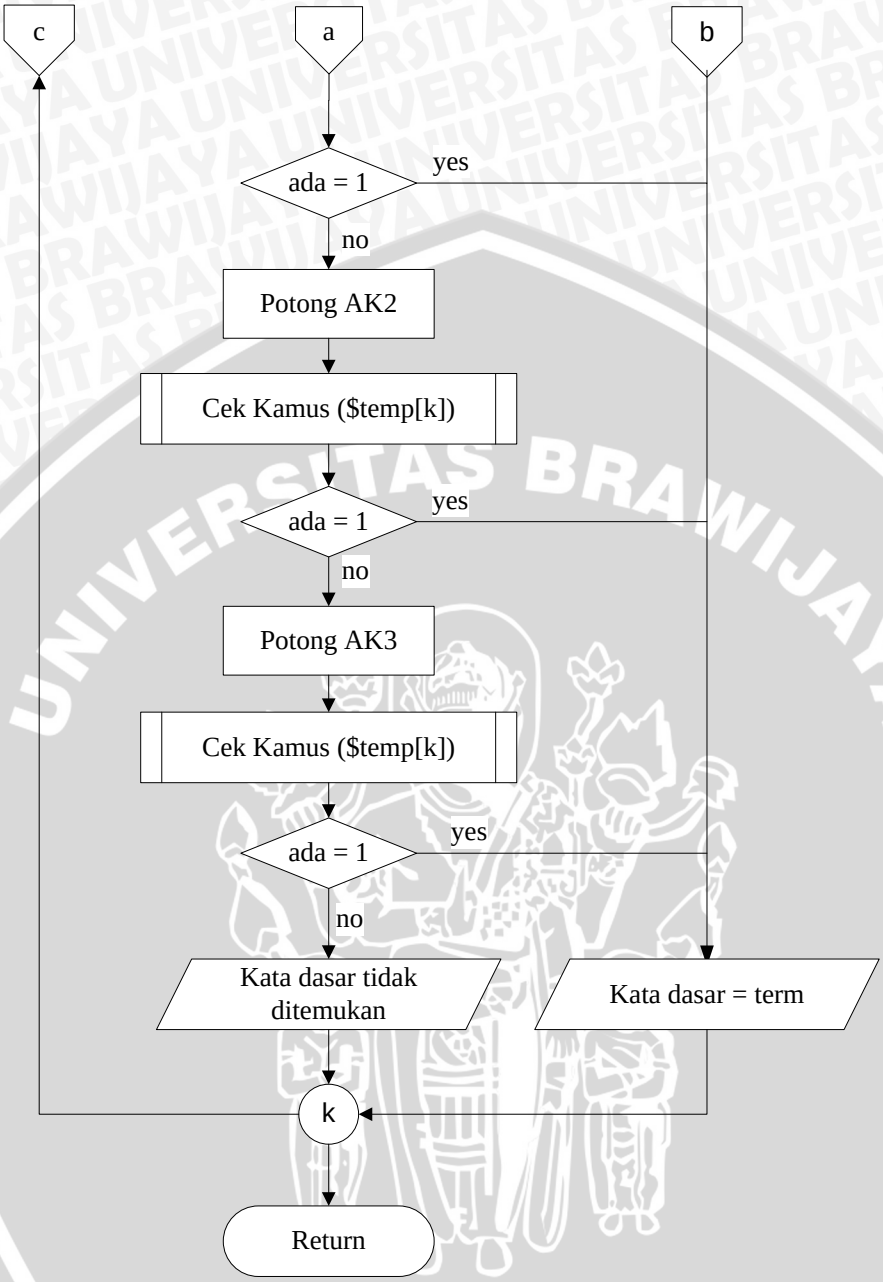


Gambar 3. 5 Diagram Alir *Filtering/Stopword Removal*

Sumber: Perancangan

Stemming arifin-setiono dilakukan untuk mendapatkan kata dasar dari sebuah *term*. Proses ini digunakan untuk mencari kata dasar dari suatu *term*. Gambar 3.6 menunjukkan diagram alir *stemming* arifin-setiono.

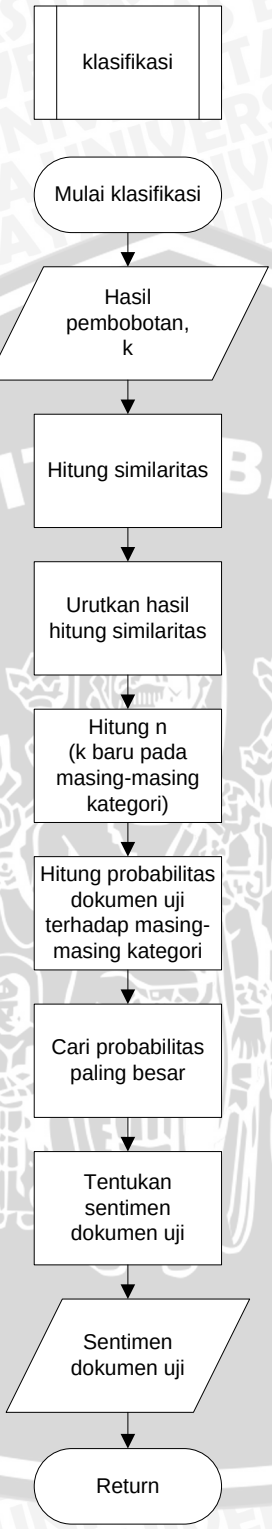




Gambar 3. 6 Diagram Alir *Stemming* Arifin-Setiono

Sumber: [DGT-09]

Term-term yang telah melalui proses *stemming* kemudian dihitung bobotnya dengan menggunakan TF-IDF. Hasil dari perhitungan bobot kemudian akan disimpan untuk proses selanjutnya yaitu klasifikasi dengan menggunakan *Improved K-Nearest Neighbor*. Gambar 3.7 menunjukkan diagram alir *Improved K-Nearest Neighbor*.



Gambar 3. 7 Diagram Alir *Improved K-Nearest Neighbor*

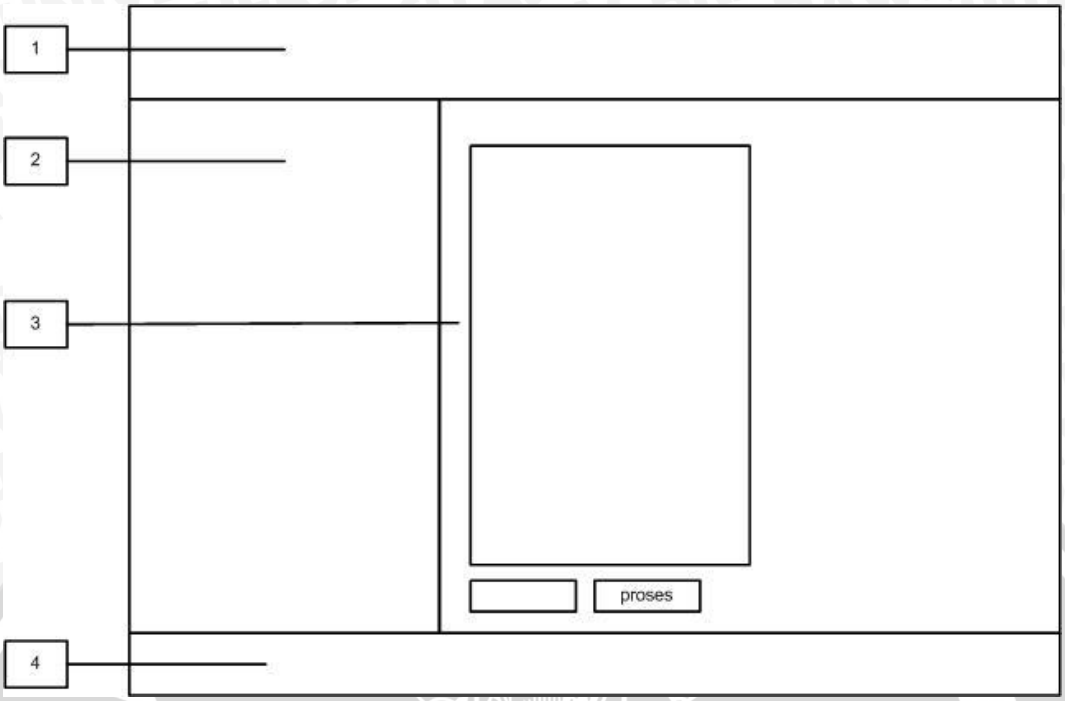
Sumber: Perancangan

Proses klasifikasi menerima masukan berupa hasil dari penghitungan bobot dan nilai k. Bobot kata digunakan untuk menghitung similaritas antara dokumen uji terhadap semua dokumen latih dengan menggunakan *cosine similarity* (cossim) sesuai dengan rumus (2-5). Setelah didapatkan semua nilai similaritas dokumen uji terhadap dokumen latih, nilai similaritas tersebut diurutkan dari yang terbesar hingga yang terkecil. Proses selanjutnya adalah menghitung nilai n (k baru) pada masing-masing kategori dengan rumus (2-9) berdasarkan k yang sudah ditentukan sebelumnya. Setelah didapatkan nilai n yang menyatakan n tetangga terdekat, langkah selanjutnya adalah menentukan kelas sentimen dokumen uji dengan menghitung probabilitas dokumen uji terhadap masing-masing kategori sesuai persamaan (2-12). Berdasarkan perhitungan probabilitas tersebut, dokumen uji kemudian dikategorikan ke dalam kelas sentimen yang memiliki probabilitas paling besar.

Setelah semua proses mulai preprocessing hingga klasifikasi dengan *Improved K-Nearest Neighbor* dilalui, hasil dari analisis sentimen kemudian ditampilkan kembali melalui antarmuka.

3.4.6 Desain Antar Muka

Perancangan desain ini bertujuan agar pengguna dapat nyaman menggunakan aplikasi yang akan dirancang. Desain antar muka sistem ini terdiri dari empat komponen utama. Isi pada antar muka sistem adalah *option form* untuk menerima *input* pengguna berupa nilai k, dan satu tombol proses untuk mengirim masukan dari pengguna ke sistem. Perancangan desain antar muka sistem amalisis sentimen ini dapat dilihat pada gambar 3.8.



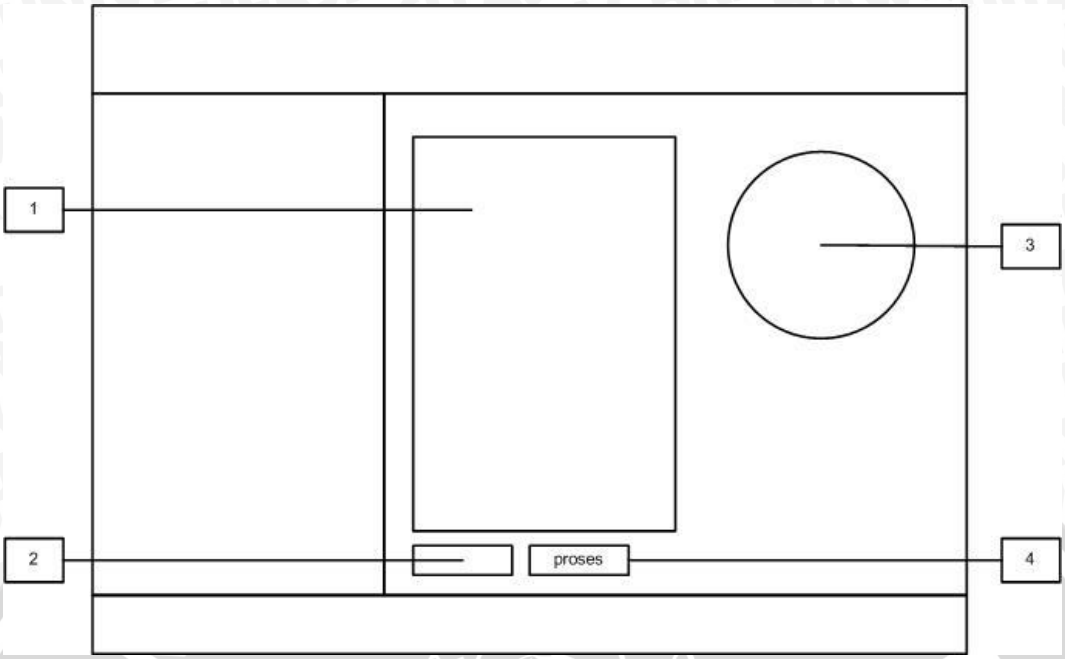
Gambar 3. 8 Perancangan Desain Antar Muka Sistem

Sumber: Perancangan

Keterangan gambar 3.8:

1. *Header*
2. Menu
3. Isi
4. *Footer*

Pada halaman hasil, akan ditampilkan sampel dokumen yang diuji dan grafik hasil sentimen analisis. Perancangan desain antarmuka untuk halaman hasil dapat dilihat pada gambar 3.9.



Gambar 3. 9 Perancangan Desain Antar Muka Halaman Hasil

Sumber: Perancangan

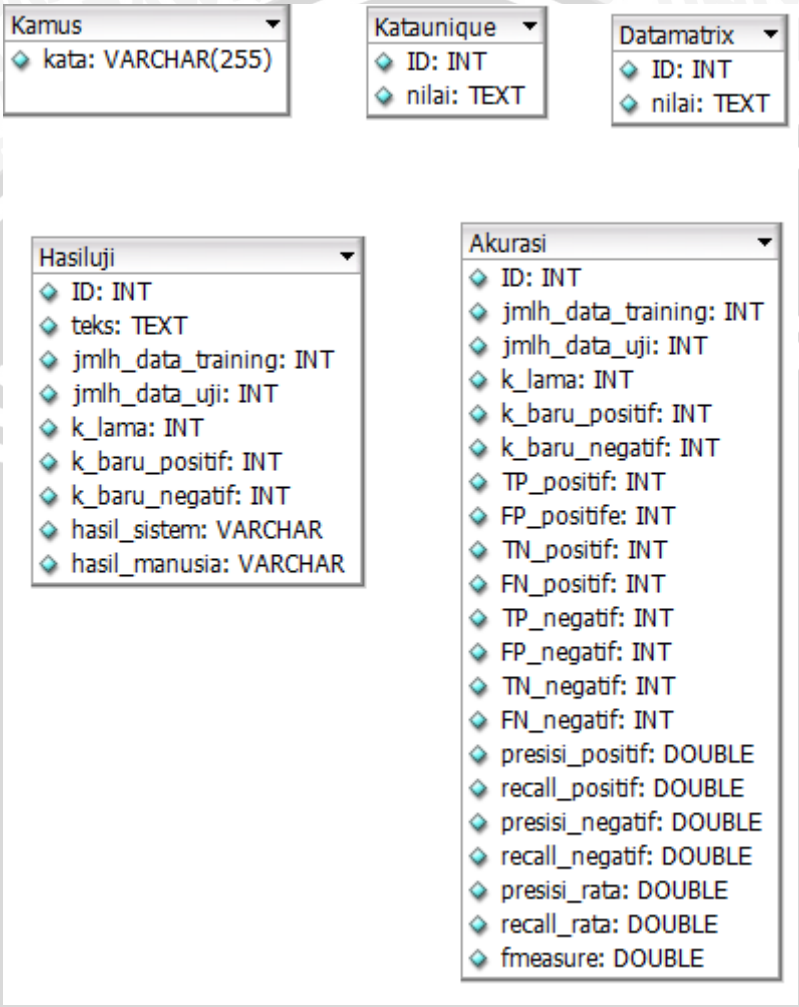
Keterangan gambar 3.9:

1. Data uji
2. *Option form* untuk masukan k
3. Grafik hasil sentimen analisis
4. Tombol proses

3.4.7 Perancangan Basis Data

Perancangan basis data merupakan perancangan manajemen data yang akan digunakan. Manajemen data termasuk basis data yang mengandung data relevan untuk berbagai situasi dan diatur oleh software yang disebut Database Management System (DBMS). Sistem ini menggunakan MySQL sebagai DBMSnya. Perancangan basis data sistem ini menggunakan lima tabel yaitu tabel hasiluji, tabel akurasi, tabel datamatrix, tabel kataunique, dan tabel kamus. Tabel hasiluji digunakan untuk menyimpan hasil pengujian yang telah dilakukan. Tabel akurasi berfungsi untuk menyimpan nilai-nilai yang dibutuhkan dalam proses penghitungan akurasi dan nilai akurasinya. Tabel kataunique merupakan tabel yang berfungsi untuk menyimpan hasil preprocessing berupa kata unik atau keyword. Tabel datamatrix berfungsi untuk menyimpan matrix TF setelah

dilakukan preprocessing pada dataset. Tabel kamus berfungsi untuk menyimpan kumpulan kata dasar yang digunakan pada proses stemming. Tabel pada sistem ini tidak berelasi dengan tabel yang lainnya. Tabel perancangan database ditunjukkan pada gambar 3.10.



Gambar 3. 10 Perancangan Basis Data

Sumber: Perancangan

3.4.8 Manualisasi Analisis Sentimen

Penghitungan manual berfungsi untuk memberikan gambaran umum perancangan sistem yang akan dibangun. Contoh manualisasi setiap proses dari analisis sentiment dengan *Improved K-Nearest Neighbor* adalah sebagai berikut:

1. Input dokumen *tweets*

   Dokumen *tweets* berlabel akan dimasukkan ke dalam sistem sebagai proses pembelajaran. Pada manualisasi ini, dokumen 1 sampai dengan 5 adalah dokumen latih, sedangkan dokumen x adalah dokumen uji. Tabel 3.1 merupakan contoh dokumen latih dan dokumen uji.

Tabel 3. 1 Contoh Dokumen

ID	Tweets	Sentimen
1	@leo92 indosat udah murah sinyal bagus pula. Sip banget!	positif
2	untung pake indosat, murah meriah keren kece \:D/	positif
3	indosat sialan.	negatif
4	hash indosat gangguan melulu! !@#$%	negatif
5	aduh sinyalnya indosat jelek banget -_____-	negatif
x	indosat murah tapi sinyal jelek gangguan melulu	?

Sumber: Perancangan

2. *Preprocessing*

   Dokumen *tweets* yang telah dimasukkan kemudian melalui beberapa tahap pada proses *preprocessing*, yaitu Pembersihan dokumen, *Tokenizing*, *Filtering/Stopword Removal*, dan *Stemming*.

   a. *Pembersihan dokumen*

      Pembersihan dokumen merupakan tahap menghilangkan tanda baca, angka, dan karakter selain *alphabet*. Tabel manualisasi pembersihan dokumen dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Tabel Manualisasi Pembersihan dokumen

ID	Tweets	Sentimen
1	leo indosat udah murah sinyal bagus pula Sip banget	Positif
2	untung pake indosat murah meriah keren kece	Positif
3	indosat sialan	Negatif
4	hash indosat gangguan melulu	Negatif
5	aduh sinyalnya indosat jelek banget	Negatif
x	indosat murah tapi sinyal jelek gangguan melulu	?

Sumber: Perancangan

b. *Tokenizing*

Tokeenizing merupakan tahap memecah kalimat menjadi kata. Pada sistem in, kalimat akan dipecah berdasarkan spasi. Tabel manualisasi *tokenizing* dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Tabel Manualisasi *Tokenizing*

ID	Tweets									Sentimen
1	leo	indosat	udah	murah	sinyal	bagus	pula	sip	banget	positif
2	untung	pake	indosat	murah	meriah	keren	kece			positif
3	indosat				sialan					negatif
4	hash		indosat		gangguan		melulu			negatif
5	aduh		sinyalnya		indosat		jelek		banget	negatif
x	indosat	murah	tapi	sinyal	jelek	gangguan	melulu			?

Sumber: Perancangan

c. *Filtering*/*Stopword Removal*

Filtering/Stopword Removal merupakan tahap menghilangkan kata yang tidak penting berdasarkan kamus *stopword*. Tabel manualisasi *filtering/stopword removal* dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Tabel Manualisasi Filtering/Stopword Removal

ID	Tweets							Sentimen
1	leo	indosat	udah	murah	sinyal	bagus	sip	positif
2	untung	pake	indosat	murah	meriah	keren	kece	positif
3	indosat				sialan			negatif
4	hash		indosat		gangguan		melulu	negatif
5	aduh		sinyalnya		indosat		jelek	negatif
x	indosat	murah	sinyal	jelek	gangguan	melulu		?

Sumber: Perancangan

d. *Stemming*

Stemming merupakan tahap mengubah kata menjadi kata dasar. Kata-kata pada dokumen akan dicocokkan dengan kamus Bahasa Indonesia, dan diubah sesuai dengan aturan pada *Stemming* Arifin-Setiono. Apabila setelah dicek di kamus dan diubah sesuai aturan kata tidak ditemukan sebagai kata dasar, kata tersebut dikembalikan ke bentuk asalnya dan dihitung sebagai kata dasar baru. Tabel manualisasi *stemming* dapat dilihat pada Tabel 3.5.

Tabel 3. 5 Tabel Manualisasi *Stemming*

ID	Tweets							Sentimen
1	leo	indosat	udah	murah	sinyal	bagus	sip	positif
2	untung	pake	indosat	murah	meriah	keren	kece	positif
3	indosat				sialan			negatif
4	hash		indosat		gangguan		melulu	negatif
5	aduh	sinyalnya		indosat		jelek		negatif
x	indosat	murah	sinyal	jelek	gangguan	melulu		?

Sumber: Perancangan

3. Pembobotan Kata

Setelah dilakukan proses preprocessing, selanjutnya dibuat tabel informasi dokumen yang berisi frekuensi term (TF), frekuensi dokumen (DF), dan IDF dari masing-masing term. Kemudian dicari nilai TF*IDF dari masing-masing term. Nilai IDF yang digunakan adalah nilai IDF yang didapatkan setelah proses pelatihan sistem. Tabel 3.6 memuat manualisasi dari pembobotan kata.

Tabel 3. 6 Tabel Manualisasi Pembobotan Kata

TERM VECTOR MODEL BASED ON wi=Tfi*IDFi

t1: @leo92 indosat udah murah sinyal bagus pula. Sip banget!
t2: untung pake indosat, murah meriah keren kece \:D/
t3: indosat sialan.
t4: hash indosat gangguan melulu! !@#$%
t5: aduh sinyalnya indosat jelek banget -_____-
x: indosat murah tapi sinyal jelek gangguan melulu
D=5; IDF=log(D/DF);

		counts, Tfi						DF	IDF	weight, TF*IDF					
No	Term	t1	t2	t3	t4	t5	x			t1	t2	t3	t4	t5	x
1	leo	1	0	0	0	0	0	1	0,699	0,699	0,000	0,000	0,000	0,000	0,000
2	indosat	1	1	1	1	1	1	5	0,000	0,000	0,000	0,000	0,000	0,000	0,000
3	udah	1	0	0	0	0	0	1	0,699	0,699	0,000	0,000	0,000	0,000	0,000
4	murah	1	1	0	0	0	0	2	0,398	0,398	0,398	0,000	0,000	0,000	0,398
5	sinyal	1	0	0	0	0	0	1	0,699	0,699	0,000	0,000	0,000	0,000	0,699
6	bagus	1	0	0	0	0	0	1	0,699	0,699	0,000	0,000	0,000	0,000	0,000
7	sip	1	0	0	0	0	0	1	0,699	0,699	0,000	0,000	0,000	0,000	0,000
8	untung	0	1	0	0	0	0	1	0,699	0,000	0,699	0,000	0,000	0,000	0,000
9	pake	0	1	0	0	0	0	1	0,699	0,000	0,699	0,000	0,000	0,000	0,000
10	meriah	0	1	0	0	0	0	1	0,699	0,000	0,699	0,000	0,000	0,000	0,000

11	keren	0	1	0	0	0	0	1	0,699	0,000	0,699	0,000	0,000	0,000	0,000
12	kece	0	1	0	0	0	0	1	0,699	0,000	0,699	0,000	0,000	0,000	0,000
13	sialan	0	0	1	0	0	1	1	0,699	0,000	0,000	0,699	0,000	0,000	0,699
14	hash	0	0	0	1	0	0	1	0,699	0,000	0,000	0,000	0,699	0,000	0,000
15	gangguan	0	0	0	1	0	0	1	0,699	0,000	0,000	0,000	0,699	0,000	0,699
16	melulu	0	0	0	1	0	0	1	0,699	0,000	0,000	0,000	0,699	0,000	0,699
17	aduh	0	0	0	0	1	0	1	0,699	0,000	0,000	0,000	0,000	0,699	0,000
18	sinyalnya	0	0	0	0	1	0	1	0,699	0,000	0,000	0,000	0,000	0,699	0,000
19	jelek	0	0	0	0	1	0	1	0,699	0,000	0,000	0,000	0,000	0,699	0,699

Sumber: Perancangan

4. Analisis Sentimen

a. *Cosine Similarity*

Setelah didapatkan nilai TF*IDF, selanjutnya dihitung nilai similaritas dokumen uji dengan dokumen latih. Contoh perhitungan similaritas adalah seperti di bawah ini:

$$CosSim(X, t_1) = \frac{\sum_{i=1}^{m} x_i . \mathrm{t}_{1i}}{\sqrt{(\sum_{i=1}^{m} x_i)^2} . \sqrt{(\sum_{i=1}^{m} \mathrm{t}_{1i})^2}} = \frac{0.6469}{1.613 * 1.613} = 0.248703$$

Perhitungan ini diulang dan dilakukan pada semua dokumen, sehingga didapatkan data seperti pada Tabel 3.7.

Tabel 3. 7 Tabel Manualisasi *Cosine Similarity*

Tweets	Sim(x,d)	Sentimen
t1	0,248703	Positif
t2	0,060879	Positif
t3	0,433387	Negatif
t4	0,500432	Negatif
t5	0,250216	Negatif

Sumber: Perancangan

Hasil perhitungan *Cosine Similarity* kemudian diurutkan secara *descending*. Hasil pengurutan *Cosine* Similarity dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Tabel Manualisasi Pengurutan *Cosine Similarity*

Tweets	Sim(x,d)	sentimen
t4	0,500432	negatif
t3	0,433387	negatif
t5	0,250216	negatif
t1	0,248703	positif
t2	0,060879	positif

Sumber: Perancangan

b. Klasifikasi

Setelah didapatkan hasil pengurutan *cosine similarity*, sistem akan menghitung nilai k baru (n) yang akan digunakan untuk proses klasifikasi.. Misalkan nilai k yang diminta pengguna adalah k=2. Maka sistem akan menghitung nilai n (*k-values* baru) untuk masing-masing kelas sentimen sesuai dengan rumus 2.9. Hasil manualisasi beberapa nilai k baru dapat dilihat pada Tabel 3.9. Contoh perhitungan manual nilai k baru sebagai berikut:

$$n(c_{\text{pos}}) = \frac{2 * 2}{3} = \frac{4}{3} = 1.33 = 1$$

$$n(c_{\text{neg}}) = \frac{2 * 3}{3} = \frac{6}{3} = 2$$

Tabel 3. 9 Tabel Manualisasi Perhitungan Nilai n

K	n (c_{pos})	n (c_{neg})
2	1	2
3	2	3
4	3	4

Sumber: Perancangan

Setelah didapatkan nilai n, proses selanjutnya adalah mengelompokkan dokumen uji dengan menghitung nilai probabilitas dokumen uji pada masing-masing kelas sentimen sesuai dengan rumus 2.12, dan dicari nilai $P(x,c_m)$ terbesar. Hasil perhitungan manual probabilitas pada masing-masing sentimen di beberapa k dapat dilihat pada Tabel 3.10. Contoh perhitungan manual perhitungan probabilitas adalah sebagai berikut:

$$P(\mathrm{x}, c_{\mathrm{pos}}) = \frac{0.500432 * 0}{0.500432} = \frac{0}{0.500432} = 0$$

$$P(\mathrm{x}, c_{\mathrm{neg}}) = \frac{(0.500432 * 1) + (0.433387 * 1)}{0.500432 + 0.433387} = \frac{0.933819}{0.933819} = 1$$

Tabel 3. 10 Tabel Manualisasi Perhitungan Probabilitas

K	P(x, c_{pos})	P(x, c_{neg})
2	0	1
3	0	1
4	0	0,826414

Sumber: Perancangan

Dari hasil proses perhitungan di atas, maka dapat disimpulkan dokumen uji x termasuk di dalam kelas sentimen negatif karena memiliki nilai probabilitas lebih besar pada sentimen negatif yaitu bernilai 1.

3.5 Implementasi

Implementasi aplikasi analisis sentimen ini dilakukan dengan mengacu pada perancangan sistem. Implementasi perangkat lunak dilakukan menggunakan bahasa pemrograman PHP dan database MySQL. Implementasi aplikasi ini meliputi:

- Pembuatan *user interface* berupa halaman-halaman web yang menerima input dari user.
- Melakukan proses *preprocessing* pada dokumen *tweets*.
- Melakukan proses pembobotan kata dengan TF-IDF.
- Menentukan sentimen *tweets* (sentimen positif atau sentimen negatif)
- Menampilkan hasil dari analisis sentimen.

3.6 Pengujian

Pengujian perangkat lunak pada skripsi ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang melandasinya. Pengujian yang dilakukan meliputi:

- Pengujian hasil program dengan cara membandingkan *output* program dengan *output* manual berdasarkan analisis pakar.
- Pengukuran tingkat akurasi program yang akan di hitung menurut *presisi*, *recall*, dan *F-measure*

Perancangan tabel pengujian *Precision*, *Recall*, dan *F-Measure* sistem dapat dilihat pada Tabel 3.11.

Tabel 3. 11 Perancangan Tabel Pengujian

k	$n(c_m)$		*Precision*	*Recall*	*F-Measure*
	$n(c_{positif})$	$n(c_{negatif})$			

Sumber: Perancangan

3.7 Penarikan Kesimpulan

Penarikan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran. Saran bertujuan untuk memperbaiki kesalahan-kesalahan yang terjadi, menyempurnakan penulisan, dan untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

BAB IV
IMPLEMENTASI

Bab ini membahas implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat pada bab III. Bab ini terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi algoritma pada program, implementasi antarmuka, dan implementasi metode.

4.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada bab III menjadi acuan untuk melakukan implementasi menjadi aplikasi yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi aplikasi diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

4.1.1 Spesifikasi Perangkat Keras

Pengembangan aplikasi analisis sentimen Twitter berbahasa Indonesia ini menggunakan sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 4.1.

Tabel 4. 1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core 2 Duo 2.26GHz
Memori(RAM)	2 GB
Harddisk	160.04 GB
VGA	Nvidia GeForce 9400M 256 MB

Sumber: Implementasi

4.1.2 Spesifikasi Perangkat Lunak

Pengembangan aplikasi analisis sentimen Twitter berbahasa Indonesia ini menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 4.2.

Tabel 4. 2 Spesifikasi Perangkat Lunak Komputer

Nama	Spesifikasi
Sistem Operasi	Mac OS X 10.6.8 (10K540)
Bahasa Pemrograman	HTML dan PHP 5.3.1
Tools pemrograman	Aptana Studio 3 versi 3.3.2.201302081546 - 08022013154827
Server localhost	XAMPP versi 1.7.3
DBMS	MySQL versi 5.1.44
Web Browser	Google Chrome versi 26.0.1410.65

Sumber: Implementasi

4.2 Batasan-batasan Implementasi

Batasan implementasi adalah batasan proses yang dapat dilakukan sistem sesuai dengan perancangan awal sistem. Batasan implementasi ditampilkan agar penelitian ini memiliki ruang lingkup yang jelas dalam mengimplementasikan sistem. Beberapa batasan dalam mengimplementasikan aplikasi analisis sentimen Twitter berbahasa Indonesia ini adalah sebagai berikut:

- Analisis sentimen *Twitter* berbahasa Indonesia dirancang dan dijalankan menggunakan aplikasi web.
- Metode penyelesaian masalah yang digunakan adalah *Improved k-Nearest Neighbor*.
- Dokumen yang digunakan sebagai data latih dan data uji adalah berasal dari mikroblog Twitter (www.twitter.com) pada tanggal 25 Maret 2013 sampai dengan 30 Maret 2013.
- Dokumen yang digunakan merupakan dokumen berbahasa Indonesia.
- Output yang dikeluarkan berupa hasil analisis sentimen yaitu positif atau negatif.
- Penentuan sentimen *tweets* berdasarkan pada frekuensi kemunculan kata, bukan pada struktur semantik.

4.3 Implementasi Algoritma

Aplikasi analisis sentimen Twitter berbahasa Indonesia ini terdiri dari beberapa proses utama yaitu *preprocessing*, pembobotan atau *weighting*, dan klasifikasi. Data berupa *tweets* akan diolah dengan *preprocessing* hingga didapatkan kata dasar unik yang disebut *keywords*, kemudian dihitung bobot dokumennya dengan menggunakan TF-IDF, dan diklasifikasi sentimennya dengan menggunakan *improved k-nearest neighbor*. Daftar fungsi pada aplikasi analisis sentimen Twitter berbahasa Indonesia ini dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Daftar Fungsi pada Sistem

No.	Proses	Fungsi	Keterangan
1.	Preproccesing	function cleansing()	Fungsi untuk menghilangkan tanda baca seperti koma, titik, petik dua, dan lain-lain.
		function casefolding()	Fungsi untuk mengubah dokumen tweets menjadi huruf kecil semua.
		function tokenizing()	Fungsi untuk memecah kalimat menjadi kata yang berdiri sendiri.
		function stopword_removal()	Fungsi untuk menghilangkan kata yang tidak penting seperti ‘yang’,’adalah’,’itu’, dan lain – lain.
		function stemming()	Fungsi untuk memecah kata menjadi kata dasar
2.	Pembobotan	function weighting()	Fungsi untuk menghitung *Term Frequency* (tf), jumlah kemunculan kata pada semua dokumen (df), dan bobot tf-idf semua term.
3.	Improved KNN	function docossim()	Fungsi untuk menghitung cosine similarity antara dokumen uji dengan dokumen latih.
		function klasifikasi()	Fungsi untuk mengelompokkan dokumen *tweets* ke dalam kelas sentimen tertentu.

Sumber: Implementasi

4.3.1 Proses *Preprocessing*

Tahap ini terdiri dari pembersihan dokumen, *case folding*, *tokenizing*, *stopword removal*, dan *stemming*. Pembersihan dokumen berfungsi untuk menghilangkan tanda baca pada dokumen, *Case folding* berfungsi untuk mengubah dokumen menjadi huruf kecil, *tokenizing* berfungsi untuk memecah kalimat menjadi kata yg berdiri sendiri, *Stopword removal* berguna untuk menghilangkan kata-kata yang tidak penting dan mengambil kata penting menjadi *keyword*, dan *stemming* mengubah kata penting pada dokumen agar menjadi bentuk kata dasar. *Stemming* yang digunakan adalah *stemming* arifin-setiono. *Sourcecode stemming* arifin-setiono dapat dilihat pada lampiran. Pemanggilan fungsi *text preprocessing* secara keseluruhan ditunjukkan pada *Sourcecode* 4.1.

```
1  // cleansing
2  $hasil_cleansing    =  cleansing($datauji);
3  // casefolding
4  $hasil_casefolding =  casefolding($hasil_cleansing);
5  // tokenizing
6  $hasil_tokenizing  =  tokenizing($hasil_casefolding);
7  // stopword removal
8  $hasil_stopword     =  stopword_removal($hasil_tokenizing);
9  // stemming
10 $hasil_stemming    =  stemming($hasil_stopword);
```

Sourcecode 4. 1 Implementasi Algoritma *Preprocessing* pada Dokumen

Sumber: Implementasi

4.3.1.1 Proses Pembersihan Dokumen

Fungsi *cleansing* pada *preprocessing* merupakan tahap yang pertama kali dilakukan pada dokumen baik dokumen latih maupun dokumen uji. Fungsi *cleansing* berfungsi untuk menghilangkan tanda baca pada data *tweets* yang telah diambil untuk kemudian diolah ke proses selanjutnya. Implementasi pembersihan dokumen ditunjukkan pada *Sourcecode* 4.2.

```
1  Function cleansing($data){
2        $list = array(
3  '.',',','[',']','(',')','{','}','〈',':','،','、','—','!','«',
4  '»','?','`','’','‟','”','"',';','/','⁄','&','*','@','•','^',
5  '†','‡','°','〃','¡','¿','#','÷','ª','%','‰','
6  ','¶','‴','§','~','_','¦','|','©','®','℠','Ⓟ','™','⁂','☞','؟
7  ','؛','◊','※','ˆ
8  ',
```

```
9  ','₳','฿','¢','¢','₡','₲','₤','$','₫','₺','₯','€','ƒ','₣','₲'
10 ,'₴','₭','□','₥','₥','₦','₧','₱','﷼','£','₨','₪','₮','₩','¥'
11 ,'§','1','2','3','4','5','6','7','8','9','0'
12 );
13       return str_replace($list, " ", $data);
14 }
```

Sourcecode 4. 2 Implementasi Algoritma Pembersihan Dokumen

Sumber: Implementasi

4.3.1.2 Proses *Casefolding*

Fungsi *casefolding* merupakan tahap preprocessing selanjutnya yang akan dilakukan pada dokumen setelah melalui proses pembersihan dokumen. Tahap casefolding menyeragamkan bentuk dokumen menjadi huruf kecil. Implementasi fungsi *casefolding* ditunjukkan pada *Sourcecode* 4.3.

```
1 function casefolding($data){
2       $data = strtolower($data);
3       return $data;
4 }
```

Sourcecode 4. 3 Implementasi Algoritma *Casefolding*

Sumber: Implementasi

4.3.1.3 Proses *Tokenizing*

Fungsi *tokenizing* merupakan tahap *preprocessing* yang akan memecah dokumen berdasarkan spasi sehingga didapatkan kata. Implementasi fungsi *Tokenizing* ditunjukkan pada *Sourcecode* 4.4.

```
1 function tokenizing ($data){
2       $hasil = explode(" ", $data);
3       return $hasil;
4 }
```

Sourcecode 4. 4 Implementasi Algoritma *Tokenizing*

Sumber: Implementasi

4.3.1.4 Proses *Filtering/Stopword Removal*

Fungsi *stopword_removal* merupakan tahap *preprocessing* yang akan menghilangkan kata-kata tidak penting. Tahap pertama pada proses *stopword_removal* adalah mengambil setiap kata pada dokumen kemudian dibandingkan dengan kamus *stopword* yang berada di variabel `$directory`, jika ditemukan kata tidak penting maka akan dihapus, sehingga nantinya akan menghasilkankata unik yang disebut sebagai *keyword*. Implementasi fungsi *stopword removal* ditunjukkan pada *Sourcecode* 4.5.

```
1  function stopword_removal($array){
2        $directory = array("..daftar stopword di lampiran..");
3        foreach ($array as $key => $value){
4            if (in_array($value, $directory)){
5                unset($array[$key]);
6            }
7      }
8  return array_values(array_filter($array));
9        }
```

Sourcecode 4. 5 Implementasi Algoritma *Stopword Removal*

Sumber: Implementasi

4.3.1.5 Proses *Stemming*

Setelah dilakukan proses *stopword_removal*, proses yang dilakukan selanjutnya adalah *stemming*. *Stemming* adalah suatu proses untuk mengubah bentuk kata menjadi kata dasar. *Stemming* yang digunakan dalam skripsi ini adalah *stemming* Arifin-Setiono. *Stemming* ini berdasarkan penelitian yang dilakukan oleh Arifin-Setiono.

4.3.2 Proses Pembobotan (*Term Weighting*)

Proses pembobotan TF-IDF berfungsi untuk mengetahui bobot tiap dokumen. Hasil dari pembobotan TF-IDF nantinya akan digunakan sebagai nilai dari masing-masing dokumen untuk menentukan sentimen dari dokumen pada proses klasifikasi *improved KNN*. *Sourcecode* 4.6 menunjukan proses pembobotan dokumen.

```
function weighting ($data_preprocessing){

$DF = array();
$IDF = array();
for ($i=0; $i < count($data_preprocessing); $i++)
{
      $DF[$i] = 0;
      $count = 0;
      for ($j=0; $j < count($data_preprocessing[$i])-1;
$j++) {
      if($data_preprocessing[$i][$j] >= 1) $count++;
      }
      $DF[$i] = $count;
}

for ($i=0; $i <count($data_preprocessing); $i++)
{
$IDF[$i] = log10(count($data_preprocessing[$i])-1/$DF[$i]);
}

$TF_IDF=array();
for ($i=0; $i <count($data_preprocessing); $i++) {
for ($j=0; $j <count($data_preprocessing[$i]); $j++) {
$TF_IDF[$i][$j] = $IDF[$i] * $data_preprocessing[$i][$j];
      }
}
      return $TF_IDF;

}
```

Sourcecode 4. 6 Implementasi Algoritma Pembobotan

Sumber: Implementasi

4.3.3 Proses Analisis Sentimen

Tahap analisis sentimen terdiri dari dua proses yaitu penghitungan *cosine similarity* dan klasifikasi dengan menggunakan *Improved K-Nearest Neighbor*. Tahap *cosine similarity* menghitung kemiripan dokumen uji dengan dokumen *training*, yang hasilnya kemudian diurutkan secara *descending* atau dari terbesar ke yang terkecil. Nilai *cosine similarity* yang sudah diurutkan kemudian akan diolah lagi pada proses selanjutnya yaitu *Improved K-Nearest Neighbor* untuk didapatkan hasil analisis sentimennya.

4.3.3.1 Proses Hitung *Cosine Similarity*

Proses hitung *cosine similarity* ini dilakukan untuk menghitung kemiripan antara dokumen uji dengan dokumen training. Nilai *cosine similarity* yang telah dihitung selanjutnya akan digunakan sebagai acuan untuk mengklasifikasi sentimen dengan *Improved K-Nearest Neighbor* (KNN). *Sourcecode* 4.7 merupakan implementasi dari fungsi untuk menghitung *cosine similarity*.

```
1   function docossim($TF_IDF){
2
3         $similarity = array();
4         for ($i=0; $i <count($TF_IDF); $i++) {
5               $last_index = count($TF_IDF[$i]);
6               for ($j=0; $j<$last_index-1; $j++){
7                     $similarity[$i][$j] = $TF_IDF[$i][$j] *
8   $TF_IDF[$i][$last_index-1];
9               }
10        }
11
12        $sum_similarity = array();
13        $coloumn = count($similarity[0]);
14        $row     = count($similarity);
15
16        for ($i=0; $i < $coloumn; $i++) {
17              $sum_similarity[$i] = 0;
18              for ($j=0; $j < $row ; $j++) {
19                    $sum_similarity[$i] +=
20  $similarity[$j][$i];
21              }
22        }
23
24        $sqrt_similarity = array();
25        $coloumn = count($TF_IDF[0]);
26        $row     = count($TF_IDF);
27
28        for($i=0; $i < $coloumn; $i++){
29              $sqrt_similarity[$i] = 0;
30              for ($j=0; $j < $row; $j++){
31                    $sqrt_similarity[$i] += $TF_IDF[$j][$i];
32              }
33              $sqrt_similarity[$i] =
34  sqrt($sqrt_similarity[$i]);
35        }
36
37        $cossim = array();
38        for ($i=0; $i <count($sum_similarity); $i++) {
39              //echo "-->
40  ".$sum_similarity[$i]."/".$sqrt_similarity[$i]."*".$sqrt_sim
```

```
41  ilarity[count($sqrt_similarity)-1]."<br>";
42              $cossim[$i] =
43  $sum_similarity[$i]/($sqrt_similarity[$i]*$sqrt_similarity[c
44  ount($sqrt_similarity)-1]);
45          }
46          return $cossim;
47  }
```

Sourcecode 4. 7 Implementasi Algoritma *Cosine Similarity*

Sumber: Implementasi

4.3.3.2 Proses *Improved K-Nearest Neighbor*

Tahap *Improved K-Nearest Neighbor* ini melakukan klasifikasi sentimen dari *tweets* yang diuji. Pada proses ini, dilakukan pengurutan nilai *cosine similarity* secara descending atau dari yang terbesar ke yang terkecil. Setelah nilai *cosine similarity* diurutkan, selanjutnya dilakukan klasifikasi sesuai dengan *improved KNN* decision rule yang telah dibahas pada bab sebelumnya. Hasil akhir dari proses ini adalah berupa klasifikasi sentimen dari tweets yang diuji ke dalam sentimen positif atau negatif. *Sourcecode* 4.8 merupakan implementasi dari fungsi *Improved K-Nearest Neighbor*.

```
1   function klasifikasi($cossim, $label, $nilaiK){
2
3   for ($i=0;$i<count($cossim);$i++)
4   {
5   $tampung[] = array(array($cossim[$i], $label[$i]));
6   }
7   rsort($tampung);
8
9   $dataK1 = 0;$CK1 = array();
10  $dataK2 = 0;$CK2 = array();
11  for ($i=0; $i <count($tampung) ; $i++) {
12  switch ($tampung[$i][0][1]) {
13          case 1:
14          $dataK1++;
15          array_push($CK1,$tampung[$i][0][0]);
16          break;
17          case -1:
18          $dataK2++;
19          array_push($CK2,$tampung[$i][0][0]);
20          break;
21          }
22          }
23
24  $sortData = array($dataK1,$dataK2);
```

```
25  rsort($sortData);
26
27  $jumlahDataTerbanyak = $sortData[0];
28
29  $newK1 = floor(($nilaiK * $dataK1)/$jumlahDataTerbanyak);
30  $newK2 = floor(($nilaiK * $dataK2)/$jumlahDataTerbanyak);
31
32  $hasilakhir = array();
33  // cari positif
34  if($newK1!=0){
35        $up = 0;
36        $down = 0;
37        $label = 1;
38        for ($i=0; $i <$newK1; $i++) {
39              $kali = 1;
40              if($tampung[$i][0][1] != $label){
41                    $kali = 0;
42              }
43              $up += ($tampung[$i][0][0] * $kali);
44              $down+=$tampung[$i][0][0];
45        }
46        $hasilakhir["positif"] = $up/$down;
47  }
48
49  // cari negatif
50  if($newK2!=0){
51        $up = 0;
52        $down = 0;
53        $label = -1;
54        for ($i=0; $i <$newK2; $i++) {
55              $kali = 1;
56              if($tampung[$i][0][1] != $label){
57                    $kali = 0;
58              }
59              $up += ($tampung[$i][0][0] * $kali);
60              $down+=$tampung[$i][0][0];
61        }
62        $hasilakhir["negatif"] = $up/$down;
63  }
64
65  asort($hasilakhir);
66  end($hasilakhir);
67
68  return
69  array('hasil'=>key($hasilakhir),'newK'=>array($newK1,$newK2)
70  );
71  }
```

Sourcecode 4. 8 Implementasi Algoritma *Improved K-Nearest Neighbor*

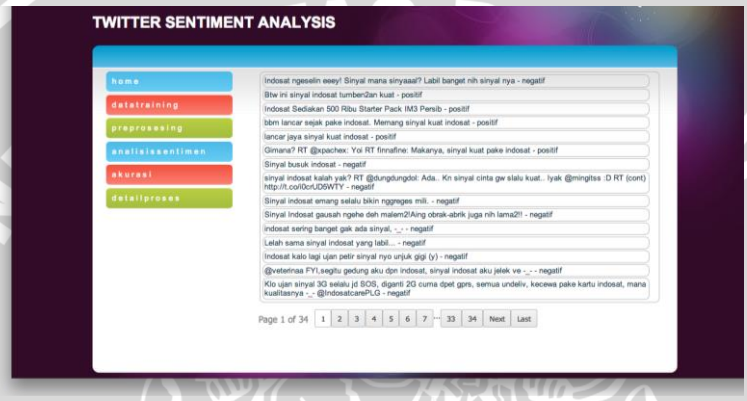
Sumber: Implementasi

4.4 Implementasi Antar Muka

Antarmuka aplikasi sentimen analisis pada Twitter berbahasa Indonesia digunakan oleh pengguna untuk berinteraksi dengan program..

4.4.1 Tampilan Halaman Data *Training*

Tampilan halaman cek kemiripan dokumen akan menampilkan daftar data latih yang digunakan pada aplikasi ini. Gambar halaman data latih ditunjukkan pada Gambar 4.1.

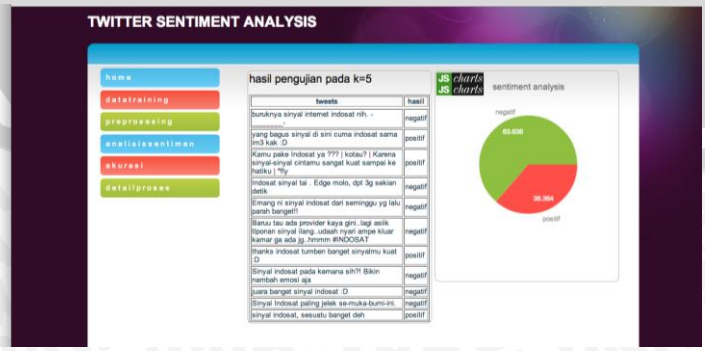


Gambar 4. 1 Halaman *Data Training*

Sumber: Implementasi

4.4.2 Tampilan Halaman Analisis Sentimen

Halaman Analisis Sentimen akan menampilkan *form option* k dan daftar dokumen uji dengan *checkbox* di sebelah kiri data *tweets* untuk memudahkan pengguna memilih dokumen dan jumlah tetangga yang digunakan pada proses analisis sentiment. Halaman analisis sentimen ditunjukkan pada Gambar 4.2.



Gambar 4. 2 Tampilan Halaman Analisis Sentimen

Sumber: Implementasi

BAB V
PENGUJIAN DAN ANALISIS

Pada bab ini akan dipaparkan tentang pengujian terhadap hasil pengkategorian sistem dengan pengkategorian secara *manual*, pengujian efektifitas terhadap sistem ini akan menggunakan metode *Precision, Recall,* dan *F- measure* seperti yang telah dijelaskan pada bab II.

5.1 Pengujian *Preprocessing*

Pengujian *Preprocessing* ini bertujuan untuk mengetahui bahwa proses *preprocessing* sudah berjalan sesuai dengan perancangan yang telah dibuat sebagaimana dipaparkan pada bab III.

1. Pembersihan Dokumen

Hasil pengujian pembersihan dokumen dapat dilihat pada Tabel 5.1.

Tabel 5. 1 Tabel Pengujian Pembersihan Dokumen

No	Teks	Hasil Pembersihan	Hasil Pembersihan Sistem
1	buruknya sinyal internet indosat nih. -_____-	buruknya sinyal internet indosat nih	buruknya sinyal internet indosat nih
2	yang bagus sinyal di sini cuma indosat sama im3 kak :D	yang bagus sinyal di sini cuma indosat sama im kak	yang bagus sinyal di sini cuma indosat sama im kak
3	Ini sinyal Indosat semacam pilah pilih. Cuma bagus pas di kamar aja. Ke luar kamar langsung drop. -___-	Ini sinyal Indosat semacam pilah pilih Cuma bagus pas di kamar aja Ke luar kamar langsung drop	Ini sinyal Indosat semacam pilah pilih Cuma bagus pas di kamar aja Ke luar kamar langsung drop

Sumber: Pengujian

2. *Tokenizing*

Hasil pengujian *Tokenizing* dapat dilihat pada Tabel 5.2.

Tabel 5. 2 Tabel Pengujian *Tokenizing*

No	Teks	Hasil Tokenizing	Hasil Tokenizing Sistem
1	buruknya sinyal internet indosat nih. -_____-	buruknya sinyal internet indosat	buruknya sinyal internet indosat

		nih	nih
2	yang bagus sinyal di sini cuma indosat sama im3 kak :D	Yang bagus sinyal di sini cuma indosat sama im3 kak	yang bagus sinyal di sini cuma indosat sama im kak
3	Ini sinyal Indosat semacam pilah pilih. Cuma bagus pas di kamar aja. Ke luar kamar langsung drop. -___-	Ini sinyal indosat semacam pilah pilih cuma bagus pas di kamar aja ke luar kamar langsung drop	ini sinyal indosat semacam pilah pilih cuma bagus pas di kamar aja ke luar kamar langsung drop

Sumber: Penguijan

3. Stopword Removal

Hasil pengujian *Stopword Removal* dapat dilihat pada Tabel 5.3.

Tabel 5. 3 Tabel Pengujian *Stopword Removal*

No	Teks	Hasil Stopword Removal	Hasil Stopword Removal Sistem
1	buruknya sinyal internet indosat nih. -_____-	buruknya sinyal internet indosat nih	buruknya sinyal internet indosat nih
2	yang bagus sinyal di sini cuma indosat sama im3 kak :D	Bagus sinyal indosat im kak	bagus sinyal indosat im kak

3	Ini sinyal Indosat semacam pilah pilih. Cuma bagus pas di kamar aja. Ke luar kamar langsung drop. -___-	Sinyal indosat pilah pilih bagus pas kamar aja luar kamar langsung drop	sinyal indosat pilah pilih bagus pas kamar aja luar kamar langsung drop

Sumber: Pengujian

4. *Stemming*

Hasil pengujian *Stemming* dapat dilihat pada Tabel 5.4.

Tabel 5. 4 Tabel Pengujian *Stemming*

No	Teks	Hasil Stemming	Hasil Stemming Sistem
1	buruknya sinyal internet indosat nih. -_____-	Buruk sinyal internet indosat nih	buruk sinyal internet indosat nih
2	yang bagus sinyal di sini cuma indosat sama im3 kak :D	bagus sinyal indosat Im3 kak	bagus sinyal indosat im kak
3	Ini sinyal Indosat semacam pilah pilih. Cuma bagus pas di kamar aja. Ke luar kamar langsung drop. -___-	sinyal indosat pilah pilih bagus pas kamar aja luar kamar langsung drop	sinyal indosat pilah pilih bagus pas kamar aja luar kamar langsung drop

Sumber: Pengujian

5.2 *Precision, Recall,* dan *F-Measure*

Untuk mempelajari pengaruh jumlah data latih dan nilai k terhadap efektivitas sistem klasifikasi maka dilakukan 3 kali uji coba pada masing-masing skenario pengujian. Masing-masing skenario pengujian memiliki jumlah data latih dan nilai k yang berbeda menggunakan 200 data uji. Pengujian ini akan menguji pesan singkat berbahasa Indonesia pada mikroblog *Twitter*, untuk skenario lebih jelasnya akan dipaparkan pada tabel berikut:

Tabel 5. 5 Skenario Pengujian

Skenario	Data Latih			Data Uji		
	Positif	Negatif	Jumlah	Positif	Negatif	Jumlah
1	252	548	800	83	117	200
2	252	448	700	83	117	200
3	252	348	600	83	117	200
4	252	248	500	83	117	200

Sumber : Pengujian

Pada tabel 5.5 diperlihatkan beberapa skenario pada pengujian sistem. Masing-masing skenario menggunakan 200 data uji dan memiliki perbedaan proporsi jumlah data latih yang digunakan dalam melakukan pengklasifikasian.

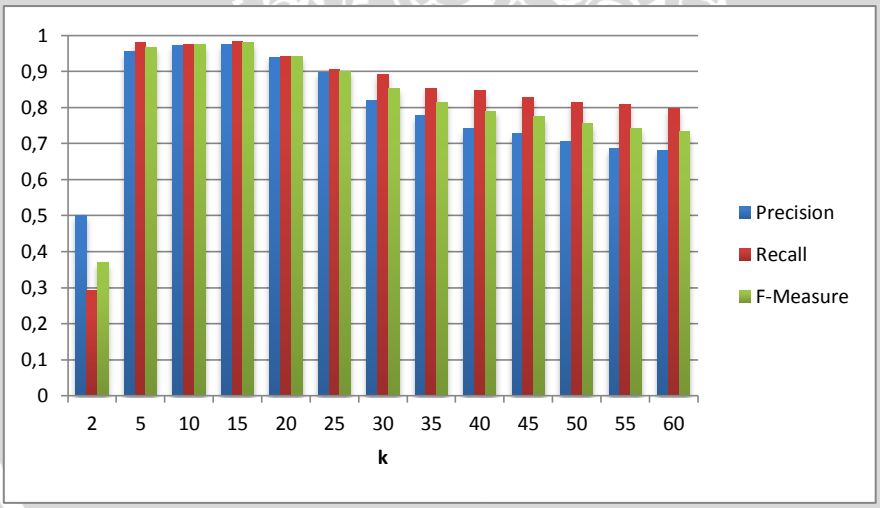
Tabel 5. 6 *Precision*, *Recall*, dan *F-measure* pada skenario 1

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	0	2	0,500	0,293	0,369
5	2	5	0,956	0,979	0,967
10	4	10	0,973	0,975	0,974
15	6	15	0,975	0,983	0,979
20	9	20	0,940	0,942	0,941
25	11	25	0,898	0,904	0,901
30	13	30	0,818	0,891	0,853
35	16	35	0,777	0,852	0,813
40	18	40	0,741	0,846	0,790
45	20	45	0,727	0,829	0,775
50	22	50	0,706	0,813	0,756
55	25	55	0,687	0,807	0,742
60	27	60	0,681	0,797	0,734

Sumber : Pengujian

Tabel 5.6 menunjukkan nilai rata-rata *Precision, Recall,* dan *F-measure* dari 3 kali pengujian dengan menggunakan skenario 1 yang telah dilakukan yaitu 800 data latih dengan proporsi jumlah 252 positif dan 548 negatif pada berbagai nilai masukan k. Nilai k baru positif dan k baru negatif didapatkan melalui hasil perhitungan dengan menggunakan persamaan (2-9). Nilai k baru pada masing-masing kategori berubah-ubah sesuai masukan k yang digunakan dan jumlah data latih masing-masing kategori yang digunakan. Pada skenario ini, *F-measure* tertinggi berada pada nilai k=15 yaitu 0,979 dan terendah pada nilai k=2 yaitu 0,369. Proporsi jumlah data latih yang tidak seimbang pada skenario 1 mengakibatkan pada nilai masukan k=2, nilai k baru yang dihasilkan pada kelas sentimen positif bernilai 0. Hal ini mengakibatkan *F-measure* skenario 1 pada k=2 bernilai rendah.

Tabel 5.6 dapat disajikan dalam bentuk grafik seperti pada Gambar 5.1.



Gambar 5. 1 Grafik Hasil Pengujian Skenario 1

Sumber: Pengujian

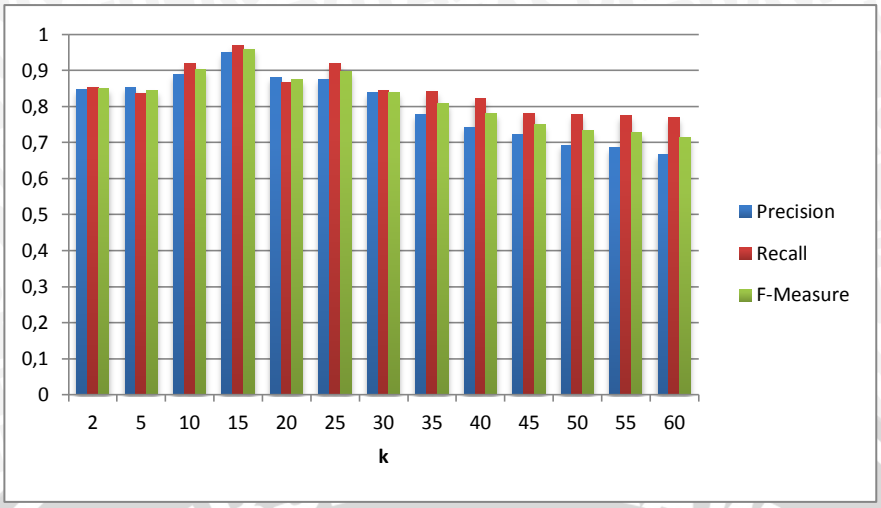
Tabel 5. 7 *Precision, Recall,* dan *F-measure* pada skenario 2

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,846	0,854	0,850
5	2	5	0,853	0,837	0,844
10	5	10	0,888	0,919	0,903
15	8	15	0,950	0,968	0,959
20	11	20	0,881	0,867	0,874
25	14	25	0,876	0,918	0,896
30	16	30	0,838	0,843	0,839
35	19	35	0,777	0,842	0,808
40	22	40	0,743	0,822	0,780
45	25	45	0,722	0,782	0,750
50	28	50	0,693	0,779	0,733
55	30	55	0,686	0,775	0,727
60	33	60	0,666	0,769	0,714

Sumber : Pengujian

Tabel 5.7 menunjukkan nilai rata-rata *Precision, Recall,* dan *F-measure* dari 3 kali pengujian dengan menggunakan skenario 2 yang telah dilakukan yaitu 700 data latih dengan proporsi jumlah 252 positif dan 448 negatif pada berbagai nilai masukan k. Nilai k baru positif dan k baru negatif didapatkan melalui hasil perhitungan dengan menggunakan persamaan (2-9). Nilai k baru pada masing-masing kategori berubah-ubah sesuai masukan k yang digunakan dan jumlah data latih masing-masing kategori yang digunakan. Pada skenario ini, *F-measure* tertinggi berada pada nilai k=15 yaitu 0,959 dan terendah pada nilai k=60 yaitu 0,714.

Tabel 5.7 dapat disajikan dalam bentuk grafik seperti pada Gambar 5.2.



Gambar 5. 2 Grafik Hasil Pengujian Skenario 2

Sumber: Pengujian

Tabel 5. 8 *Precision, Recall,* dan *F-measure* pada skenario 3

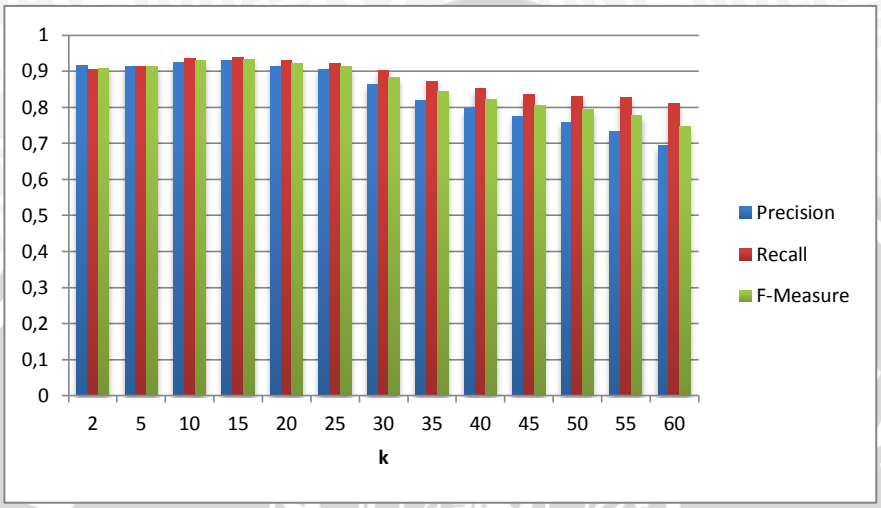
k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,915	0,904	0,909
5	3	5	0,914	0,914	0,914
10	7	10	0,924	0,936	0,930
15	10	15	0,930	0,938	0,934
20	14	20	0,914	0,930	0,922
25	18	25	0,906	0,923	0,914
30	21	30	0,863	0,902	0,882
35	25	35	0,819	0,873	0,845
40	28	40	0,796	0,852	0,823
45	32	45	0,776	0,835	0,804
50	36	50	0,759	0,831	0,793
55	39	55	0,732	0,828	0,777
60	43	60	0,694	0,811	0,748

Sumber : Pengujian

Tabel 5.8 menunjukkan nilai rata-rata *Precision, Recall,* dan *F-measure* dari 3 kali pengujian dengan menggunakan skenario 3 yang telah dilakukan yaitu 600 data latih dengan proporsi jumlah 252 positif dan 348 negatif pada berbagai nilai masukan k. Nilai k baru positif dan k baru negatif didapatkan melalui hasil perhitungan dengan menggunakan persamaan (2-9). Nilai k baru pada masing-masing kategori berubah-ubah sesuai masukan k yang digunakan dan jumlah data

latih masing-masing kategori yang digunakan. Pada skenario ini, *F-measure* tertinggi berada pada nilai k=15 yaitu 0,934 dan terendah pada nilai k=60 yaitu 0,748.

Tabel 5.8 dapat disajikan dalam bentuk grafik seperti pada Gambar 5.3.



Gambar 5. 3 Grafik Hasil Pengujian Skenario 3

Sumber: Pengujian

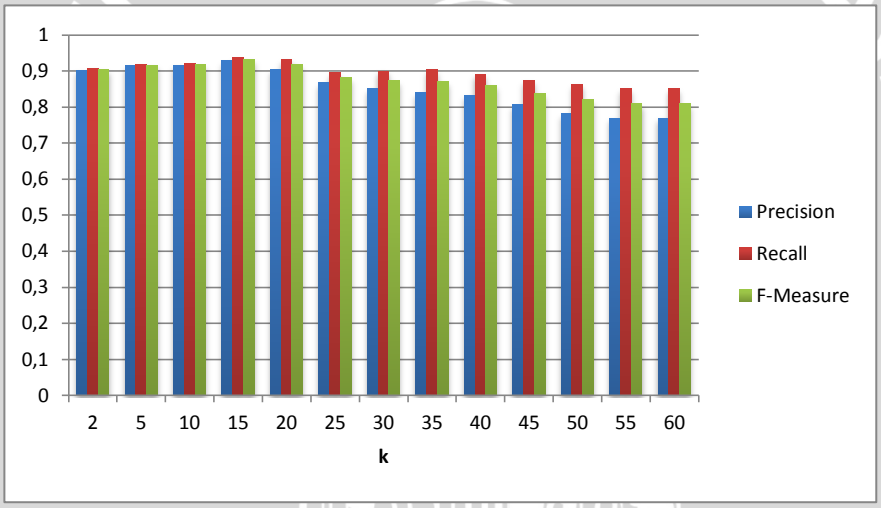
Tabel 5. 9 *Precision*, *Recall*, dan *F-measure* pada skenario 4

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	2	1	0,902	0,907	0,904
5	5	4	0,914	0,918	0,916
10	10	9	0,916	0,919	0,917
15	15	14	0,928	0,936	0,932
20	20	19	0,904	0,932	0,918
25	25	24	0,868	0,895	0,881
30	30	29	0,850	0,899	0,874
35	35	34	0,840	0,903	0,870
40	40	39	0,832	0,891	0,860
45	45	44	0,807	0,873	0,838
50	50	49	0,781	0,863	0,820
55	55	54	0,769	0,852	0,808
60	60	59	0,769	0,852	0,808

Sumber : Pengujian

Tabel 5.9 menunjukkan nilai rata-rata *Precision, Recall,* dan *F-measure* dari 3 kali pengujian dengan menggunakan skenario 4 yang telah dilakukan yaitu 500 data latih dengan proporsi jumlah 252 positif dan 248 negatif pada berbagai nilai masukan k. Nilai k baru positif dan k baru negatif didapatkan melalui hasil perhitungan dengan menggunakan persamaan (2-9). Nilai k baru pada masing-masing kategori berubah-ubah sesuai masukan k yang digunakan dan jumlah data latih masing-masing kategori yang digunakan. Pada skenario ini, *F-measure* tertinggi berada pada nilai k=15 yaitu 0,932 dan terendah pada nilai k=55 dan k=60 yaitu 0,808.

Tabel 5.9 dapat disajikan dalam bentuk grafik seperti pada Gambar 5.4.



Gambar 5. 4 Grafik Hasil Pengujian Skenario 4

Sumber: Pengujian

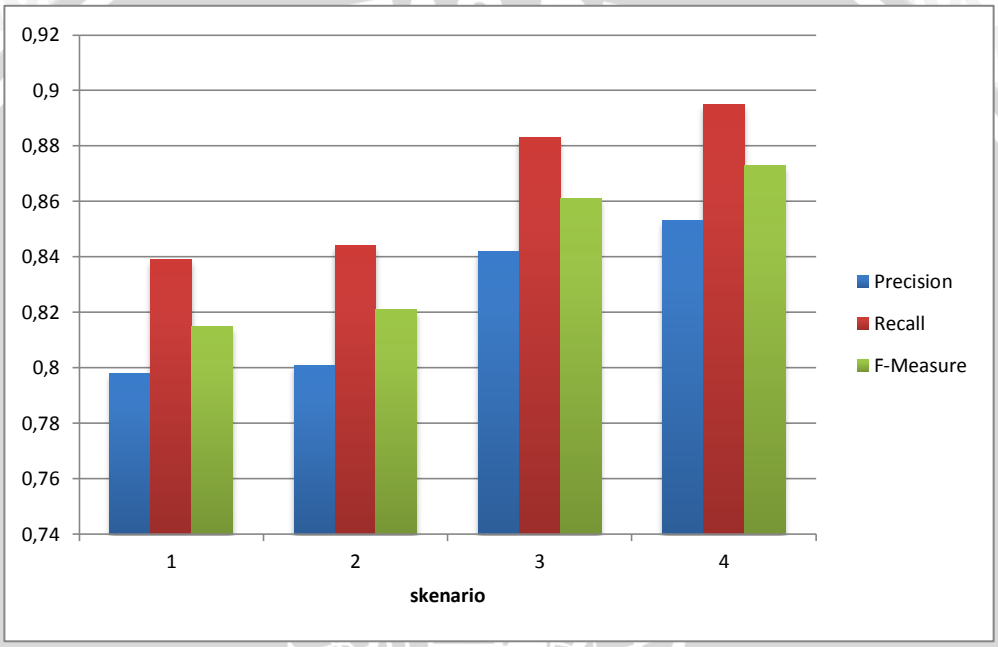
Tabel 5. 10 *Precision, Recall,* dan *F-measure* rata-rata

Skenario	Precision	Recall	F-Measure
1	0,798	0,839	0,815
2	0,801	0,844	0,821
3	0,842	0,883	0,861
4	0,852	0,895	0,873
Rata-rata	0,823	0,865	0,843

Sumber : Pengujian

Tabel 5.10 menunjukkan nilai rata-rata *Precision, Recall,* dan *F-measure* dari masing-masing skenario pengujian yang telah dilakukan. *F-measure* rata-rata terbaik berada pada skenario 4 yaitu dengan jumlah data latih 500 dengan nilai 0,873. Hal ini terjadi karena jumlah proporsi data masing-masing kelas sentimen data training pada skenario 4 lebih seimbang dibandingkan dengan skenario 1, 2, dan 3 yaitu 252 positif dan 248 negatif.

Tabel 5.10 dapat disajikan dalam bentuk grafik seperti pada Gambar 5.5.



Gambar 5. 5 Grafik Rata-Rata Hasil Pengujian

Sumber: Pengujian

5.3 Analisis Hasil

Dari hasil evaluasi seluruh skenario pengujian yang telah dilakukan, dapat diketahui bahwa terdapat beberapa factor yang mempengaruhi ketepatan analisis sentimen dengan menggunakan metode *Improved K-Nearest Neighbor* (KNN), yaitu *data training* dan nilai k yang digunakan. Tabel 5.11 menunjukkan nilai perbandingan *F-measure* tertinggi dan terendah pada masing-masing skenario.

Tabel 5. 11 Tabel Perbandingan *F-measure*

Skenario	Data Latih			*F-measure* tertinggi		*F-measure* terendah	
	Positif	Negatif	Jumlah	Nilai	k	nilai	k
1	252	548	800	0,979	15	0,369	2
2	252	448	700	0,959	15	0,714	60
3	252	348	600	0,934	15	0,748	60
4	252	248	500	0,932	15	0,808	55,60

Sumber: Pengujian

Dari pengujian dengan menggunakan 200 data uji yang telah dilakukan, diketahui bahwa semakin banyak jumlah data latih, semakin baik nilai *F-measure* yang dihasilkan. Seperti yang dapat dilihat pada Tabel 5.11, nilai *F-measure* tertinggi berada pada skenario 1 dengan jumlah data latih 800 yang bernilai 0,979 dan cenderung menurun seiring berkurangnya jumlah data latih. Walaupun nilai *F-measure* tertinggi berada pada skenario 1, namun *F-measure* terendah bernilai 0.369 juga didapatkan dari hasil pengujian sistem pada skenario 1. Hal ini disebabkan proporsi jumlah data latih positif dan negatif yang tidak seimbang pada skenario 1, yaitu 252 positif dan 548 negatif. Proporsi jumlah data latih positif dan negatif yang tidak seimbang mengakibatkan pada saat proses pengujian, hasil analisis sentimen memiliki kecenderungan terklasifikasi ke kelas yang lebih banyak jumlah data latihnya. Pada Tabel 5.10, didapatkan nilai rata-rata *F-measure* terbaik berada pada skenario 4 dengan jumlah data latih 500 dengan nilai 0,873 karena skenario ini memiliki proporsi data latih yang paling seimbang yaitu 252 positif dan 248 negatif. Dapat ditarik kesimpulan semakin seimbang proporsi jumlah masing-masing kelas yang digunakan dalam tahap pembelajaran maka semakin meningkat nilai *Precision, Recall,* dan *F-measure*. Oleh karena itu dibutuhkan kecermatan dalam memilih besarnya jumlah dan isi dari data latih, agar sistem dapat berjalan dengan baik dan seimbang.

Pada tabel 5.11 juga dapat diamati bahwa nilai *F-measure* terendah pada skenario 2, 3, dan 4 berada pada k=60. Hanya pada skenario 1 *F-measure* terendah berada pada k=2, seperti yang sudah dijelaskan pada bagian sebelumnya. Dari tabel hasil pengujian setiap skenario dapat ditarik kesimpulan semakin meningkat nilai k, nilai *Precision, Recall,* dan *F-measure* semakin menurun, karena kemungkinan terjadinya kesalahan klasifikasi semakin besar.

BAB VI
PENUTUP

Bab ini akan membahas mengenai kesimpulan dan saran yang dapat diambil dari pembuatan analisis sentimen pada Twitter berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor* (KNN).

6.1 Kesimpulan

Kesimpulan dari hasil skripsi sentimen analisis pada Twitter berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor* (KNN) sebagai berikut:

1. Metode *Improved K-Nearest Neighbor* (KNN) pada analisis sentimen *Twitter* berbahasa Indonesia dapat diterapkan untuk mengklasifikasikan sentimen dari dokumen *tweets* secara otomatis, dengan cara melakukan *preprocessing* pada dokumen sehingga didapatkan kata-kata, kemudian menghitung bobot dan similaritas dari dokumen uji terhadap dokumen semua latih serta mengurutkan similaritasnya, setelah itu dihitung probabilitasnya pada masing-masing kelas terhadap n-tetangga terdekat, dimana nilai n adalah nilai k masukan pengguna yang sudah dimodifikasi.
2. Pengujian analisis sentimen pada Twitter berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor* (KNN) menghasilkan rata-rata *Precision* (akurasi) sebesar 0,823, rata-rata *Recall* (sensitifitas atau kemampuan sistem memilih hasil yang sesuai) sebesar 0,865, dan rata-rata *F-measure* sebesar 0,843.
3. Jumlah dokumen dan keseimbangan proporsi data latih, serta nilai k yang digunakan berpengaruh terhadap baik atau tidaknya proses pengklasifikasian dokumen *tweets*.

6.2 Saran

Saran dari hasil skripsi analisis sentimen pada Twitter berbahasa Indonesia dengan metode *Improved K-Nearest Neighbor* (KNN) untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Analisis sentimen ini menghitung kemiripan berdasarkan frekuensi kemunculan kata, sehingga untuk mendapatkan hasil yang optimal sebaiknya digunakan sistem untuk pengecekan sinonim kata, dan pengecekan kemiripan berdasarkan makna.
2. Metode *Improved K-Nearest Neighbor* pada sistem ini kurang dapat menangani jumlah data latih yang tidak seimbang secara signifikan, sehingga dapat dilakukan pengembangan sistem dengan menggunakan metode lain seperti *Weighted K-Nearest Neighbor* atau menggabungkan metode lain dengan *Improved K-Nearest Neighbor* sehingga dihasilkan sistem yang lebih baik.

DAFTAR PUSTAKA

[BPL-08] Bo Pang and Lilian Lee. 2008. *Opinion Mining and Sentiment Analysis, Foundations and Trends in Information Retrieval*, vol. Volume 2, no. Issue 1-2, pp. 1-135.

[BPL-02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. *Thumbs up? Sentiment Classification using Machine Learning, in Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, vol. Volume 10, pp. 79–86, Morristown, NJ, USA.

[JNR-05] Read, Jonathon. 2005. *Use Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification. The Association for Computer Linguistics* (ACL).

[BGL-10] Bing Liu. 2010. *Sentiment Analysis and Subjectivity, in Handbook of Natural Language Processing*.

[BGL-10] Bing Liu. 2010. *Sentiment Analysis: A Multi-Faceted Problem, IEEE Intelligent Systems.*

[TSK-06] P.N. Tan, M. Steinbach, dan V. Kumar. 2006. *Introduction to Data Mining*. *1st penyunt. Boston: Pearson Addison Wesley*.

[BSQ-03] Baoli, Li., Shiwen, Yu., dan Qin, Lu. 2003. *An Improved k-Nearest Neighbors for Text Categorization. To appear in the Proceedings of the 20th International Conference of Computer Processing of Oriental Language.*

[APP-10] Alexander Pak, Patrick Paroubek. 2010. *Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Universite de Paris-Sud, Laboratoire LIMSI-CNRS, Batiment 508, F-91405 Orsay Cedex, France.*

[ALG-09] Alec Go, Lei Huang, *and* Richa Bhayani. 2009. *Twitter Sentiment Analysis. Final Projects from CS224N for Spring 2008/2009. The Stanford Natural Language Processing Group.*

[FRS-07] Feldman, Ronen and Sanger, James. 2007. *The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data.* New York:Cambridge University Press.

[CDF-08] Cahyono, Dwi dan Fadlil, Junaidillah dan Sumpeno, Surya dan Hariadi, Mochamad. *2008. Temu Kembali Informasi Untuk Pembangkitan Basis Pengetahuan Dari Teks Bebas Yang Digunakan Oleh Agen Percakapan Bahasa Alami*, Surabaya, Institut Teknologi Sepuluh November.

[AHA-10] Hatta A, Ahmad. 2010. *Rancang Bangun Sistem Pengelolaan Dokumen Dokumen Penting Menggunakan Text Mining.* Surabaya, Institut Teknologi Sepuluh November.

[DGT-09] Putri, Amelia Yosi. 2009. *Stemming*. http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=574:stemming-&catid=20:informatika&Itemid=14 diakses tanggal 25 september 2012

[ASA-01] Zainal Arifin, Agus dan Setiono,Ari Novan. 2001. *Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering*. Surabaya, Institut Teknologi Sepuluh November.

[HEL-09] Elisabeth, Hendrice. 2009. Klasifikasi Teks Berita dengan Weight Adjusted K-Nearest Neighbor. Jakarta: IT-Telkom.

[YYJ-99] Yiming Yang, Jaime G.Carbonell, Rulf D. Brown, Thomas Pierce, Brian T. Achibald, Xin Liu. 1999. *Learning Approaches for Detecting and Tracking News Events*. IEEE Intelligent Systems, Language Techlonolies Institute, Carbegie Mellon University.

[KAO-07] Kao, A., dan Poteet, Stephen R. 2007. *Natural Language Processing and Text Mining*. London : Springer

[SAL-12] Saleh, Muhammad. 2012. *Menentukan Kemiripan Topik Tugas Akhir Berdasarkan Deskripsi Pada Jurusan Teknik Informatika Menggunakan Metode Inner Product dan Single Linkage Hierarchical*. Skripsi Teknik Informatika, Fakultas Teknik,

Universitas Muhammadiyah Malang.

[RIZ-12] Rizqa, Arroyida. 2012. *Sistem Penilaian Otomatis Jawaban Essay Menggunakan Deteksi Similarity*. Tugas Akhir, Program Studi Teknik Informatika, Universitas Pembangunan Veteran, Jawa Timur.

[YYX-09] Yong Z, Youwen L, Xhixion X. 2009. *An Improved kNN Text Classification Algotihm based on Clustering, J+ournal of Computers, Vol. 4, No. 3. Conference on Neural Information Processing.*

[AXB-01] Bergo, Alexander. 2001. *Text Categorization and Prototypes*. http://www.illc.uva.nl/Publications/ResearchReports/MoL-2001-08.text.pdf. Diakses pada tanggal 15 Februari 2013.

[IGA-11] Widiarsana, I.G.A.O dkk. 2011. *Data Mining*: Metode Klasifikasi *K-Nearest Neighbor*. Bali: Teknik Elektro Universitas Udayana.

[RSY-10] Yuniarti, Ressty. 2010. *Implementasi Sistem Temu Kembali Citra Berbasis Isi Dengan Metode Jarak Informasi Yang Dinormalisasi.*

[PDM-11] Powers, David M W. 2011. *Evaluation: From Precision, Recall And F-Measure To Roc, Informedness, Markedness & Correlation.* Journal of Machine Learning Technologies 2 (1): 37–63.

[GHR-05] Hripcsak, George and S Rothschild, Adam. 2005. *Agreement,The F-Measure, and Reliability in Information Retrieval*. *J Am Med Inform Assoc* 2005

LAMPIRAN

LAMPIRAN 1 Daftar *Stopword*

ada	banyak	dahulu	hendak	karenanya	masing	rupanya	segala	seolah	tadinya
adalah	beberapa	dalam	hendaklah	kau	masingmasing	saat	segalanya	seolah-olah	tak
adanya	begini	dan	hendaknya	ke	mau	saatnya	segera	seorang	tanpa
adapun	beginian	dapat	hingga	kecil	maupun	saja	seharusnya	sepanjang	tapi
agak	beginikah	dari	ia	kemudian	melainkan	sajalah	sehingga	sepantasnya	telah
agaknya	beginilah	daripada	ialah	kenapa	melakukan	saling	sejak	sepantasnyalah	tentang
agar	begitu	dekat	ibarat	kepada	melalui	sama	sejenak	seperti	tentu
akan	begitukah	demi	ingin	kepadanya	memang	sama-sama	sekali	sepertinya	tentulah
akankah	begitulah	demikian	inginkah	ketika	mengapa	sambil	sekali-kali	sering	tentunya
akhirnya	begitupun	demikianlah	inginkan	khususnya	menjadi	sampai	sekalian	seringnya	terdiri
aku	belakangan	dengan	ini	kini	mereka	sana	sekaligus	serta	terhadap
akulah	belum	depan	inikah	kinilah	merekalah	sangat	sekalipun	serupa	terhadapnya
amat	belumlah	di	inilah	kiranya	merupakan	sangatlah	sekarang	sesaat	terlalu
amatlah	berapa	dia	itu	kita	meski	saya	seketika	sesama	terlebih
anda	berapakah	dialah	itukah	kitalah	meskipun	sayalah	sekiranya	sesegera	tersebut
andalah	berapalah	diantara	itulah	kok	mungkin	se	sekitar	sesekali	tersebutlah
antar	berapapun	diantaranya	jangan	lagi	mungkinkah	sebab	sekitarnya	seseorang	tertentu
antara	berdasarkan	dikarenakan	jangankan	lagian	nah	sebabnya	sela	sesuatu	tetapi
antaranya	berkali	dilakukan	janganlah	lah	namun	sebagai	selain	sesuatunya	tiap
apa	bermacam	dini	jika	lain	nanti	sebagaimana	selaku	sesudah	tidak
apaan	bersama	diperoleh	jikalau	lainnya	nantinya	sebagainya	selalu	sesudahnya	tidakkah
apabila	bersama-sama	diri	juga	lalu	nyaris	sebaliknya	selama	setelah	tidaklah
apakah	berupa	dirinya	justru	lama	oleh	sebanyak	selama-lamanya	seterusnya	toh
apalagi	betulkah	disini	kala	lamanya	olehkah	sebegini	selamanya	setiap	untuk
apatah	biasa	disinilah	kalau	lebih	olehnya	sebegitu	selanjutnya	setidak-tidaknya	waduh
atau	biasanya	dong	kalaulah	macam	pada	sebelum	seluruh	setidaknya	wah
ataukah	bila	dulu	kalaupun	maka	padahal	sebelumnya	seluruhnya	sewaktu	wahai
ataupun	bilakah	enggak	kali	makanya	padanya	sebenarnya	semacam	siapa	walau
bagai	bisa	enggaknya	kalian	makin	paling	seberapa	semakin	siapakah	walaupun
bagaikan	bisakah	entah	kami	malah	pantas	sebetulnya	semasih	siapapun	wong
bagaimana	boleh	entahlah	kamilah	malahan	para	sebisanya	semaunya	sini	yaitu
bagaimanakah	bolehlah	hal	kamu	mampu	pasti	sebuah	sementara	sinilah	yakni
bagaimanapun	buat	hampir	kamulah	mampukah	pastilah	secara	sempat	suatu	yang
bagi	bukan	hanya	kan	mana	per	sedang	semua	sudah	
bahkan	bukankah	hanyalah	kapan	manakala	percuma	sedangkan	semuanya	sudahkah	
bahwa	bukanlah	harus	kapankah	manalagi	pernah	sedemikian	semula	sudahlah	
bahwasanya	bukannya	haruslah	kapanpun	masih	pula	sedikit	sendiri	supaya	
banget	cuma	harusnya	karena	masihkah	pun	sedikitnya	sendirinya	tadi	

LAMPIRAN 2 Tabel Hasil Pengujian 1 Skenario 1

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	0	2	0,500	0,293	0,369
5	2	5	0,979	0,979	0,979
10	4	10	0,973	0,975	0,974
15	6	15	0,976	0,983	0,979
20	9	20	0,970	0,980	0,975
25	11	25	0,940	0,961	0,950
30	13	30	0,898	0,937	0,917
35	16	35	0,819	0,898	0,857
40	18	40	0,777	0,880	0,825
45	20	45	0,741	0,866	0,799
50	22	50	0,711	0,855	0,776
55	25	55	0,687	0,846	0,758
60	27	60	0,681	0,844	0,754

LAMPIRAN 3 Tabel Hasil Pengujian 2 Skenario 1

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	0	2	0,500	0,293	0,369
5	2	5	0,950	0,979	0,964
10	4	10	0,973	0,975	0,974
15	6	15	0,962	0,983	0,972
20	9	20	0,873	0,897	0,885
25	11	25	0,802	0,825	0,813
30	13	30	0,745	0,844	0,791
35	16	35	0,705	0,798	0,749
40	18	40	0,683	0,798	0,736
45	20	45	0,680	0,791	0,731
50	22	50	0,648	0,760	0,700
55	25	55	0,634	0,757	0,690
60	27	60	0,631	0,742	0,682

LAMPIRAN 4 Tabel Hasil Pengujian 3 Skenario 1

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	0	2	0,500	0,293	0,369
5	2	5	0,940	0,979	0,959
10	4	10	0,973	0,975	0,974
15	6	15	0,986	0,983	0,984
20	9	20	0,977	0,950	0,963
25	11	25	0,951	0,926	0,938
30	13	30	0,811	0,892	0,850
35	16	35	0,807	0,860	0,833
40	18	40	0,763	0,860	0,809
45	20	45	0,760	0,831	0,794
50	22	50	0,759	0,825	0,791
55	25	55	0,740	0,818	0,777
60	27	60	0,731	0,805	0,766

LAMPIRAN 5 Tabel Hasil Pengujian 1 Skenario 2

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,948	0,941	0,944
5	2	5	0,923	0,933	0,928
10	5	10	0,861	0,899	0,880
15	8	15	0,895	0,931	0,913
20	11	20	0,770	0,750	0,760
25	14	25	0,757	0,852	0,802
30	16	30	0,738	0,843	0,787
35	19	35	0,726	0,837	0,778
40	22	40	0,702	0,826	0,759
45	25	45	0,672	0,811	0,735
50	28	50	0,654	0,802	0,720
55	30	55	0,642	0,795	0,710
60	33	60	0,654	0,802	0,720

LAMPIRAN 6 Tabel Hasil Pengujian 2 Skenario 2

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,827	0,851	0,839
5	2	5	0,834	0,763	0,797
10	5	10	0,888	0,916	0,902
15	8	15	0,972	0,987	0,979
20	11	20	0,892	0,931	0,911
25	14	25	0,892	0,931	0,911
30	16	30	0,883	0,811	0,845
35	19	35	0,763	0,810	0,786
40	22	40	0,725	0,806	0,763
45	25	45	0,697	0,716	0,706
50	28	50	0,684	0,713	0,698
55	30	55	0,678	0,709	0,693
60	33	60	0,643	0,692	0,667

LAMPIRAN 7 Tabel Hasil Pengujian 3 Skenario 2

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,762	0,771	0,766
5	2	5	0,803	0,814	0,808
10	5	10	0,915	0,943	0,929
15	8	15	0,983	0,987	0,985
20	11	20	0,982	0,920	0,950
25	14	25	0,978	0,972	0,975
30	16	30	0,892	0,875	0,883
35	19	35	0,841	0,879	0,860
40	22	40	0,802	0,834	0,818
45	25	45	0,796	0,819	0,807
50	28	50	0,741	0,823	0,780
55	30	55	0,738	0,821	0,777
60	33	60	0,702	0,814	0,754

LAMPIRAN 8 Tabel Hasil Pengujian 1 Skenario 3

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,918	0,907	0,912
5	3	5	0,887	0,895	0,891
10	7	10	0,903	0,924	0,913
15	10	15	0,903	0,924	0,913
20	14	20	0,885	0,913	0,899
25	18	25	0,879	0,910	0,894
30	21	30	0,825	0,879	0,851
35	25	35	0,770	0,850	0,808
40	28	40	0,746	0,838	0,789
45	32	45	0,734	0,831	0,779
50	36	50	0,722	0,825	0,770
55	39	55	0,710	0,819	0,761
60	43	60	0,694	0,800	0,743

LAMPIRAN 9 Tabel Hasil Pengujian 2 Skenario 3

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,914	0,907	0,910
5	3	5	0,928	0,914	0,921
10	7	10	0,937	0,939	0,938
15	10	15	0,945	0,945	0,945
20	14	20	0,937	0,939	0,938
25	18	25	0,927	0,931	0,929
30	21	30	0,897	0,914	0,905
35	25	35	0,874	0,893	0,883
40	28	40	0,845	0,887	0,865
45	32	45	0,823	0,834	0,828
50	36	50	0,792	0,831	0,811
55	39	55	0,773	0,837	0,804
60	43	60	0,713	0,820	0,763

LAMPIRAN 10 Tabel Hasil Pengujian 3 Skenario 3

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	1	2	0,914	0,897	0,905
5	3	5	0,927	0,932	0,929
10	7	10	0,931	0,945	0,938
15	10	15	0,942	0,945	0,943
20	14	20	0,921	0,939	0,930
25	18	25	0,913	0,927	0,920
30	21	30	0,867	0,914	0,890
35	25	35	0,814	0,876	0,844
40	28	40	0,798	0,832	0,815
45	32	45	0,772	0,841	0,805
50	36	50	0,763	0,836	0,798
55	39	55	0,713	0,827	0,766
60	43	60	0,675	0,814	0,738

LAMPIRAN 11 Tabel Hasil Pengujian 1 Skenario 4

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	2	1	0,942	0,936	0,939
5	5	4	0,929	0,921	0,925
10	10	9	0,919	0,911	0,915
15	15	14	0,925	0,916	0,920
20	20	19	0,894	0,911	0,902
25	25	24	0,844	0,844	0,844
30	30	29	0,819	0,876	0,847
35	35	34	0,823	0,885	0,853
40	40	39	0,817	0,881	0,848
45	45	44	0,811	0,878	0,843
50	50	49	0,775	0,860	0,815
55	55	54	0,763	0,855	0,806
60	60	59	0,763	0,855	0,806

LAMPIRAN 12 Tabel Hasil Pengujian 2 Skenario 4

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	2	1	0,836	0,852	0,844
5	5	4	0,876	0,892	0,884
10	10	9	0,889	0,892	0,890
15	15	14	0,914	0,934	0,924
20	20	19	0,879	0,924	0,901
25	25	24	0,834	0,911	0,871
30	30	29	0,813	0,892	0,851
35	35	34	0,805	0,887	0,844
40	40	39	0,793	0,878	0,833
45	45	44	0,778	0,843	0,809
50	50	49	0,769	0,861	0,812
55	55	54	0,746	0,834	0,788
60	60	59	0,746	0,834	0,788

LAMPIRAN 13 Tabel Hasil Pengujian 3 Skenario 4

k	n (k baru)		Precision	Recall	F-Measure
	Positif	Negatif			
2	2	1	0,928	0,932	0,930
5	5	4	0,937	0,941	0,939
10	10	9	0,941	0,953	0,947
15	15	14	0,946	0,957	0,951
20	20	19	0,938	0,962	0,950
25	25	24	0,927	0,930	0,928
30	30	29	0,919	0,928	0,923
35	35	34	0,892	0,937	0,914
40	40	39	0,885	0,914	0,899
45	45	44	0,831	0,897	0,863
50	50	49	0,798	0,867	0,831
55	55	54	0,798	0,867	0,831
60	60	59	0,798	0,867	0,831