# ANALISIS SENTIMEN TERHADAP TELKOM INDIHOME BERDASARKAN OPINI PUBLIK MENGGUNAKAN METODE IMPROVED K-NEAREST NEIGHBOR

Herdiawan[1]

[1]Teknik Informatika – Universitas Komputer Indonesia
Jalan Dipati Ukur No. 112-114-116, 40132 Bandung, Indonesia
Email: if.herdiawan@gmail.com[1]

**ABSTRAK**

Indihome adalah salah satu produk layanan intenet terbaru dari PT. Telkom. Pengguna Indihome sampai saat ini mencapai 300 ribu pengguna. Banyaknya jumlah pengguna Indihome yang akan semakin bertambah, membuat PT. Telkom ingin menyediakan layanan berupa *feedback* penilaian produk Indihome agar dapat mengetahui respon dari konsumen terhadap produk Indihome. Banyak konsumen yang membahas tentang Indihome di media sosial khususnya twitter, baik dari kualitas layanan internet yang bagus ataupun sebaliknya. Sayangnya media sosial tidak mempunyai kemampuan untuk mengagregasi informasi mengenai suatu perbincangan yang ada menjadi sebuah kesimpulan.

Salah satu cara untuk menarik kesimpulan dari hasil agregasi adalah menggunakan *text mining*. Algoritma *Improved K-Nearest Neighbor* adalah salah satu algoritma yang bisa dimanfaatkan untuk implementasi pengklasifikasiannya. Proses penyelesaian algoritma *Improved K Nearest Neighbor* diawali dengan *preprocessing* yang terdiri dari *Convert Emoticon, Cleansing, Case Folding, Convert Negation, Tokenizing, Filtering,* dan *Stemming.* Proses selanjutnya pembobotan kata, kemudian pengkategorian yang terdiri dari penghitungan *cosine similarity,* perhitungan nilai *k-values* dan kalsifikasi sentimen berupa grafik. Sehingga hasil dari analisis sentimen ini bisa dijadikan evaluasi dalam menentukan langkah bisnis selanjutnya atau perbaikan kualitas yang lebih baik.

**Kata kunci** : analisis sentimen, *text mining*, klasifikasi, *Improved K-Nearest Neighbor, Indihome*

## 1. PENDAHULUAN

Pengguna Internet di Indonesia semakin bertambah banyak. Menurut Kementrian Komunikasi dan Informatika di tahun 2015 jumlah pengguna Internet di Indonesia telah mencapai angka 150 juta orang, atau sekitar 61% dari total penduduk[1].

PT.Telekomunikasi Indonesia, Tbk selaku penyedia layanan IndiHome dengan menggabungkan beberapa layanan menjadi satu, saat ini memiliki promosi layanan menarik yang mampu menarik banyak konsumen. Mulai banyaknya jumlah pengguna membuat PT.Telekomunikasi Indonesia, Tbk ingin menyediakan layanan berupa *feedback* penilaian terhadap produk IndiHome. Bapak Sony Budi Winarso selaku *Manager Marketing Integration* Reg-3 berencana ingin mengetahui bagaimana respon konsumen terdahap produk Indihome dari media sosial karena menurutnya banyak konsumen Telkom yang memberikan komentar terhadap produk Telkom di media sosial twitter.

Melihat permasalahan tersebut maka perlu adanya cara bagaimana mengklasifikasi informasi sentimen publik terhadap Telkom IndiHome dari opini publik yang ada di sosial media, untuk mendapatkan informasi dari hasil pengklasifikasian data melalui media sosial twitter berupa hasil persentase kepuasan konsumen yang dapat digunakan sebagai bahan evaluasi Telkom Indihome agar dapat lebih meningkatkan kualitas layanannya sehingga dapat memperbaiki dan menentukan langkah bisnis selanjutnya yang lebih baik lagi.

### 1.1. Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini untuk menghasilkan opini yang baru [6]. Analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berpandangan atau beropini negatif atau positif. Analisis Sentimen biasanya dilakukan untuk memantau perkembangan pasar

atau untuk melihat respon terhadap suatu masalah, salah satu contoh penggunaan analisis sentimen dalam dunia nyata adalah identifikasi kecenderungan pasar dan opini pasar terhadap suatu objek produk [7].

Pada dasarnya analisis sentimen merupakan klasifikasi, tetapi tidak semudah proses klasifikasi biasa karena terkait penggunaan bahasa yang terus berkembangan. Dimana media yang digunakan dalam kasus ini adalah sebuah teks yang ambigu karena tidak ada intonasi dalam sebuah teks [8].

Manfaat sentimen analisis terhadap perkembangan suatu bisnis sangat besar, sehingga banyak perusahaan menerapkan analisis sentimen sebagai media untuk melihat perkembangan pasar dalam menentukan langkah bisnis yang diambil sebagai bahan pertimbangan perusahaan tersebut.

### 1.2. Text Mining

*Text Mining* merupakan data berupa teks dimana sumber data biasanya didapatkan dari dokuman, yang bertujuan mencari kata-kata yang dapat mewaliki isi dari dokumen sehingga dapat dilakukan analisis keterhubungan antar dokumen [9].

*Text mining* (penambangan teks) adalah penambangan yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, sesuatu yang tidak diketahui sebelumnya atau menemukan kembali informasi yang tersirat secara implisit, yang berasal dari informasi yang diekstrak secara otomatis dari sumber-sumber data teks yang berbeda-beda.

### 1.3. Regular Expression

*Regular Expression* atau yang biasa disingkat *regex* adalah sebuah teks khusus untuk menggambarkan pencarian sebuah pola. *Regex* biasa digunakan untuk pencarian atau manipulasi teks. *Regex* didukung oleh banyak basaha pemrograman, seperti Java, PHP, C# dan masih banyak bahasa pemrograman lainnya. Berikut adalah aturan-aturan penulisan *Regex* dalam bahasa pemrograman Java [10].

### 1.4. Preprocessing

*Text Preprocessing* yang merupakan tahap awal dari *text mining* yang akan memproses data latih dan data uji. *Text Preprocessing* ini bertujuan untuk mempersiapkan dokumen teks yang tidak terstruktur menjadi data yang terstruktur yang siap digunakan untuk proses selanjutnya. Tahapan *Text Preprocessing* dalam penelitian ini meliputi:

1. *Convert Negation*

Emoticon adalah kata gabungan dari "*emotion*" dan "*icon*" yang berarti icon yang digunakan untuk mengekspresikan emosi sebuah pernyataan tertulis, dan bisa mengubah serta meningkatkan interpretasi terhadap tulisan tersebut.

2. *Cleansing*

Tahap ini akan menghapus semua karakter selain alfabetis dengan tujuan untuk mengurangi *nois*. Sebagaimana diketahui bahwa *emoticon* ini disimbolkan dengan kombinasi karakter khusus dan juga angka, sehingga *emoticon* ini tidak terhapus. Selain karakter khusus, , URL, *hashtag (#)*, *username* (@username), tanda koma(,), tanda titik(.), tanda seru (!), tanda titik koma (;), tanda titik dua (:), tanda hubung (-), tanda elipsis (…), tanda tanya (?), tanda kurung ((..)), tanda kurung siku ({..}), tanda petik ("..."), tanda petik tunggal ('..'), tanda garis miring (/) dan (\), dan tanda penyingkat (') akan dihilangkan.

3. *Case Folding*

Merupakan tahapan merubah semua masukkan huruf menjadi huruf kecil semua (*lower case*). Karena sistem yang akan dibangun menggunakan basaha pemrograman java, maka disamakan dahulu kedalam bentuk yang sama, dalam hal ini menjadi huruf kecil semua.

4. *Convert Negation*

*Convert negation* dilakukan jika terdapat kata negasi sebelum kata yang bernilai positif, maka kata tersebut akan diubah nilainya menjadi negatif dan begitupun sebaliknya. Kata-kata yang bersifat negasi seperti "bukan", "bkn", "tidak", "enggak", "g", "gak","tidak", "tdk", "enggak", "engga", "ga", "gk", "jangan", "jgn", "nggak", "tak" dan "gak".

5. *Tokenizing*

*Tokenizing* adalah tahap pemotongan dokumen teks bedasarkan tiap kata yang menyusunnya. Potongan kata tersebut disebut dengan token atau *term*. Pada tahap ini akan dilakukan pengecekan tweets dari karakter pertama sampai karakter terakhir.

6. *Filtering*

*Filtering* berperan untuk membuang kata-kata yang sering muncul dan bersifat umum, kurang menunjukan relevansinya dengan teks. Proses ini akan menghilangkan kata-kata yang sering muncul namun tidak memiliki pengaruhapapun dalam ekstraksi sentimen suatu *tweet*.

7. *Stemming*

Tahap *Stemming* adalah tahap menacari *root* kata dari setiap kata hasil *filtering*. Kata-kata yang muncul didalam dokumen sering kali mengandung imbuhan. Oleh karena itu, setiap kata yang tersisa dari proses hasil tahapan *filtering* dibentuk kedalam

kata dasar dengan cara menghilangkan imbuhannya.

Algoritma *Stemming* yang digunakan pada penelitian ini yaitu algoritma *Confix Stripping Stemmer*. Algoritma ini menambahkan suatu algoritma tambahan untuk mengatasi kesalahan pemenggalan akhiran yang seharusnya tidak dilakukan.

### 1.5. Confix Stripping Stemmer

*Confix stripping Stemmer* adalah metode stemming pada Bahasa Indonesia yang diperkenalkan oleh Jelita Asian yang merupakan pengembangan dari metode stemming yang dibuat oleh Nazief dan Adriani (1996). Kata-kata yang muncul didalam dokumen sering kali mengandung imbuhan. Oleh karena itu, setiap kata yang tersisa dari proses hasil tahapan *filtering* dibentuk kedalam kata dasar dengan cara menghilangkan imbuhannya. Pada dasarnya, algoritma ini mengelompokkan imbuhan ke dalam beberapa kategori sebagai berikut:

1. *Inflection Suffixes* yakni kelompok-kelompok akhiran yang tidak mengubah bentuk kata dasar. Kelompok ini dapat dibagi menjadi dua:
   a. *Particle* (P) atau partikel, termasuk di dalamnya adalah partikel "-lah", "-kah", "-tah", dan "-pun".
   b. *Possessive Pronoun* (PP) atau kata ganti kepunyaan, termasuk di dalamnya adalah "-ku" , "-mu", dan "-nya".

2. *Derivation Suffixes* (DS) yakni kumpulan akhiran yang secara langsung dapat ditambahkan pada kata dasar. Termasuk di dalam tipe ini adalah akhiran "-i", "-kan", dan "-an".

3. Tahapan *Derivation Prefixes* (DP) yakni kumpulan awalan yang dapat langsung diberikan pada kata dasar murni, atau pada kata dasar yang sudah mendapatkan penambahan sampai dengan 2 awalan. Termasuk di dalamnya adalah awalan yang dapat bermorfologi ("me-", "be-", "pe-", dan "te-") dan awalan yang tidak bermorfologi ("di-", "ke-" dan "se-"). Berdasarkan pengklasifikasian imbuhan-imbuhan tersebut, maka bentuk kata dalam bahasa Indonesia dapat dimodelkan sebagai berikut:

**[DP+[DP + [DP+]]] Kata Dasar [[+DS][+PP][+P]]** (1)

Dengan batasan-batasan sebagai berikut :

a. Tidak semua kombinasi imbuhan diperbolehkan. Kombinasi imbuhan yang dilarang dapat dilihat pada Tabel 1.
b. Penggunaan imbuhan yang sama secara berulang tidak diperkenankan.
c. Jika suatu kata hanya terdiri dari satu atau dua huruf, maka proses stemming tidak dilakukan.
d. Penambahan suatu awalan tertentu dapat mengubah bentuk asli kata dasar, ataupun awalan yang telah diberikan sebelumnya pada kata dasar bersangkutan (bermorfologi).

Tabel 1 Kombinasi Awalan dan Akhiran

| Awalan ( Prefiks) | Akhiran (Suffiks) |
|---|---|
| be- | -i |
| di- | -an |
| ke- | -i –kan |
| me- | -an |
| se- | -i –kan |
| te- | -an |

Tabel 2 Aturan Peluruhan Kata Dasar

| Aturan | Bentuk Awalan | Peluruhan |
|---|---|---|
| 1 | berV… | ber-V… \| be-rV… |
| 2 | belajar… | bel-ajar |
| 3 | beC$_1$erC$_2$… | be-C$_1$erC$_2$…dimana C$_1$!={'r'\|'l'} |
| 4 | terV… | ter-V…\|te-rV… |
| 5 | terCer… | ter-Cer…dimana C!='r' |
| 6 | teC$_1$erC$_2$ | te-C$_1$erC$_2$…dimana C$_1$!='r' |
| 7 | me{l\|r\|w\|y}V… | me-{l\|r\|w\|y}V… |
| 8 | mem{b\|f\|v}… | mem-{b\|f\|v}… |
| 9 | mempe… | mem-pe… |
| 10 | mem{rV\|V}… | me-m{rV\|V}…\|me-p{rV\|V}… |
| 11 | men{c\|d\|j\|z}… | men-{c\|d\|j\|z}… |
| 12 | menV… | me-nV…\|me-tV… |
| 13 | meng{g\|h\|q\|k }… | meng-{g\|h\|q\|k}… |
| 14 | mengV… | meng-V…\|meng-kV… |
| 15 | mengeC | menge-C |
| 16 | menyV… | me-ny…\| meny-sV… |
| 17 | mempV… | mem-pV… |
| 18 | pe{w\|y}V… | pe-{w\|y}V… |
| 19 | perV… | per-V…\|pe-rV… |
| 20 | pem{b\|f\|v}… | pem-{b\|f\|v}… |
| 21 | pem{rV\|V}… | pe-m{rV\|V}…\|pe-p{rV\|V} |
| 22 | pen{c\|d\|j\|z}… | pen-{c\|d\|j\|z}… |
| 23 | penV… | pe-nV…\|pe-tV… |
| 24 | peng{g\|h\|q} | peng-{g\|h\|q} |
| 25 | pengV | peng-V\|peng-kV |
| 26 | penyV… | pe-nya\|peny-sV |
| 27 | pelV.. | pe-lV…; kecuali untuk kata "pelajar" menjadi "ajar" |
| 28 | peCP | pe-CP…dimana C!={r\|w\|y\|l\|m\|n} dan P!='er' |
| 29 | perCerV | Per-CerV… dimana C!={r\|w\|y\|l\|m\|n} |

Algoritma *CS stemmer* bekerja sebagai berikut:

1. Kata yang hendak di-*stemming* dicari terlebih dahulu pada kamus. Jika ditemukan, berarti kata

tersebut adalah kata dasar, jika tidak maka langkah 2 dilakukan.

2. Cek *rule precedence*. Apabila suatu kata memiliki pasangan awalan-akhiran "be-lah", "be-an", "me-i", "di-i", "pe-i", atau "te-i" maka langkah *stemming* selanjutnya adalah (5, 6, 3, 4, 7). Apabila kata tidak memiliki pasangan awalan-akhiran tersebut, langkah *stemming* berjalan normal (3, 4, 5, 6, 7).

3. Hilangkan *inflectional particle P* ("-lah", "-kah", "-tah", "-pun") dan kata ganti kepunyaan atau *possessive pronoun PP* ("-ku", "-mu", "-nya").

4. Hilangkan *Derivation Suffixes DS* ("-i", "-kan", atau "-an").

5. Hilangkan *Derivational Prefixes DP* {"di-","ke-","se-","me-","be-","pe", "te-"} dengan iterasi maksimum adalah 3 kali:
   a. Langkah 5 ini berhenti jika :
      1. Terjadi kombinasi imbuhan terlarang
      2. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
      3. Tiga awalan telah dihilangkan.
   b. Identifikasikan tipe awalan dan hilangkan. Awalan ada dua tipe:
      1. Standar: "di-", "ke-", "se-" yang dapat langsung dihilangkan dari kata.
      2. Kompleks: "me-", "be-", "pe", "te-" adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel II-14 untuk mendapatkan pemenggalan yang tepat.
   c. arti kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 5 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

6. Apabila setelah langkah 5 kata dasar masih belum ditemukan, maka proses *recoding* dilakukan dengan mengacu pada aturan pada Tabel II-14. *Recoding* dilakukan dengan menambahkan karakter *recoding* di awal kata yang dipenggal. Pada Tabel II-14, karakter *recoding* adalah karakter setelah tanda hubung ('-') dan terkadang berada sebelum tanda kurung. Sebagai contoh, pada kata "menangkap" (aturan 15), setelah dipenggal menjadi "nangkap". Karena tidak valid, maka *recoding* dilakukan dan menghasilkan kata "tangkap". Perlu diperhatikan bahwa aturan ke-22 tidak ditemukan dalam tesis Jelita Asian

7. Jika semua langkah gagal, maka input kata yang diuji pada algoritma ini dianggap sebagai kata dasar.

Apabila pada kata yang hendak di-*stemming* ditemukan tanda hubung ('-'), maka kemungkinan kata yang hendak di-*stemming* adalah kata ulang. *Stemming* untuk kata ulang dilakukan dengan memecah kata menjadi dua bagian yakni bagian kiri dan kanan (berdasarkan posisi tanda hubung '-') dan lakukan stemming (langkah 1-7) pada dua kata tersebut. Apabila hasil *stemming* keduanya sama, maka kata dasar berhasil didapatkan.

### 1.6. Pembobotan (Term Weighting)

*Term Weighting* adalah teknik pembobotan pada setiap *term* atau kata. Tahapan ini sebagian besar teknik pembobotan pada *text mining* menggunakan TF.IDF. TF.IDF menerapkan pembobotan kombinasi keduanya berupa perkalian bobot lokal(*term frequency)* dan bobot global (*global inverse document frequency*). [13] Metode TF-IDF dapat dirumuskan sebagai berikut:

$$idf = \log\left(\frac{N}{df}\right) \quad (2)$$

Dimana :
N = Banyaknya Data
df = *document frequency*

$$\mathbf{w(t, d) = tf\ (t, d) * IDF} \quad (3)$$

Dimana :
tf = *term frequency*
IDF = *Inverse Document Frequency*
d = dokumen ke-d
t = kata ke-t dari kata kunci
w(t,d) = bobot dokumen ke-d terhadap kata ke-t

### 1.7. Improved K-Nearest Neighbor

Penentuan *k-values* yang tepat diperlukan agar didapatkan akurasi yang tinggi dalam proses kategorisasi dokumen uji. Algoritma *Improved k-Nearest Neighbors* 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 tiap-tiap 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.

Untuk menghitung similaritas antara dua dokumen menggunakan metode Cosine Similarity (CosSim). 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*. [13] Rumus yang digunakan untuk menghitung *cosine similarity* adalah sebagai berikut:

$$Cos\left(\theta_{QD}\right) = \frac{\sum_{i=1}^{n} Q_i \cdot D_i}{\sqrt{\sum_{i=1}^{n}(Q_i)^2} \cdot \sqrt{\sum_{i=1}^{n}(D_i)^2}} \quad (4)$$

Dimana :
$Cos(\theta_{QD})$ = Kemiripan dokumen Q terhadap D
Q = Data Uji
D = Data Latih
n = Banyaknya data

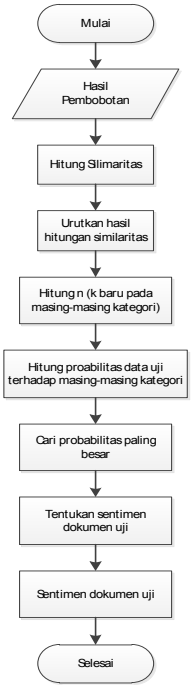
Perhitungan penetapan k-values pada algoritma *Improved k-Nearest Neighbor* dilakukan dengan menggunakan persamaan (II-4), dengan terlebih dahulu mengurutkan secara menurun hasil perhitungan similaritas pada setiap kategori.

Selanjutnya pada algoritma *Improved k-Nearest Neighbor*, *k-values* yang baru disebut dengan n. Persamaan (II-4) menjelaskan mengenai proporsi penetapan *k-values* (n) pada setiap kategori.

$$n = \left[\frac{k * N(c_m)}{maks\{N(c_m) \mid j=1.....N_c\}}\right] \quad (5)$$

Dimana :
n = *k-values* baru
k = *k-values* yang ditetapkan
$N(c_m)$ = Jumlah dokumen latih di kategori / kategori m
$maks\{N(c_m) \mid j=1.....N_c\}$ = 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.




Gambar 1 *Flowchart Improved K-Nearest Neighbor*

### 1.8. 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). 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 (6).

Tabel 3 Tabel Kontingensi

| | **Predicted Class** | | |
|---|---|---|---|
| **Actual Class** | | + | - |
| | + | TP | FN |
| | - | FP | TN |

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Dengan menjabarkan tabel 3 diatas maka kita bisa mendapatkan persamaan (6) dan (7) 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).

Kombinasi precision dan recall biasa dikombinasikan sebagai harmonic mean, biasa disebut F-measure yang mana dapat di formulasikan seperti persamaan (8).

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (8)$$

*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.

## 2. ISI PENELITIAN

### 2.1. Analisis Masalah

Permasalahan dari penelitian ini adalah bagaimana mengklasifikasi informasi dari media sosial khususnya twitter yang berisikan opini konsumen terhadap Telkom IndiHome kedalam dua kelas yaitu negatif atau positif. Kemudian hasil dari klasifikasi tersebut disajikan kedalam bentuk grafik

### 2.2. Analisis Sistem Yang Akan Dibangun

Sistem yang akan dibangun pada penelitian ini adalah aplikasi yang digunakan untuk analisis sentimen terhadap Telkom IndiHome. Dengan demikian alur atau proses-proses dari sistem yang akan dibangun adalah sebagai berikut:

1. Proses pengambilan data
   Proses pengambilan data berupa data uji dan data latih. Data yang dibutuhkan diambil dari media sosial twitter
2. Proses *Preprocessing*
   Data latih dan data uji akan melalui proses *text preprocessing* yang merupakan tahap awal dari *text mining*. *Text processing* ini bertujuan untuk mempersiapkan dokumen teks yang tidak terstruktur menjadi data yang terstruktur yang siap digunakan untuk proses selanjutnya.
3. Proses Pembobotan
   Melalui proses *preprocessing* data yang didapat akan melalui tahap pembobotan
4. Proses Klasifikasi
   Tahapan proses klasifikasi ini bertujuan untuk membagi data yang masuk kedalam *class-class* yang telah ditentukan sehingga menghasilkan hasil sentimen analisis.

### 2.3. Analisis Pengambilan Data

Data *Tweet* dalam penelitian ini dperoleh dengan memanfaatkan API yg disediakanoleh Twitter. Dengan memanfaatkan API tersebut dibangunlah sebuah aplikasi untuk mengambil data *Tweet* tersebut dari Twitter kemudian disimpan ke dalam Database.

Pada saat pengumpulan data, menggunakan Twitter AP I *Search*, kemudian memasukkan *keyword-keyword* yang berhubungan dengan produk Telkom Indihome yang dikombinasikan dengan kata-kata sentimen

Tabel 4. Contoh kata-kata sentimen

| | |
|---|---|
| Kata Negatif | Lelet, lemot, lambat, parah, jelek, buruk, kecewa, payah |
| Kata Positif | Bisa, bagus, lancar, cepat, cepet, mantap, mantep, baik, kenceng, kencang, lancar, senang, canggih |
| Kata Netral | ikutan, selamat, kompetisi, dapatkan, gratis, kesempatan, info, nikmati |
| Emoticon Negatif | :-(, :(, =(, ;( |
| Emoticon Positif | :-), :), =), :D |
| Nama produk | Indihome, speedy, useetv, telkomcare |

Tabel 5 Contoh *Tweet*

| Tweet |
|---|
| Koneksinya speedy galau nih :( gimana ini @TelkomCare |
| @TelkomCare Jaringan WIFI speedy kenapa lemot sudah dua hari, mengganggu aktivitas saya. |
| Asik nih UseeTV bisa playback acara hari kemarin :D |
| Mau hujan pun, streaming youtube lancar, speedy emang mantap :) |
| Download film di ganool.com pake speedy cepet banget. |
| Gak Asik dari kemarin koneksi speedy lelet parah :( #speedyLELET |
| @TelkomPromo Raih silaturahmi 100% di hari yg fitri dgn GRATIS nelpon 1000 menit lokal/Interlokal indihome. Info: http://indihome.co.id . |
| Mau dimana2 jaringan speedy selalu lancar :') |
| @TelkomCare Speedy koneksi tidak bagus. gak bisa dengan restart modem saja. |
| Mau dimana2 jaringan speedy selalu lancar :') |

### 2.4. Analisis Pembobotan (*Term Weighting*)

Tahap ini merupakan tahap pembobotan, yang dilakukan setelah proses preprocessing. Metode pembobotan yang digunakan adalah metode TF.IDF. Pada metode ini *Term Frequency* (TF) akan dikalikan dengan *Inverse Document Frequency* (IDF). Rumus yang digunakan untuk menyatakan bobor (w) masing-masing dokumen terhadap dokumen terhadap kata kunci adalah pada persamaan (II-2) dan (II-3).

Tabel 6 Data Latih Yang Diketahui

| Kode | Data Latih | |
|---|---|---|
| | *Data Latih* | *Kelas* |
| D1 | [koneksi] [galau] [kecewa] | Negatif |
| D2 | [jaringan] [wifi] [lelet] [ganggu] | Negatif |
| D3 | [download] [film] [lancar] [banget] [senang] | Positif |
| D4 | [streaming] [youtube] [lancar] [mantap] [senang] | Positif |
| D5 | [koneksi] [selalu] [lancar] [senang] | Positif |

Tabel 7 Data Uji Yang Akan Dianalisis

| Kode | Data Uji | |
|---|---|---|
| | *Data Uji* | *Kelas* |
| D6 | [gak_asik] [koneksi] [lelet] [parah] [kecewa] [speedylelet] | ? |

Berdasarkan Tabel 6 dan Tabel 7, D1 sampai D6 merupakan data yang akan kita uji bobot dokumennya. D1 sampai D5 merupakan data yang sudah diketahui kelasnya, sedangkan D6 data yang belum diketahui kelasnya dan yang akan diuji. Untuk menentukan masuk ke kelas manakan D6. Pertama hitung bobot setiap *term.*

Tabel 8. Penerapan Contoh Kasus Tahapan *Term Weighting*

| Term | TF | | | | | | | IDF | Wdt=TF.IDF | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | D6 | D1 | D2 | D3 | D4 | D5 | DF | LOG (n/df) | D6 | D1 | D2 | D3 | D4 | D5 |
| gak_asik | 1 | | | | | | 1 | 0.778 | 0.778 | 0 | 0 | 0 | 0 | 0 |
| Koneksi | 1 | 1 | | | | 1 | 3 | 0.301 | 0.301 | 0.301 | 0 | 0 | 0 | 0.301 |
| Lelet | 1 | | 1 | | | | 2 | 0.477 | 0.477 | 0 | 0.477 | 0 | 0 | 0 |
| Parah | 1 | | | | | | 1 | 0.778 | 0.778 | 0 | 0 | 0 | 0 | 0 |
| Kecewa | 1 | 1 | | | | | 2 | 0.477 | 0.477 | 0.477 | 0 | 0 | 0 | 0 |
| speedylelet | 1 | | | | | | 1 | 0.778 | 0.778 | 0 | 0 | 0 | 0 | 0 |
| Galau | | 1 | | | | | 1 | 0.778 | 0 | 0.778 | 0 | 0 | 0 | 0 |
| Jaringan | | | 1 | | | | 1 | 0.778 | 0 | 0 | 0.778 | 0 | 0 | 0 |
| Wifi | | | 1 | | | | 1 | 0.778 | 0 | 0 | 0.778 | 0 | 0 | 0 |
| Ganggu | | | 1 | | | | 1 | 0.778 | 0 | 0 | 0.778 | 0 | 0 | 0 |
| download | | | | 1 | | | 1 | 0.778 | 0 | 0 | 0 | 0.778 | 0 | 0 |
| film | | | | 1 | | | 1 | 0.778 | 0 | 0 | 0 | 0.778 | 0 | 0 |
| Lancer | | | | 1 | 1 | 1 | 3 | 0.301 | 0 | 0 | 0 | 0.301 | 0.301 | 0.301 |
| Banget | | | | 1 | | | 1 | 0.778 | 0 | 0 | 0 | 0.778 | 0 | 0 |
| Senang | | | | 1 | 1 | 1 | 3 | 0.301 | 0 | 0 | 0 | 0.301 | 0.301 | 0.301 |
| Streaming | | | | | 1 | | 1 | 0.778 | 0 | 0 | 0 | 0 | 0.778 | 0 |
| Youtube | | | | | 1 | | 1 | 0.778 | 0 | 0 | 0 | 0 | 0.778 | 0 |
| Mantap | | | | | 1 | | 1 | 0.778 | 0 | 0 | 0 | 0 | 0.778 | 0 |
| Selalu | | | | | | 1 | 1 | 0.778 | 0 | 0 | 0 | 0 | 0 | 0.778 |
| | | | | | | | | | 3.59 | 1.56 | 2.81 | 2.94 | 2.93 | 1.68 |

## 2.5. Analisis Penerapan *Improve K-Nearest Neighbor*

Setelah melalui proses pembobotan dokumen akan melalui tahap pengklasifikasian, pada proses ini akan digunakan algoritma *improve k-nearest neighbor*. Adapun langkah langkahnya adalah sebagai berikut:

Menghitung similaritas antara dua dokumen menggunakan metode *Cosine Similarity* (*CosSim).* Hitung kemiripan vektor dokumen D6 dengan setiap dokumen yang telah terklasifikasi (D1, D2, D3, D4, dan D5). Kemiripan antar dokumen dapat menggunakan *Cosine Similarity*. Rumusnya adalah sebagai berikut:

$$Cos\left(\theta_{QD}\right) = \frac{\sum_{i=1}^{n} Q_i \cdot D_i}{\sqrt{\sum_{i=1}^{n}(Q_i)^2}.\sqrt{\sum_{i=1}^{n}(D_i)^2}} \qquad (4)$$

Di mana :

$Cos(\theta_{QD})$ = Kemiripan dokumen Q terhadap D

Q = Data Uji

D = Data Latih

n = Banyaknya data

Untuk menyelesaikan persamaan (4) dapat dibagi menjadi dua langkah berikut:

1. Hitung hasil perkalian skalar antara D6 dan D5 dokumen yang telah terklasifikasi. Hasilnya perkalian dari setiap dokumen dengan D6 dijumlahkan dengan menggunakan rumus persamaan (4) bagian atas $\sum_{i=1}^{n} Q_i \cdot D_i$
2. Hitung panjang setiap dokumen, termasuk D6. Caranya, kuadratkan bobot setiap *term* dalam setiap dokumen, jumlahkan nilai kuadrat tersebut dan kemudian akarkan dengan menggunakan rumus persamaan (4) bagian bawah $\sqrt{\sum_{i=1}^{n}(Q_i)^2}.\sqrt{\sum_{i=1}^{n}(D_i)^2}$

Sisi kiri (WD6*WDi) pada Tabel 9 mewakili langkah pertama dimana WD6 itu W dari pembobotan persamaan (3), WDi Data latih pada saat pembobotan (3) dan sisi kanan (panjang vektor) memperlihatkan langkah kedua.

Tabel 9 Penyelasaian *Cosine Similarity*

| WD6*Wdi | | | | | Panjang Vektor | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| D6XD1 | D6XD2 | D6XD3 | D6XD4 | D6XD5 | D6 | D1 | D2 | D3 | D4 | D5 |
| 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 | 0 | 0 | 0 |
| 0.0906 | 0 | 0 | 0 | 0.091 | 0.091 | 0.091 | 0 | 0 | 0 | 0.091 |
| 0 | 0.2276 | 0 | 0 | 0 | 0.228 | 0 | 0.228 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 | 0 | 0 | 0 |
| 0.2276 | 0 | 0 | 0 | 0 | 0.228 | 0.228 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.091 | 0.091 | 0.091 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.091 | 0.091 | 0.091 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.606 |
| 0.31826 | 0.22764 | 0 | 0 | 0.0906 | 2.3625 | 0.9238 | 2.0442 | 1.9978 | 1.9978 | 0.8774 |
| | | | | | 1.537 | 0.961 | 1.43 | 1.413 | 1.413 | 0.937 |
| | | | | | | **0.215** | **0.104** | **0** | **0** | **0.063** |

Dari Perhitungan Tabel 9 diketahui nilai *cosine similiarity* dari D1,D2,D3,D4, dan D5 adalah:

Tabel 10 Nilai *Cosine Similiarity*

| D1 | D2 | D3 | D4 | D5 |
|---|---|---|---|---|
| 0.215 | 0.104 | 0 | 0 | 0.063 |

Langkah selanjutnya adalah urutkan tingkat kemiripan dari data tersebut diperoleh:

Tabel 11 Urutan Tingkat Kemiripan

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| D1 | D2 | D5 | D3 | D4 |

(II-4)

Selanjutnya pada algoritma Improved k-Nearest Neighbor, *k-values* yang baru disebut dengan n. Persamaan (5) menjelaskan mengenai proporsi penetapan *k-values* (n) pada setiap kategori.

$$n = \left[\frac{k * N(c_m)}{maks\{N(c_m) \mid j=1.....N_c\}}\right] \quad (5)$$

Dimana :
n = *k-values* baru
k = *k-values* yang ditetapkan
$N(c_m)$ = Jumlah dokumen latih di kategori / kategori m
$maks\{N(c_m) \mid j=1.....N_c\}$ = jumlah dokumen latih terbanyak pada semua kategori

Hasil pertitungan nilai n :

Tabel 12 Jumlah Data Latih

| Data Latih | | |
|---|---|---|
| Positif | Negatif | Jumlah |
| 3 | 2 | 5 |

$$n = \frac{3 \times 3}{3} = \frac{9}{3} = 3$$

$$n = \frac{3 \times 2}{3} = \frac{6}{3} = 2$$

Tabel 13 Hasil Nilai n (k-baru)

| | n (k baru) | |
|---|---|---|
| Nilai k | n Positif | n Negatif |
| 3 | 3 | 2 |

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.

Setelah diketahui urutan tingkat kemiripannya Ambil sebanyak *k-values* baru (n) yang paling tinggi tingkat kemiripannya dengan D6 dan tentukan kelas dari D6. Hasilnya :

Tabel 14 Hasil Akhir Urutan Tingkat Kemiripan

| 1 | 2 | 3 |
|---|---|---|
| D1 | D2 | D5 |
| 0.215 | 0.104 | 0.063 |
| Negatif | Negatif | Positif |

Terakhir, adalah tentukan kelas D6 berdasarkan kelas yang muncul paling banyak. Karena kelas yang muncul adalah kelas mayoritas negatif, maka D6 masuk ke kelas negatif.

Jika terjadi kasus khusus di mana nilai K yang diambil mempunyai nilai genap dan kelas yang muncul berjumlah sama, maka dokumen uji dimasukan ke kelas yang memiliki nilai kemiripan paling tinggi

### 2.6. Pengujian Sistem

Pengujian Metode merupakan suatu proses pengujian mengenai algoritma klasifikasi. Tujuan dari pengujian ini untuk mengetahui ada tidaknya kesalahan pada saat mengimplementasikan logika algoritma *improved k-nearest neighbor*.

Pengujian akurasi klasifikasi *tweets* dilakukan untuk mengetahui tingkat akurasi klasifikasi *tweets* yang dilakukan secara manual dengan klasifikasi *tweets* yang dilakukan oleh sistem dengan menggunakan *Improved K-Nearest Neighbor*. Pengujian dilakukan dengan menggunakan *confusion matrix* yaitu sebuah matrik dari prediksi yang akan dibandingkan dengan kelas yang asli dari data masukkan. Pengujian dilakukan menggunakan 20 *sample tweets*. untuk skenario lebih jelasnya akan dipaparkan pada tabel berikut:

Tabel 15 *Sample* Pengujian klasifikasi *tweet*

| No | Tweet | Sentimen | |
|---|---|---|---|
| | | Kasifikasi Manual | Klasifikasi Sistem |
| 1 | @TelkomCare min saya indihome 10 mbps fiber kok lemot ya ? tiba tiba jadi ke 2 mbps ? kenapa mmin ? ping pun jadi 70 , | N | N |
| 2 | Dan janji telemarketer 147 buat naikin kecepatan internet dgn ganti pake indihome jg ga kejadian. Di test masi lemot juga ni wifi □ | N | N |
| 3 | @TelkomCare indihome saya bermasalah terus beberapa hari ini, sinyal lambat dan koneksi internet sering putus. | N | P |
| 4 | @TelkomCare Mohon segera perbaiki koneksi Indihome kami, sudah satu minggu lebih tidak bisa digunakan. | N | P |
| 5 | @wifi_id min saya mau register wifi.id tapi bingung no internet indihome saya langganan spedy ada koneksi wifi.id nya penjelasannya | P | N |
| 6 | Ga bisa pake periscope dgn koneksi indihome, pake 3G malah bisa... @TelkomCare | N | N |
| 7 | @TelkomCare indihome seharian jelek nih.. kadang nyambung kadang ngga.. lampu indikator internet jg merah mulu.. | N | N |
| 8 | @ardynp saya pernah pake @transvision_ind sebelum pindah ke indihome @TelkomPromo @UseeTV_Cable , oke juga. | P | P |
| 9 | Tadi blm nonton . Coba ILK dulu ah di Tipiiii. Pakai fitur playback @UseeTVcom nya IndiHome.#IamIndihome #KipasSakti @IndihomeFiber | P | N |
| 10 | @TelkomCare tolong dong dicek kan indihome gw, dari sore sampe sekarang gangguan nauzubilah, -_- daerah pontianak | N | N |
| 11 | @kih00ns Iya sama2 Kak, semoga lancar terus dan tidak ada gangguan lagi. ^AY | P | P |
| 12 | @TelkomCare internetnya lancar jaya | P | P |
| 13 | @TelkomSpeedy ga **bagus** penanganan gangguannya, gmn mau masuk ke market **indohome**?! Haha nanganin gangguan rusak modem aja sampe 4hr ga jelas! | N | N |
| 14 | Penasaran sama speedy. Uda bayar tapi koneksi antara lemot atau putus nyambung. Mau banting HP, sayang Hp blm ada sebulan umumta :'( | N | N |
| 15 | Sedikit kecewa dengan pelayanan telkom speedy | N | N |
| 16 | @Adityaa_Prtm IndiHome masup di daerah lo nggak? temen2 gw pada bilang **bagus** sih | P | P |
| 17 | Layanan Indihome dari @TelkomIndonesia sangat bagus, puas deh :) | P | P |
| 18 | Sejauh ini puas dengan @TelkomPromo yaitu indihome. Internet cepat plus tv yang mumpuni. Masakinian banget dehh.. Makasih telkom :) | P | P |
| 19 | Sejauh ini puas dengan @TelkomPromo yaitu indihome. Internet cepat plus tv yang mumpuni. Masakinian banget dehh.. Makasih telkom :) | P | p |
| 20 | @TelkomCare kalau telepon tdk tau km tidak dipasang, tp usee tv msh berjalan dgn lancar | P | P |

Ket : P (Positif), N (Negatif)

Berikut tabel dari *confuion matrix* : 0,64

Tabel 16 *Confusion Matrix*

| | **Positif** | **Negatif** |
|---|---|---|
| **Positif** | 8 | 2 |
| **Negatif** | 2 | 8 |

Setelah sistem melakukan klasifikasi, kemudian hitung precision, recall dan akurasinya berdasarkan persamaan (6) dan (7)

$$Precision = \frac{8}{8+2} = 0{,}80$$

$$Recall = \frac{8}{8+2} = 0{,}80$$

$$F - Measure = 2 * \frac{0{,}80 * 0{,}80}{0{,}80 + 0{,}80} = 0{,}80$$

Data pengujian yang digunakan pada Tabel 15 menggunakan sample *tweet* sebanyak 20 *tweet*. Dari pengujian yang telah dilakukan, dapat diketahui bahwa terdapat beberapa factor yang mempengaruhi ketepatan analisis sentimen dengan menggunakan metode *Improved K-Nearest Neighbor*. Berdasarkan pengujian *Precision, Recall* dan *F-Measure*, didapatkan hasil *F-Measure* klasifikasi tweets dari sistem analisis sentimen dengan menggunakan *Improved K-Nearest Neighbor* sebesar 80% dengan *precision* sebesar 80% dan recall sebesar 80%.

### 2.7. Implementasi Antarmuka

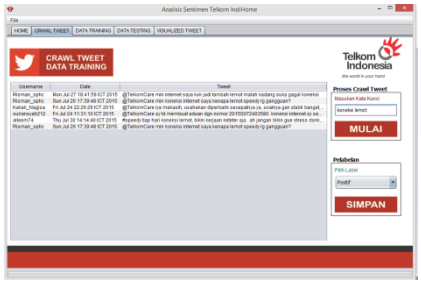
Berikut tampilan antarmuka yang ada pada aplikasi ini.

**1. Antarmuka *Home***



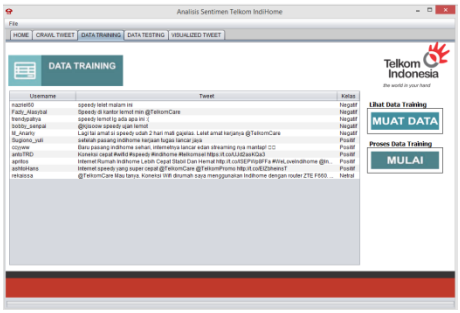

Gambar 2 Tampilan Antarmuka *Home*

**2. Antarmuka *Crawl Tweet***



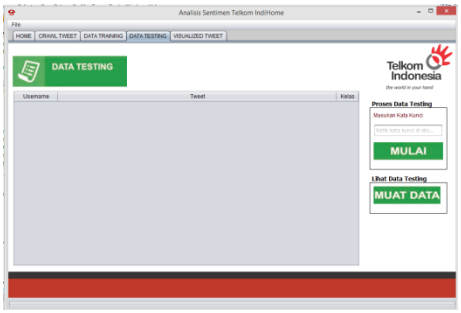

Gambar 3 Tampilan Antarmuka *Crawl Tweet*

**3. Antarmuka Data Training**



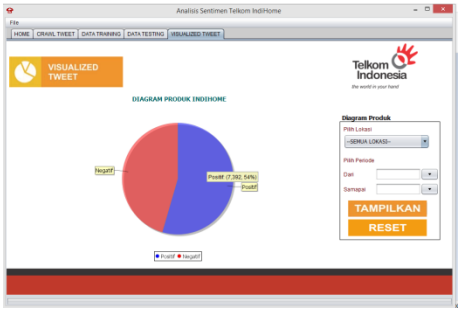

Gambar 4 Tampilan Antarmuka *Data Training*

**4. Antarmuka Data Testing**




Gambar 4 Tampilan Antarmuka *Data Testing*

**5. Antarmuka Visualized Tweet**




Gambar 4 Tampilan Antarmuka *Visualized Tweet*

## 3. PENUTUP

### 3.1. Kesimpulan

Dari hasil penelitian yang telah dilakukan terlihat bahwa algoritma *Improved K-Nearest Neighbor* dapat mengklasifikasikan suatu opini yang berupa *tweet* ke dalam dua kelas yaitu positif dan negatif dengan akurat. Tingkat keakurasian dari pengklasifikasian tersebut sangat dipengaruhi oleh proses *training*. Sehingga dapat disimpulkan dari hasil pengklasifikasian yang disajikan dalam bentuk grafik di visualized tweet dapat terlihat dengan jelas informasi sentimen publik terhadap suatu produk Indihome dan dapat dijadikan sebagai bahan evaluasi Telkom IndiHome agar dapat lebih meningkatkan kualitas layanannya sehingga dapat memperbaiki dan menentukan langkah bisnis selanjutnya yang lebih baik lagi.

### 3.2. Saran

Adapun saran dari penelitian ini adalah sebagai berikut:

1. Dibutuhkannya penelitian lebih lanjut atau pengembangan untuk penelitian analisis sentimen menggunakan metode pengklasifikasian lain seperti *Weighted K-Nearest Neighbor* atau menggabungkan metode lain dengan metode metode *Improved K-Nearest Neighbor* yang bisa lebih baik dari metode *Improved K-Nearest Neighbor* agar didapat hasil pengklasifikasian analisis sentimen yang lebih baik dan lebih akurat.
2. Pada penelitian selanjutnya diharapakan dapat mengenali kalimat sarkasme seperti “koneksi indihome lancaaarr sekali, sampai browsing aja susah :)”.
3. Dalam penelitian ini ketika melakukan pembobotan, sistem menghitung kemiripan berdasarkan frekuensi kemunculan kata, sehingga untuk mendapatkan hasil yang optimal sebaiknya digunakan sistem yang dapat mengecek kata yang bersinonim.

## DAFTAR PUSTAKA

[1] https://dailysocial.net/post/kemenkominfo-targetkan-pengguna-internet-di-indonesia-tahun-2015-capai-150-juta-orang
[2] http://tekno.liputan6.com/read/2164377/pengguna-internet-indonesia-kuasai-media-sosial-di-2015
[3] http://tekno.liputan6.com/read/2164377/pengguna-internet-indonesia-kuasai-media-sosial-di-2015?p=1
[4] Iwan Arif, Text Mining http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm/6Text%20Mining.pdf
[6] B. P. a. L. Lee, "Opinion Mining and Sentiment Analysis, Foundations and Trends in Information Retrieval," vol. 2, no. 1-2, pp. 1-135, 2008.
[7] Fahrur Rozi Imam, "Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi", 2012
[8] Yusuf Nur Muhammad dan Santika D. Diaz "ANALISIS SENTIMEN PADA DOKUMEN BERBAHASA INDONESIA DENGAN PENDEKATAN SUPPORT VECTOR MACHINE" 2011
[9] Raymon J. Mooney. CS, Machine Learning Text Categorozation, 2006
[10] L. Vogel, "Java Regex - Tutorial, Vogella,," 14 Januari 2014.
[11] Sunni Ismail "Analisis Sentimen dan Ekstraksi Topik PenentuSentimen pada Opini Terhadap Tokoh Publik" volume 1, nomor 2, 2012
[12] Utomo manalu Boy, "Analisis Sentimen Pada Twitter Menggunakan teks mining" 2014
[13] Arfianda Putri Prima "IMPLEMENTASI METODE IMPROVED K-NEAREST NEIGHBOR PADA ANALISIS SENTIMEN TWITTER BERBAHASA INDONESIA"
[14] Kroenke M. David "Database Processing Jilid 1" edisi 9, 2005
[15] Prodase Labolarotium, "Object-Oriented Programming Module" 2013/2014
[16] Dwiyoga Tahitoe Andita “Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming”,
[17] Ngesti Waluyo Catur, “Confix Stripping Stemmer”, 2012.