

**SISTEM KLASIFIKASI OPINI PENGGUNA MASKAPAI  
PENERBANGAN DI INDONESIA PADA JEJARING  
SOSIAL TWITTER MENGGUNAKAN  
METODE *K-NEAREST NEIGHBOR***

**SKRIPSI**

**Oleh:**

**AJI SUPRAPTO**

**NIM. 12650105**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2017**

**SISTEM KLASIFIKASI OPINI PENGGUNA MASKAPAI  
PENERBANGAN DI INDONESIA PADA JEJARING  
SOSIAL TWITTER MENGGUNAKAN  
METODE *K-NEAREST NEIGHBOR***

**SKRIPSI**

**Diajukan Kepada:  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S. Kom)**

**Oleh:  
AJI SUPRAPTO  
NIM. 12650105**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2017**

**LEMBAR PERSETUJUAN**

**SISTEM KLASIFIKASI OPINI PENGGUNA MASKAPAI  
PENERBANGAN DI INDONESIA PADA JEJARING  
SOSIAL TWITTER MENGGUNAKAN  
METODE K-NEAREST NEIGHBOR**

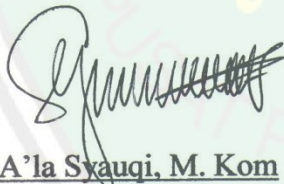
**SKRIPSI**

Oleh:  
**Aji Suprpto**  
**NIM. 12650105**

Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal 20 Desember 2016

Pembimbing I,

Pembimbing II,

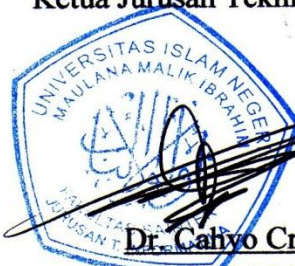


**A'la Syauqi, M. Kom**  
NIP. 19771201 200801 1 007



**Supriyono, M. Kom**  
NIDT. 19841010 20161801 1 078

Mengetahui,  
Ketua Jurusan Teknik Informatika



**Dr. Cahyo Crysdian**  
NIP. 19740424 200901 1 008

**HALAMAN PENGESAHAN**

**SISTEM KLASIFIKASI OPINI PENGGUNA MASKAPAI  
PENERBANGAN DI INDONESIA PADA JEJARING  
SOSIAL TWITTER MENGGUNAKAN  
METODE *K-NEAREST NEIGHBOR***

**SKRIPSI**

Oleh:  
**Aji Suprpto**  
NIM. 12650105

Telah Dipertahankan di Depan Penguji Skripsi dan  
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk  
Memperoleh Gelar Sarjana Komputer (S.Kom)  
Tanggal 30 Desember 2016

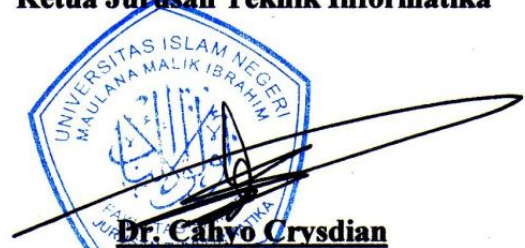
**Susunan Dewan Penguji**

1. Penguji Utama : **Dr. M. Faisal, MT**  
NIP. 19740510 200501 1 007
2. Ketua : **Yunifa Miftachul Arif, MT**  
NIP. 19830616 201101 1 004
3. Sekretaris : **A'la Svauqi, M. Kom**  
NIP. 19771201 200801 1 007
4. Anggota : **Supriyono, M. Kom**  
NIDT. 19841010 20161801 1 078

**Tanda Tangan**

(  )  
(  )  
(  )  
(  )

**Mengetahui dan Mengesahkan,  
Ketua Jurusan Teknik Informatika**

  
**Dr. Cahyo Crysdiyan**  
NIP. 19740424 200901 1 008

**HALAMAN PERNYATAAN  
ORISINALITAS PENELITIAN**

Saya yang bertanda tangan di bawah ini:

Nama : Aji Suprpto  
NIM : 12650105  
Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika  
Judul Penelitian : Sistem Klasifikasi Opini Pengguna Maskapai  
Penerbangan Di Indonesia Pada Jejaring Sosial Twitter  
Menggunakan Metode K-Nearest Neighbor

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur – unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis di kutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka. Apabila ternyata hasil penelitian ini terbukti terdapat unsur – unsur jiplakan, maka saya bersedia mempertanggung jawabkan, serta diproses sesuai aturan yang berlaku.

Malang, 20 Desember 2016  
Yang membuat pernyataan,



Aji Suprpto  
NIM. 12650105

## MOTTO

وَلِكُلِّ وِجْهَةٌ هُوَ مُوَلِّيَهَا ۖ فَاسْتَبِقُوا الْخَيْرَاتِ ۚ أَيْنَ مَا تَكُونُوا يَأْتِ بِكُمْ  
اللَّهُ جَمِيعًا ۚ إِنَّ اللَّهَ عَلَىٰ كُلِّ شَيْءٍ قَدِيرٌ ﴿١٤٨﴾

“ Dan bagi tiap-tiap umat ada kiblatnya (sendiri) yang ia menghadap kepadanya.  
Maka berlomba-lombalah (dalam membuat) kebaikan. di mana saja kamu berada  
pasti Allah akan mengumpulkan kamu sekalian (pada hari kiamat). Sesungguhnya

Allah Maha Kuasa atas segala sesuatu ”

## HALAMAN PERSEMBAHAN

Dengan segala puja dan puji syukur kepada Allah SWT yang masih melimpahkan rahmat dan hidayah-Nya dan atas dukungan dan do'a dari orang-orang tercinta, akhirnya skripsi ini dapat dirampungkan dengan baik. Oleh karena itu, dengan rasa bangga dan bahagia saya persembahkan rasa syukur dan terima kasih saya kepada:

1. Bapak Sularso dan Ibu Tu'Ah yang telah mengajarkan penulis segala kebaikan yang ada pada kehidupan ini, yang tiada henti memberikan do'a dan pengorbanan yang besar hingga penulis menyelesaikan studi ini
2. Kakak saya Eni Setyawati dan Fery Widiyanto yang telah memberi dukungan moril maupun materil kepada penulis selama ini.
3. Keluarga IMM UIN Malang yang selalu memberikan ilmu dan pengalaman dalam perkuliahan ini, serta kebahagiaan tersendiri bisa berada dalam keluarga ini.
4. Teman-teman teknik informatika 2012 tercinta yang telah membantu dalam menyelesaikan studi ini.

## KATA PENGANTAR

Segala puji syukur kehadirat Allah SWT sehingga skripsi ini dapat diselesaikan dengan baik dan kesulitan yang banyak ditemukan dalam penyelesaiannya dapat terselesaikan berkat bantuan dan bimbingan dari berbagai pihak dan akhirnya penulis berhasil menyelesaikan skripsi ini.

Dalam penulisan skripsi ini penulis banyak mendapatkan bantuan serta dorongan motivasi dari berbagai pihak dan dengan segala kerendahan hati, penulis mengucapkan terimakasih kepada:

1. Bapak A'la Syauqi, M.Kom dan Bapak Supriyono, M.Kom selaku dosen pembimbing penulis yang telah memberikan kritik, saran dan masukan serta bersedia meluangkan waktu, tenaga dan pikiran untuk membantu penulis menyelesaikan skripsi ini.
2. Bapak Dr. Cahyo Crysdiyan, M.Cs selaku ketua jurusan dan bapak Yunifa Miftachul Arif, M.T. selaku dosen pembimbing akademik penulis.
3. Seluruh Dosen dan Staf Administrasi jurusan Teknik Informatika UIN Malang, terima kasih atas segala ilmu dan bimbingannya.
4. Seluruh rekan-rekan studi yang tidak dapat disebutkan satu persatu, terima kasih atas segala kebaikan yang diberikan kepada penulis.

Akhir kata penulis mengharapkan semoga skripsi ini dapat memberikan kontribusi ilmiah dalam riset bidang informatika serta bermanfaat dan membantu semua pihak yang membutuhkan.

Malang, 20 Desember 2016

Penulis



## DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>HALAMAN PERNYATAAN.....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>MOTTO .....</b>	<b>IV</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>V</b>
<b>KATA PENGANTAR.....</b>	<b>VI</b>
<b>DAFTAR ISI.....</b>	<b>VII</b>
<b>DAFTAR GAMBAR.....</b>	<b>IX</b>
<b>DAFTAR TABEL .....</b>	<b>X</b>
<b>ABSTRAK .....</b>	<b>XII</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Tujuan Penelitian .....	3
1.4. Batasan Masalah.....	4
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1. Penelitian Terkait .....	5
2.2. <i>Data Mining</i> .....	7
2.3. <i>Text Mining</i> .....	8
2.4. <i>Sentiment Analysis</i> .....	10
2.5. Twitter.....	10
2.6. <i>Text Preprocessing</i> .....	11
2.7. Stemming Nazief Adriani .....	12
2.8. <i>Feature Weighting WIDF</i> .....	16
2.9. <i>K-Nearest Neighbor</i> .....	17
2.10. Evaluasi.....	19
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>21</b>
3.1. Perancangan Sistem .....	21
3.2. <i>Input</i> .....	22
3.3. <i>Dataset</i> .....	22
3.4. <i>Preprocessing</i> .....	24

3.4.1.	<i>Case Folding</i> .....	25
3.4.2.	<i>Cleansing</i> .....	26
3.4.3.	<i>Tokenizing</i> .....	27
3.4.4.	<i>Convert Number</i> .....	27
3.4.5.	<i>Normalization</i> .....	28
3.4.5.1.	Konversi Kata Singkatan.....	28
3.4.5.2.	Konversi Kata Baku .....	29
3.4.5.3.	Konversi Kata Inggris .....	29
3.4.6.	<i>Stopword Removal</i> .....	30
3.4.7.	<i>Stemming</i> .....	31
3.5.	Pembobotan WIDF .....	31
3.6.	<i>K-Nearest Neighbor</i> .....	33
3.7.	Hasil Analisis Sistem .....	40
3.8.	Analisis Sistem.....	40
3.9.	Sumber Data.....	41
<b>BAB IV UJI COBA DAN PEMBAHASAN</b> .....		<b>42</b>
4.1.	Implementasi .....	42
4.1.1.	Pengumpulan <i>Data Sampling</i> .....	42
4.1.2.	Pengumpulan <i>Data Testing</i> .....	44
4.1.3.	Pengolahan Data pada <i>Database Server</i> .....	46
4.1.4.	Proses <i>Preprocessing</i> Dokumen .....	50
4.1.5.	Pembobotan WIDF .....	56
4.1.6.	Kemiripan Data dengan <i>Cosine Similarity</i> .....	58
4.1.7.	Desain dan Implementasi GUI.....	61
4.2.	Pengujian Sistem.....	64
4.3.	Integrasi Islam.....	67
<b>BAB V KESIMPULAN DAN SARAN</b> .....		<b>70</b>
5.1.	Kesimpulan .....	70
5.2.	Saran.....	70
<b>DAFTAR PUSTAKA</b> .....		<b>71</b>

## DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Sistem .....	21
Gambar 3.2 Blok Diagram Persiapan Dataset.....	23
Gambar 3.3 Blok Diagram Alir Preprocessing .....	25
Gambar 3.4 Blok Diagram Alir Normalization .....	28
Gambar 3.5 Blok Diagram Alir K-Nearest Neighbor .....	33
Gambar 3.6 Rancangan Tampilan GUI.....	40
Gambar 4.1 Tampilan Pencarian Query .....	43
Gambar 4.2 Tampilan Inspect Elemen Web .....	43
Gambar 4.3 Fungsi Cek Koneksi Internet.....	45
Gambar 4.4 Fungsi Cek Aplikasi Python.....	46
Gambar 4.5 Fungsi Menjalankan File Installer Python .....	46
Gambar 4.6 Fungsi Case Folding Dokumen Tweet .....	50
Gambar 4.7 Fungsi Cleansing Dokumen Tweet .....	50
Gambar 4.8 Fungsi Tokenizing Dokumen Tweet .....	51
Gambar 4.9 Fungsi Resize Dokumen Tweet .....	51
Gambar 4.10 Fungsi Convert Number Dokumen Tweet .....	52
Gambar 4.11 Fungsi Resize Dokumen Tweet.....	52
Gambar 4.12 Fungsi Normalisasi Kata Singkatan Dokumen Tweet .....	53
Gambar 4.13 Fungsi Stopword Removal Dokumen Tweet .....	54
Gambar 4.14 Fungsi Penghapusan Infleksional Suffiks .....	55
Gambar 4.15 Fungsi Bobot TF Dokumen.....	56
Gambar 4.16 Fungsi Bobot TF Seluruh Dokumen .....	57
Gambar 4.17 Fungsi Perkalian Vektor.....	60
Gambar 4.18 Tampilan GUI 1 .....	62
Gambar 4.19 Tampilan GUI 2 .....	63
Gambar 4.20 Grafik Perbandingan Range .....	66

## DAFTAR TABEL

Tabel 2.1 Gabungan Awalan dan Akhiran Tidak Diiijinkan .....	13
Tabel 2.2 Aturan <i>Derivation Prefix</i> .....	14
Tabel 2.3 Confusion Matrix (Novantirani, 2015) .....	19
Tabel 3.1 Dokumen Tweet .....	22
Tabel 3.2 Contoh Tahapan Case Folding .....	26
Tabel 3.3 Contoh Tahapan Cleansing .....	26
Tabel 3.4 Contoh Tahapan Tokenizing .....	27
Tabel 3.5 Contoh Tahapan Convert Number .....	27
Tabel 3.6 Contoh Tahapan Konversi Kata Singkatan .....	29
Tabel 3.7 Contoh Tahapan Konversi Kata Baku .....	29
Tabel 3.8 Contoh Tahapan Konversi Kata Inggris .....	30
Tabel 3.9 Contoh Tahapan Stopword Removal .....	30
Tabel 3.10 Contoh Koleksi Data .....	32
Tabel 3.11 Hasil Bobot Koleksi Data .....	32
Tabel 3.12 Contoh Bobot WIDF Dokumen Latih .....	34
Tabel 3.13 Contoh Bobot dan Opini Dokumen Latih .....	35
Tabel 3.14 Contoh Bobot WIDF Dokumen Uji .....	35
Tabel 3.15 Contoh Bobot Baru Kata .....	36
Tabel 3.16 Contoh Bobot Baru Dokumen Baru .....	37
Tabel 3.17 Contoh Nilai Kemiripan Data Dokumen .....	38
Tabel 3.18 Contoh Pengurutan Nilai Kemiripan Data .....	39
Tabel 3.19 Contoh Hasil Pembatasan Ketetanggan .....	39
Tabel 4.1 Rincian Jumlah Data Sampling .....	44
Tabel 4.2 Struktur Tabel Acuan .....	47
Tabel 4.3 Struktur Tabel Datalatih .....	47
Tabel 4.4 Struktur Tabel Datauji .....	48
Tabel 4.5 Struktur Tabel dtlatih_kata .....	49
Tabel 4.6 Struktur Tabel dtlatih_bobotkata .....	49
Tabel 4.7 Struktur Database dt_singkatan .....	53
Tabel 4.8 Struktur Database dt_stopword .....	54
Tabel 4.9 Daftar Nilai TF Dokumen .....	56

Tabel 4.10 Daftar Nilai TF Seluruh Dokumen.....	57
Tabel 4.11 Hasil Perhitungan WIDF.....	58
Tabel 4.12 Daftar Bobot Dokumen Uji.....	59
Tabel 4.13 Daftar Bobot Query Dokumen Latih .....	59
Tabel 4.14 Daftar Jumlah Bobot Query Dokumen Latih.....	60
Tabel 4.15 Bobot Perkalian Vektor.....	60
Tabel 4.16 Bobot Nilai Jarak Dokumen.....	61
Tabel 4.17 Hasil Pengujian Sistem Range 3 .....	64
Tabel 4.18 Hasil Pengujian Sistem Range 5 .....	65
Tabel 4.19 Hasil Pengujian Sistem Range 9 .....	66



## ABSTRAK

Suprpto, Aji. 2017. *Sistem Klasifikasi Opini Pengguna Maskapai Penerbangan Di Indonesia Pada Jejaring Sosial Twitter Menggunakan Metode K-Nearest Neighbor*. Skripsi Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang.  
Pembimbing: (I) A'la Syauqi, M. Kom (II) Supriyono, M. Kom

---

**Kata Kunci:** Microblogging, Text Mining, Klasifikasi, K-Nearest Neighbor, WIDF

*Pada saat ini media sosial telah menjadi alat komunikasi yang sangat populer di kalangan pengguna internet di Indonesia. Salah satu media sosial tersebut yakni twitter dengan jumlah opini yang besar dan didalamnya terdapat informasi yang sangat berharga sebagai alat penentu kebijakan dan ini bisa dilakukan dengan menggunakan text mining. Sebagai contoh, bagaimana masyarakat bereaksi terhadap suatu pelayanan maskapai penerbangan di Indonesia atas segala pengalaman atau kejadian yang saat itu menjadi isu hangat. Data Sampling dalam sistem ini menggunakan 4.342 opini tweet yang diproses dengan algoritma WIDF dalam pembobotan dan metode K-Nearest Neighbor dalam tahap klasifikasi opini. Hasil pengujian diperoleh dengan range 3 pada algoritma Cosine Similarity menjadi nilai akurasi yang tertinggi dengan nilai akurasi sebesar 86.674% pada opini positif dan 93.345% pada opini negatif.*

## ABSTRACT

Suprpto, Aji. 2017. *Sistem Klasifikasi Opini Pengguna Maskapai Penerbangan Di Indonesia Pada Jejaring Sosial Twitter Menggunakan Metode K-Nearest Neighbor*. Undergraduate Thesis Informatics Engineering Department. Faculty of Science and Technology. State Islamic University of Maulana Malik Ibrahim Malang.

Adviser: (I) A'la Syauqi, M. Kom (II) Supriyono, M. Kom

---

**Keywords:** Microblogging, Text Mining, Classification, K-Nearest Neighbor, WIDF

*At this time social media has become a very popular means of communication among Internet users in Indonesia. One of the twitter social media with a large number of opinions and information contained therein is very valuable as a tool of policy makers and this can be done using text mining. For example, how people react to an airline services in Indonesia on the experiences or events when it became a hot issue. Data Sampling in this system uses 4,342 tweets opinion which is processed by the algorithm WIDF in weighting and K-Nearest Neighbor method in the stage classification opinions. The test results obtained by the range 3 on Cosine Similarity algorithm becomes the highest accuracy value of 86.674% accuracy on a positive opinion and 93.345% on the negative opinion.*

## الملخص

سوبرابنتو، العاجي. 2016. الخطوط الجوية المستخدمين الرأي نظام التصنيف في أقرب طريقة الجار. وزارة أطروحة إندونيسيا على تويتر شبكة اجتماعية عن طريق المعلوماتية، كلية العلوم والتكنولوجيا. جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانج

كوم، - كوم، - علاء المشرف

كلمات البحث: المدونات الصغيرة، والتعددين النص، تصنيف، الجار أقرب، -

في هذا الوقت أصبحت وسائل الإعلام الاجتماعية وسيلة شعبية جدا للاتصال بين مستخدمي الإنترنت في إندونيسيا. وسيلة من وسائل الإعلام الاجتماعية هي "تويتر" مع عدد كبير من الآراء والمعلومات الواردة فيه هي قيمة للغاية باعتبارها أداة من صانعي السياسات، وهذا يمكن أن يتم ذلك باستخدام التعددين النص. على سبيل المثال، كيف يتفاعل الناس إلى خدمات شركات الطيران في اندونيسيا على خبرات أو أحداث عندما أصبح قضية ساخنة. أخذ العينات البيانات في هذا النظام يستخدم 4342 تويت الرأي أقرب طريقة الجار في الآراء تصنيف في الترجيح والتي تتم معالجتها بواسطة خوارزمية المرحلة. نتائج الاختبار التي حصل عليها مجموعة 3 في جيب التمام تشابه خوارزمية تصبح أعلى قيمة الدقة لقيمة 86 674% من الدقة على رأي إيجابي و 93 345% على رأي سلبي



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Saat ini transportasi udara merupakan pilihan utama masyarakat dalam perjalanan lintas kota ataupun antar pulau. Hal ini dikarenakan waktu yang cukup singkat yang dapat ditempuh oleh maskapai di Indonesia dan setiap tahunnya jumlah penumpang transportasi udara mengalami peningkatan. Hal tersebut terbukti dari data statistik tahun 2001 hingga 2013, jumlah penumpang keberangkatan dalam negeri naik sebesar 708% dan untuk keberangkatan luar negeri naik sebesar 283%. (bps.go.id, 2016)

Dari peningkatan jumlah penumpang tersebut, tidak bisa dikatakan bahwa transportasi udara merupakan transportasi yang menjanjikan keselamatan penumpang sampai pada tempat tujuan. Terbukti pada akhir tahun 2014, terjadi insiden jatuhnya salah satu maskapai penerbangan ke dalam laut Jawa dan menewaskan seluruh penumpang yang ada dalam pesawat sejumlah 166 orang. (voaIndonesia.com, 2015)

Kementrian perhubungan langsung bergerak cepat merespon kejadian tersebut dan pada awal tahun 2015 merencanakan akan melakukan pemeringkatan keselamatan penerbangan (*safety rating*) untuk mendorong keselamatan dan kemandirian industri penerbangan nasional dan akan mempublikasikannya tiap 3 bulan sekali (dephub.go.id, 2015). Hal tersebut bertujuan agar setiap maskapai penerbangan dapat selalu memberikan pelayanan yang terbaik kepada konsumen dengan memperbaiki pelayanan dan keselamatan awak penumpang.

Adapun maskapai penerbangan yang menjadi objek penelitian ini adalah Lion Air dan Citilink. Dilansir dari artikel yang ditulis dalam media online [tribunnews.com](http://tribunnews.com), kementerian perhubungan melansir total jumlah penumpang pesawat di Indonesia hingga akhir Juni 2015, Lion Air dan Citilink termasuk dalam 5 peringkat teratas dengan jumlah penumpang terbesar. Dengan demikian dapat dijelaskan bahwa semakin banyak penumpang pada suatu maskapai penerbangan, maka semakin banyak pula opini yang akan muncul.

Sehubungan dengan akan diadakannya *safety rating* pada maskapai penerbangan di Indonesia oleh kementerian perhubungan, maka opini konsumen juga dapat dipertimbangkan sebagai salah satu kunci utama yang dapat digunakan sebagai parameternya dan hal itu banyak diutarakan melalui sebuah teks pada media sosial khususnya twitter yang nantinya dapat diolah menjadi informasi yang berguna.

Di Indonesia, twitter merupakan media sosial urutan ke-3 yang sering digunakan dengan penetrasi sebesar 11% dari total populasi akun aktif diberbagai jaringan sosial seperti facebook, whatsapp, instagram dan line. Pengguna media sosial twitter di Indonesia pada tahun 2014 sebesar 12 juta pengguna dan diperkirakan pada tahun 2019 meningkat menjadi 22,8 juta pengguna, meningkat sebanyak 190% ([statista.com](http://statista.com), 2016). Kesederhanaan dan kemudahan dalam penggunaan twitter merupakan alasan mengapa twitter lebih digemari masyarakat Indonesia dalam berkomunikasi. Tentu saja, informasi yang terkandung dalam tweet ini sangat berharga sebagai alat penentu kebijakan dan ini bisa dilakukan dengan menggunakan *text mining*.

*Text mining* sebagai salah satu solusi dalam mengatasi masalah di atas yang dapat didefinisikan sebagai pengambilan informasi yang bersumberkan dari beberapa dokumen. Dalam *text mining* terdapat beberapa tujuan penggunaan yang khas diantaranya *text categorization*, *text clustering* dan *sentiment analysis* (Feldman, 2007). Salah satu tujuan *text mining* yang dapat digunakan dalam hal ini adalah analisis sentimen yang merupakan proses pengklasifikasian sebuah opini ke dalam opini positif atau opini negatif.

Berdasarkan penelitian (Khamar, 2013) yang mencoba membandingkan tingkat akurasi dari 3 algoritma klasifikasi yaitu KNN, *Naive Bayes* dan SVM pada teks pendek yang hanya berisi beberapa kata, didapat algoritma KNN memberikan akurasi yang paling baik daripada dua algoritma lainnya. Penggunaan metode KNN ini juga berdasarkan pada kenyataan penelitian sebelumnya yang belum ditemukan kombinasi antara metode KNN dengan pembobotan kata *Weighting Inverse Document Frequency* (WIDF) yang digunakan pada penelitian ini. Maka dari itu pada penelitian ini akan digunakan metode *K-Nearest Neighbor* untuk melakukan tahap klasifikasi opini.

## 1.2. Rumusan Masalah

Seberapa akurat metode *k-Nearest Neighbor* digunakan untuk pengklasifikasian opini maskapai penerbangan Lion Air dan Citilink?

## 1.3. Tujuan Penelitian

- 1) Mengetahui tingkat akurasi sistem dalam klasifikasi opini untuk maskapai penerbangan Lion Air dan Citilink.
- 2) Membuat program klasifikasi opini otomatis untuk maskapai penerbangan Lion Air dan Citilink.

#### 1.4. Batasan Masalah

- 1) Data yang dianalisis adalah semua tweet berbahasa Indonesia.
- 2) Tweet yang diperoleh dengan mengabaikan akun usernya.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Penelitian Terkait

Penelitian mengenai analisis sentimen telah dilakukan menggunakan F3 (*F3 is Factor Finder*) yang memiliki beberapa metode praproses yang diperkirakan mampu menangani permasalahan model bahasa yang ditemukan. F3 dalam penelitian ini menggunakan metode *Naive Bayes* untuk melakukan analisis sentimen karena telah teruji di berbagai penelitian. Sedangkan untuk mengetahui perubahan sentimen akan digunakan metode Tf-Idf dengan *discounted-cumulative* untuk menangani karakter topik yang muncul di Twitter yang berkelanjutan. Hasil analisis dan pengujian menunjukkan tahapan praproses yang dicoba dalam penelitian ini tidak memiliki pengaruh yang signifikan terhadap akurasi sistem. Sedangkan penggunaan Tf-Idf dengan *discounted-cumulative* mampu meningkatkan jumlah topik terekstraks yang sesuai, namun memiliki kelemahan ketika menghadapi topik yang termuat di hampir seluruh selang waktu atau topik yang bukan bersumber dari berita media internet. (Sunni, I. & Widyantoro, D. H., 2012)

Penelitian sentimen pernah dilakukan untuk mengkategorikan sentimen positif, negatif dan netral pada twitter. Sumber data yang dibahas dalam penelitian ini mengambil semua data yang ada seperti film, produk, isu politik dsb. Ekstraksi kata yang dilakukan untuk mendapatkan opini yang sesuai dan kemudian mendapatkan orientasi yang ada. Pertimbangan dalam pengkategorian opini dalam penelitian ini digunakan kombinasi dari kata sifat, kerja dan kata keterangan. Data yang sudah siap di proses akan dikalkulasikan menggunakan *linear equation* yang

akan digabungkan dengan kekuatan emosi yang ada. Hasil dari penelitian ini, prototype yang digunakan telah dievaluasi sehingga teknik yang digunakan dalam penelitian ini dapat digunakan sebagai acuan penelitian selanjutnya. (Kumar, A. & Sebastian, T. M., 2012)

Penelitian lainnya mencoba melakukan klasifikasi sentimen terhadap data yang diperoleh dari twitter dengan mengambil tweet akun presiden RI @SBYudhoyono. Metode yang digunakan yaitu *Naive Bayes* dengan hasil sentimen tweet berupa positif, negatif dan netral. Penelitian ini terkadang menemukan beberapa tweet adalah kultwit yang merupakan serangkaian tweet yang berturut-turut membahas satu topik tertentu. Dalam hal ini, kadang-kadang sulit untuk menentukan sentimen tweet jika tidak mempertimbangkan konteksnya. Selain itu sulit menentukan jenis sentimen jika pada sebuah tweet mengungkapkan dua pendapat yang berbeda. Penelitian ini mendapatkan hasil yang baik dengan akurasi 79,42% dengan hasil yang didominasi oleh sentimen netral. (Aliandu, P., 2013)

Penelitian klasifikasi subtopik berita dilakukan menggunakan algoritma *K-Nearest Neighbor* dengan *decision rule* yang diharapkan dapat menambah hasil keakurasian dalam penelitian ini. Pengujian sistem ini menggunakan aturan keputusan nilai k terdekat untuk mengetahui tingkat nilai akurasi yang paling tinggi. Sistem ini memiliki tingkat akurasi 88,29% dengan nilai k=3, akan tetapi penggunaan *decision rule* masih dirasa kurang mampu memaksimalkan performa dari metode *K-Nearest Neighbor*, buktinya hasil yang didapatkan dengan nilai k = 3 dengan *decision rule* menjadi 89,36%. (Samuel, Y., Delima, R. & Rachmat, A., 2014)

Penelitian lainnya pernah melakukan klasifikasi terhadap sentimen opini suatu acara televisi pada microblogging twitter. Hal yang dilakukan dalam penelitian ini yakni menggunakan R programming dalam mengklasifikasikan opini masyarakat dalam opini positif, negatif dan netral. Tahap klasifikasi membutuhkan kamus sentimen sebagai acuan setiap kategori yang ada berdasarkan pemilihan kata yang biasa digunakan dalam komentar pada twitter dan berdasarkan analisa penulis terhadap hasil dari crawling yang telah dilakukan. Sistem ini menghasilkan kesimpulan bahwa opini yang tertinggi yaitu opini netral terhadap keyword tersebut dan melihat polaritas kemunculan kata “suka” pada wordcloud dapat dijadikan indikator bahwa acara tersebut merupakan salah satu acara favorit masyarakat Indonesia. (Sussolaikah, Kelik & Alwi, Aslan. 2016)

Penelitian yang dilakukan nantinya menyesuaikan dengan beberapa jurnal terkait, terdapat beberapa persamaan dalam pembahasan bidang *text mining*, analisa sentimen, metode K-NN maupun media sosial twitter. Penelitian ini mempunyai kelebihan dalam algoritma pembobotan yang digunakan, diperkirakan tingkat nilai akurasi akan lebih baik jika dibandingkan dengan metode pembobotan yang pernah dilakukan sebelumnya dalam bidang *text mining*.

## 2.2. *Data Mining*

*Data mining* adalah proses untuk menemukan pengetahuan yang menarik dari data dalam jumlah besar (Han, 2000). Ini merupakan bidang interdisipliner dengan kontribusi dari berbagai bidang seperti statistik, pembelajaran mesin, pencarian informasi, pengenalan pola dan bioinformatika. *Data mining* secara luas digunakan di banyak domain, seperti ritel, keuangan, telekomunikasi dan media

sosial. Dalam aplikasi dunia nyata, proses *data mining* dapat dipecah menjadi enam fase utama: *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation* dan *deployment*. (Zhao, 2013)

Menurut Liu (2007), *data mining* merupakan bidang multi-disiplin yang melibatkan pembelajaran mesin, statistik, *database*, kecerdasan buatan, pencarian informasi dan visualisasi. *Data mining* juga disebut *knowledge discovery in databases* (KDD). Hal ini biasanya didefinisikan sebagai proses menemukan pola yang berguna atau pengetahuan dari sumber data, misalnya *database*, teks, gambar, web, dll. Pola yang didapat harus valid, berpotensi berguna dan mudah dipahami. Ada banyak tugas *data mining*, beberapa yang umum adalah *supervised learning* (klasifikasi), *unsupervised learning* (pengelompokan), *association rule mining* dan *sequential pattern mining*.

*Data mining* terdiri dari algoritma inti yang memungkinkan seseorang untuk mendapatkan wawasan dasar dan pengetahuan dari data besar. Ini adalah bidang interdisipliner yang merupakan bagian dari proses penemuan pengetahuan yang lebih besar, yang mencakup tugas pra-pengolahan seperti ekstraksi data, pembersihan data, pencampuran data, reduksi data dan fitur konstruksi serta langkah-langkah pasca-pengolahan seperti *pattern and model interpretation*, *hypothesis confirmation and generation* dan sebagainya. (Zaki, 2014)

### 2.3. *Text Mining*

*Text mining* adalah bidang baru yang sedang berkembang yang mencoba untuk mengumpulkan informasi yang berarti dari teks bahasa alami. Ini mungkin dicirikan sebagai proses menganalisis teks untuk mengekstrak informasi yang



berguna untuk tujuan tertentu. Bidang *text mining* biasanya berhubungan dengan teks-teks yang fungsinya adalah komunikasi informasi faktual atau opini. (Witten, 2005)

Menurut Feldman (2007), *text mining* didefinisikan secara luas sebagai proses pengetahuan intensif di mana pengguna berinteraksi dengan koleksi dokumen dari waktu ke waktu dengan menggunakan seperangkat alat analisis. Dalam cara yang sejalan dengan *data mining*, *text mining* berupaya untuk mengekstrak informasi yang berguna dari sumber data melalui identifikasi dan eksplorasi pola yang menarik. Pola menarik yang ditemukan tidak ditemukan di catatan *database* formal namun ditemukan dalam data tekstual yang tidak terstruktur pada dokumen ini. Selain itu *text mining* juga mengacu pada kemajuan yang dibuat dalam disiplin ilmu komputer yang berhubungan dengan penanganan bahasa alami. Terutama yang mengeksploitasi teknik dan metodologi dari bidang pencarian informasi, ekstraksi informasi dan berbasis korpus linguistik komputasi.

Penelitian dibidang *text mining* menangani masalah yang berkaitan dengan representasi teks, klasifikasi, pengelompokan, ekstraksi atau pencarian informasi dan pemodelan pola. Dalam hal ini pemilihan karakteristik domain dan prosedur penelitian menjadi peran penting. Oleh karena itu, adaptasi dari algoritma *data mining* dari teks yang diketahui sangat diperlukan. Untuk mencapai hal tersebut berdasarkan penelitian sebelumnya *text mining* bergantung pada *information retrieval*, *natural language processing* dan *information extraction*. Selain itu, penerapan metode *data mining* dan statistik juga diterapkan untuk menangani masalah ini. (Hotho, 2005)

#### 2.4. *Sentiment Analysis*

*Sentiment analysis* atau yang sering juga disebut *opinion mining* adalah sebuah studi yang menganalisis opini dari seseorang atau sentimen, evaluasi, pencapaian, sikap dan emosi terhadap entitas tertentu seperti produk, layanan, organisasi topik, event atau bahkan individu dan berbagai atributnya. Kecendrungan penelitian tentang analisis sentimen berfokus pada pendapat yang menyatakan atau menyiratkan suatu sentimen positif atau negatif. Pendapat mewakili hampir semua aktivitas manusia, karena pendapat dapat mempengaruhi terhadap perilaku seseorang. Setiap kali kita perlu membuat keputusan, kita ingin tahu pendapat orang lain. Dalam dunia nyata, bisnis dan organisasi selalu ingin melihat opini publik tentang suatu produk atau jasa. (Liu, 2012)

Sementara itu, menurut Barawi (2013) *sentiment analysis* adalah sebuah aplikasi dari komputasi linguistik, analisis teks dan pengolahan bahasa alami yang digunakan untuk mengidentifikasi dan mengambil isi subjektif dalam sumber bahan (teks). Ini merupakan bidang yang mempelajari emosi manusia dalam pengolahan bahasa alami, hal itu didefinisikan sebagai klasifikasi orientasi semantik dokumen berdasarkan sentimen yang diungkapkan oleh seseorang.

#### 2.5. Twitter

Twitter merupakan salah satu layanan *microblogging* paling dikenal dan digunakan oleh banyak komunitas, politisi, media dan sebagainya. Selain itu, survei terbaru menunjukkan bahwa 19% dari pengguna web menggunakan layanan status pembaruan untuk berbagi dan melihat pembaruan status. (Letierce, 2010)

Disamping itu menurut Suh (2010), twitter adalah layanan paling populer dikarenakan kemudahan untuk berbagi informasi secara *real-time* yang dapat berdampak terhadap wacana publik di masyarakat. Twitter memungkinkan pengguna untuk mengirim dan membaca pesan singkat sepanjang 140 karakter yang dikenal sebagai tweets dan menemukan topik yang menarik secara *real-time*. Dalam jurnalnya, Davidov (2010) menyimpulkan bahwa sebuah tweet biasanya mengandung alamat URL, *username* (@), *hashtag* (#) yang biasanya digunakan untuk menandai atau menentukan topik tertentu dan *emoticon* yang mewakili ekspresi wajah dengan karakter tertentu, hal ini untuk menggambarkan suasana hati atau emosi pengguna.

## 2.6. Text Preprocessing

Struktur data yang baik dapat memudahkan proses komputerisasi secara otomatis. Pada *text mining*, informasi yang akan digali berisi informasi-informasi yang strukturnya sembarang. Oleh karena itu, diperlukan proses perubahan bentuk menjadi data yang terstruktur sesuai kebutuhannya untuk proses dalam *data mining*, yang biasanya akan menjadi nilai-nilai numerik. Proses ini sering disebut dengan *text preprocessing*. (Feldman, 2007)

*Text preprocessing* merupakan tahap yang dilakukan sebelum masuk dalam pengklasifikasian opini pada penelitian ini. Proses ini bertujuan untuk melakukan *treatment* awal terhadap data demi menghilangkan permasalahan-permasalahan yang dapat mengganggu hasil daripada proses *data mining*. Pada penelitian ini, langkah-langkah yang dilakukan dalam *text preprocessing* adalah *tokenizing*, *case folding*, *cleansing*, *converting*, *stopword removal* dan *stemming*.

## 2.7. *Stemming Nazief Adriani*

Algoritma ini didasarkan pada aturan morfologi komprehensif yang dirangkum atas boleh atau tidaknya kata afiks, termasuk prefiks, sufiks, dan confixes (kombinasi awalan dan akhiran) yang juga dikenal sebagai circumfixes.

Algoritma *stemming Nazief & Adriani* memiliki tahap-tahap sebagai berikut:

(Asian, 2007)

- 1) Cari kata yang akan di proses dalam kamus kata dasar, jika ditemukan maka diasumsikan kata adalah *root word* dan algoritma ini berhenti.
- 2) Hapus *inflectional suffixes* (“-lah”, “-kah”, “-tah” atau “-pun”) dan dilanjutkan dengan penghapusan *possesive pronouns* (“-ku”, “-mu” atau “-nya”) jika ada. Contohnya dalam kata “bajumlah” yang nantinya akan menjadi kata “baju”. Cek dalam kamus kata dasar, jika ditemukan maka diasumsikan kata adalah *root word* dan algoritma ini berhenti.
- 3) Hapus *derivational suffixes* (“-i”, “-an” atau “-kan”) seperti contoh kata “membelian” yang akan menjadi kata “membeli”. Jika sampai ditahap ini masih belum ditemukan dalam kamus kata dasar, maka dilanjutkan dalam penghapusan *prefix* ditahap selanjutnya.
- 4) Hapus *derivational prefixes* (“be-”, “di-”, “ke-”, “me-”, “pe-”, “se-”, atau “te-”).
  - a. Hentikan proses ini jika:
    - Kata awal dalam langkah 3 mempunyai gabungan awalan dan imbuhan yang tidak diijinkan dalam tabel 2.1.

Tabel 2.1 Gabungan Awalan dan Akhiran Tidak Diijinkan

Awalan	Akhiran
“ber-“	“-i”
“di-“	“-an”
“ke-“	“-i”, -kan”
“me-“	“-an”
“ter-“	“-an”
“per-“	“-an”

- Awalan yang dihilangkan sama dengan awalan yang dihilangkan sebelumnya.
  - Awalan telah dihilangkan sebanyak 3 kali.
- b. Identifikasi tipe awalan dan disambiguitasnya jika perlu. Awalan ini dibagi menjadi 2 tipe:
- *Plain*: awalan (“di-”, “ke-” atau “se-”) dapat dihilangkan secara langsung.
  - *Complex*: awalan (“be-”, “te-”, “me-” atau “pe-”) harus dianalisis ambiguitasnya menggunakan aturan yang dijelaskan pada tabel 2.2 karena hal ini memiliki varian yang berbeda. Awalan “me-” bisa menjadi “mem-”, “men-”, “meny-” atau “meng-” tergantung pada huruf di awal.

Tabel 2.2 Aturan *Derivation Prefix*

<i>Rule</i>	<i>Construct</i>	<i>Return</i>
1	berV...	ber-V...   be-rV...
2	berCAP...	ber-CAP... where C!= 'r' and P!= 'er'
3	berCAerV...	ber-CAerV... where C!= 'r'
4	belajar...	bel-ajar
5	beC <sub>1</sub> erC <sub>2</sub> ...	be-C <sub>1</sub> erC <sub>2</sub> ... where C <sub>1</sub> != {'r'   'l'}
6	terV...	ter-V...   te-rV...
7	terCerV...	ter-CerV... where C!= 'r'
8	terCP...	ter-CP... where C!= 'r' and P!= 'er'
9	teC <sub>1</sub> erC <sub>2</sub> ...	te-C <sub>1</sub> erC <sub>2</sub> ... where C <sub>1</sub> != 'r'
10	me{l r w y}V...	me- {l r w y} V...
11	mem{b f v}...	mem- {b f v} ...
12	mempe{r l}...	mem-pe...
13	mem{rV V}...	me-m{rV V}...   me-p{rV V}...
14	men{c d j z}...	men- {c d j z}...
15	menV...	me-nV...   me-tV...
16	meng{g h q}...	meng- {g h q}...
17	mengV...	meng-V...   meng-kV...
18	menyV...	meny-sV...
19	mempV...	mem-pV... where V!= 'e'
20	pe{w y}V...	pe- {w y} V...
21	perV...	per-V...   pe-rV...
22	perCAP...	per-CAP... where C!= 'r' and P!= 'er'

<i>Rule</i>	<i>Construct</i>	<i>Return</i>
23	perCAerV...	per-CAerV... where C!= 'r'
24	pem{b f v}...	pem-{b f v}...
25	pem{rV V}...	pe-m{rV V}...   pe-p{rV V}...
26	pen{c d j z}...	pen-{c d j z}...
27	penV...	pe-nV...   pe-tV...
28	peng{g h q}...	peng-{g h q}...
29	pengV...	peng-V...   peng-kV...
30	penyV...	peny-sV...
31	peIV...	pe-IV... Exeption: for "pelajar" return ajar
32	peCerV...	per-erV... where C!={r w y l m n}
33	peCP...	pe-CP... where C!={r w y l m n} and P!= 'er'

V = vokal

C = konsonan

A = huruf apapun

P = fragmen kecil kata seperti "er"

Dalam langkah sebelumnya kata "membelikan" menjadi kata "membeli" dan sekarang akan dihilangkan kata "mem-" untuk mendapatkan kata "beli".

- c. Jika dalam kamus masih tidak ditemukan, maka ulangi langkah 4 ini secara berulang sampai menemukan akar katanya atau sampai melanggar kondisi 4a. Bila kondisi 4a terjadi, penghapusan awalan rekursif tidak berhasil karena tidak ada awalan yang berlaku untuk dihapus dan lanjutkan pada tahap selanjutnya.

5) Periksa kolom terakhir pada tabel 2.2 yang menunjukkan varian awalan dengan pengodean akar kata dimulai dengan huruf tertentu dan tidak semua imbuhan awalan yang memiliki karakter recoding.

Dari contoh kata “menangkap”, ada dua karakter recoding yang mungkin berdasarkan peraturan nomor 15, “n” (seperti dalam “men-nV...”) dan “t” (seperti dalam “men-tV...”). Ini merupakan hal yang luar biasa dikarenakan sebagian besar hanya terdapat satu karakter recoding. Ditemukan kata “nangkap” untuk perulangan pertama dan tidak ditemukan dalam kamus, maka kembali ke langkah 4 yang menghasilkan kata “tangkap” dan cek kembali ke dalam kamus. Proses berhenti karena kata “tangkap” adalah kata yang valid dalam kamus.

Bila semua proses di atas gagal, maka algoritma mengembalikan kata aslinya.

## 2.8. *Feature Weighting* WIDF

*Feature weighting* merupakan sebuah proses yang memberikan nilai pada setiap *feature* berdasarkan relevansi dan pengaruhnya terhadap hasil kategorisasi. Proses ini akan menghasilkan nilai bobot kata yang merupakan sebuah indikator untuk mengetahui tingkat kepentingan setiap kata dalam dokumen. Adapun metode *feature weighting* yang digunakan dalam penelitian ini adalah *Weighting Inverse Document Frequency* (WIDF).

Metode WIDF merupakan perkembangan dari metode IDF dimana kelemahan dari metode tersebut yakni bahwa semua dokumen yang mengandung istilah tertentu diperlakukan sama dengan perhitungan biner. Dengan penambahan



fitur frekuensi dan koleksi dokumen, metode WIDF telah teruji performansinya dan dapat dikatakan lebih baik dari *feature weighting* TF dan TF-IDF (Purnomo, 2010). Rumus metode WIDF ditunjukkan pada persamaan 2.1 berikut ini:

$$WIDF(d, t) = \frac{TF(d, t)}{\sum_{i \in D} TF(i, t)} \quad (2.1)$$

$d$  = dokumen

$t$  = kata

$i$  = dokumen berkesesuaian

Dimana  $TF(d, t)$  adalah munculnya suatu kata( $t$ ) dalam suatu dokumen( $d$ ) yang dibagi dengan  $TF(i, t)$  yaitu jumlah total kata( $t$ ) yang ada pada dokumen berkesesuaian( $i$ ).

### 2.9. *K-Nearest Neighbor*

Metode *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Metode ini termasuk algoritma *supervised learning* dimana hasil dari *query instance* yang baru diklasifikasi berdasarkan mayoritas dari kategori pada KNN. (Nugraha, 2014)

Tujuan dari algoritma KNN adalah untuk mengklasifikasi objek baru berdasarkan atribut dan *data sampling*. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Pada proses pengklasifikasian, algoritma ini tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru. (Krisdandi, 2013)

Adapun algoritma *K-Nearest Neighbor* dapat dijelaskan dengan keterangan berikut: (Kurniawan, 2012)

1. Menentukan nilai parameter k(jumlah tetangga terdekat).
2. Hitung jarak antara data yang akan dievaluasi dengan semua data pelatihan atau data sampel.
3. Urut naikkan jarak yang terbentuk dan tentukan jarak terdekat sampai urutan ke - k.
4. Pasangkan kategori atau kelas yang bersesuaian.
5. Cari jumlah terbanyak dari tetangga terdekat tersebut dan tetapkan kategori tersebut sebagai kategori dari data yang akan dicari.

Penelitian ini menggunakan algoritma *cosine similarity* untuk menghitung kedekatan jarak antara dokumen *data sampling* dan data uji pada metode KNN ini. *Cosine similarity* digunakan untuk menghitung pendekatan relevansi *query* terhadap dokumen. Penentuan relevansi sebuah *query* terhadap suatu dokumen dipandang sebagai pengukuran kesamaan antara vektor *query* dengan vektor dokumen. Semakin sama suatu vektor dokumen dengan vektor *query* maka dokumen dapat dipandang semakin sesuai dengan *query*.

Rumus yang digunakan untuk menghitung *cosine similarity* adalah sebagai berikut:

$$\text{cosSim}(X, d_j) = \frac{\sum_{i=1}^m x_i \cdot d_{ji}}{\sqrt{\sum_{i=1}^m x_i^2} \cdot \sqrt{\sum_{i=1}^m d_{ji}^2}} \quad (2.2)$$

$X$  = dokumen uji

$d_j$  = dokumen sampel

$x_i$  = nilai bobot *term i* pada dokumen uji

$d_{ji}$  = nilai bobot *term i* pada dokumen sample

Dimana perkalian bobot setiap *term* antar dokumen ( $x_i.d_{ji}$ ) dijumlahkan dan dibagi dengan perkalian jumlah bobot keseluruhan term pada dokumen uji dengan dokumen sampel.

## 2.10. Evaluasi

Parameter pengukuran digunakan untuk mengevaluasi performansi dari model yang telah dibuat dalam melakukan klasifikasi dengan benar atau tidak. Pengevaluasian dalam penelitian ini dilakukan dengan menggunakan *confusion matrix*. Menurut Visa (2011), *confusion matrix* merupakan informasi mengenai hasil klasifikasi aktual dan yang telah diprediksi oleh sistem klasifikasi menggunakan data dalam sebuah matriks. Tabel 2.3 menampilkan sebuah *confusion matrix* untuk pengklasifikasian ke dalam dua kelas.

**Tabel 2.3 Confusion Matrix (Novantirani, 2015)**

		<i>True Class</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Predicted Class</i>	<i>Positive</i>	<i>True Positive</i> (TP)	<i>False Positive</i> (FP)
	<i>Negative</i>	<i>False Negative</i> (FN)	<i>True Negative</i> (TN)

Matriks tersebut memiliki empat nilai yang dijadikan patokan dalam perhitungan, yaitu:

- a. *True Positive* (TP): Ketika kelas yang diprediksi positif dan faktanya positif.
- b. *True Negative* (TN): Ketika kelas yang diprediksi negatif dan faktanya negatif.
- c. *False Positive* (FP): Ketika kelas yang diprediksi positif dan faktanya negatif.
- d. *False Negative* (FN): Ketika kelas yang diprediksi negatif dan faktanya positif.

Selanjutnya dari matriks tersebut dapat ditarik kesimpulan performansi dari hasil klasifikasi berupa nilai *accuracy*. *Accuracy* merupakan rasio dari jumlah ketepatan prediksi tiap kelas terhadap jumlah total semua prediksi yang diklasifikasikan ke dalam kelas-kelas tersebut. Rumus *accuracy* ditunjukkan pada persamaan 2.3 berikut ini:

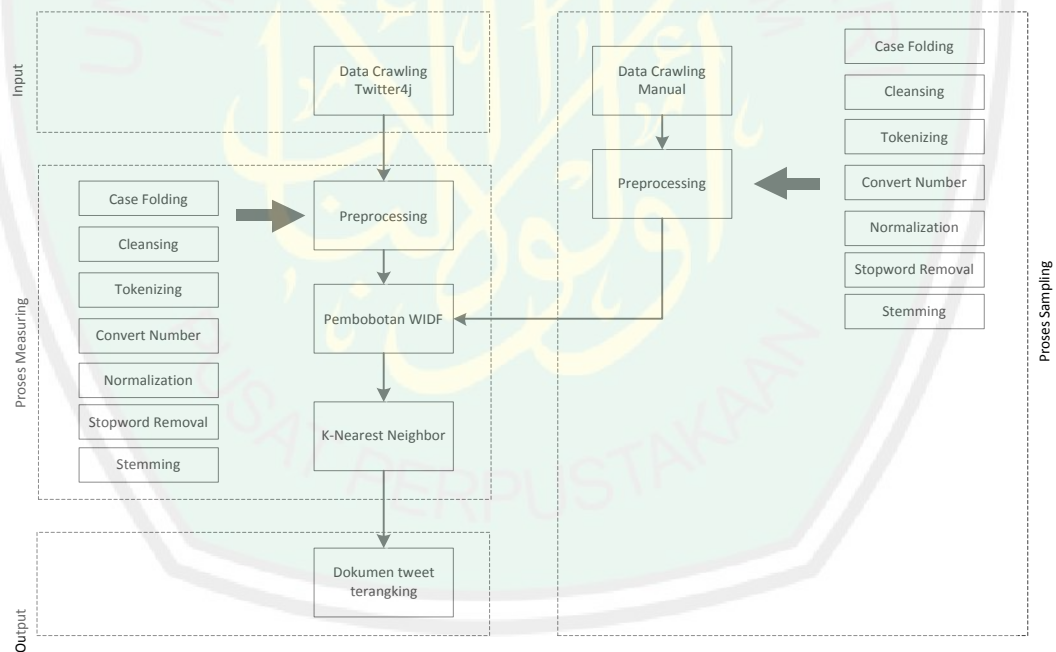
$$Accuracy(A) = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.3)$$

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Perancangan Sistem

Sistem yang sedang dikerjakan ini tentunya memiliki suatu rancangan bagaimana alur sistem ini nantinya akan berjalan. Sistem ini dibuat bertujuan agar dapat melakukan proses klasifikasi opini terhadap data teks menggunakan *K-Nearest Neighbor*. *Output* dari sistem ini merupakan *data testing* dengan nilai output berupa opini negatif atau positif yang ditentukan oleh sistem berdasarkan *data sampling* yang ada. Gambaran umum sistem yang akan dibuat dalam penelitian ini adalah sebagai berikut:



**Gambar 3.1 Diagram Alir Sistem**

Alur sistem yang ditunjukkan pada gambar 3.1 tersebut menggambarkan proses jalannya sistem dari awal *user* memasukkan data hingga *user* mendapatkan *output* informasi opini. Setiap tahapan pada alur sistem tersebut

akan dijelaskan pada subbab selanjutnya.

### 3.2. *Input*

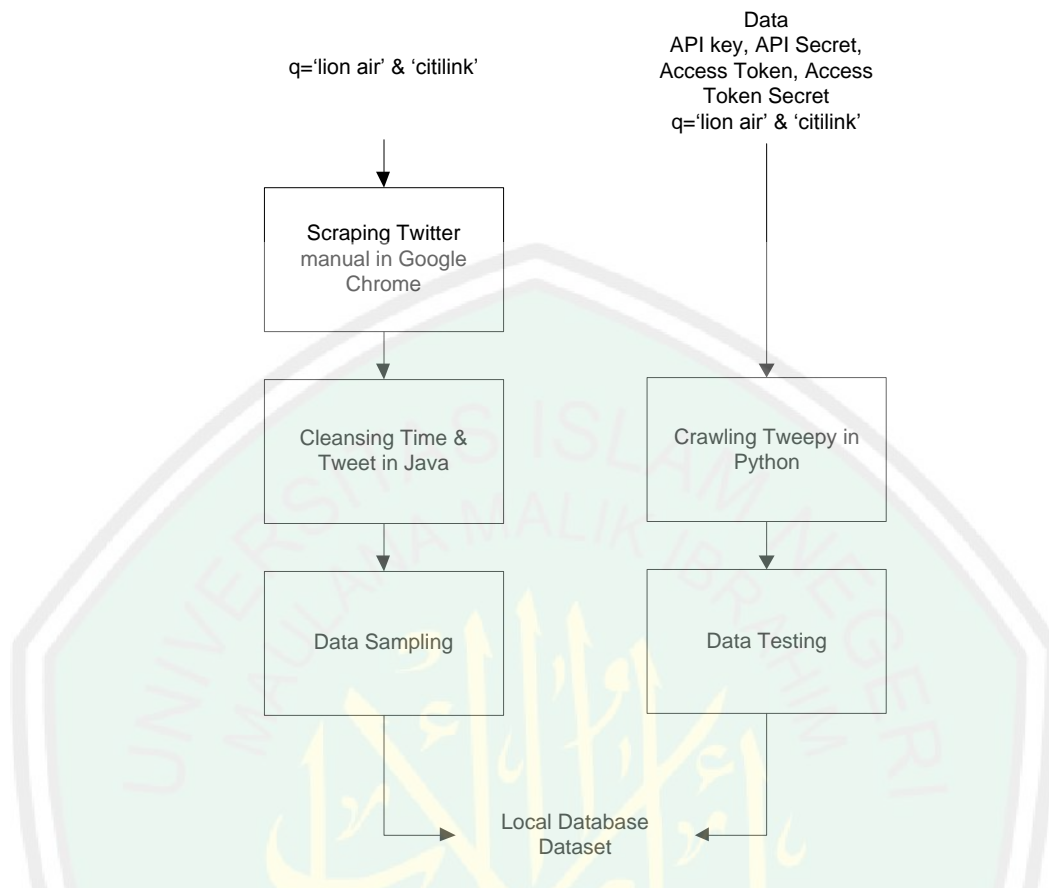
*Input* yang digunakan yaitu dokumen yang berupa tweet yang berisi kata-kata opini. Dokumen yang dimasukan disaring hanya yang berbahasa Indonesia seperti yang ditunjukkan pada tabel 3.1.

**Tabel 3.1 Dokumen Tweet**

Tweet	Opini
Ampun ya hari ini @citolink dari DPS-CGK udah delay 2 jam, turun di terminal 3, nunggu bagasi 1 jam pula, lengkaaaappp gilaaakkk beneerr....	N
Penerbangan pagi yg nyaman with QG117 @Citilink pic.twitter.com/STfn7iMJLK	P

### 3.3. *Dataset*

*Dataset* yang digunakan dalam penelitian ini diambil dari *website* <http://www.twitter.com/>. Data yang dihasilkan berupa *tweet* yang telah diambil seputar maskapai penerbangan di Indonesia dengan menggunakan *query* ‘lionair’ dan ‘citolink’ pada 1 September 2015 hingga 31 Agustus 2016.



**Gambar 3.2 Blok Diagram Persiapan *Dataset***

*Dataset* ini nantinya akan dipisah menjadi 2 bagian yaitu *data sampling* dan *data testing* yang dapat dilihat pada gambar 3.2. Untuk memperoleh data yang diperlukan dalam penelitian ini, digunakan aplikasi google chrome dalam pencarian *query* untuk persiapan data sampling dan data *API key*, *API secret*, *access token* dan *access token secret* yang bisa didapatkan dengan mendaftarkan aplikasi yang akan dibuat pada <https://apps.twitter.com/> guna persiapan *data testing*.

*Data sampling* dalam penelitian ini akan didapatkan dengan menggunakan *scraping* manual yang dapat diartikan dengan mengambil data *element* yang ada

pada hasil pencarian *query* dan waktu yang sudah ditentukan dan disalin kedalam file berekstensi *.txt*. Data pada *file .txt* tersebut masih berupa *tag* yang nantinya akan diproses lagi dalam proses *cleansing* untuk mendapatkan hasil waktu dan *tweet* yang tersdapat pada *file* tersebut. *Output* yang dihasilkan nantinya berupa *file* yang berekstensi *.txt* yang berisi waktu dan *tweet*, yang nantinya akan disaring secara manual kategori opini (positif/negatif) dan dikirim kedalam *database mysql* dengan ditambahkan label opini positif atau negatif dengan kata lain bahwasannya standar opini yang digunakan dalam penelitian ini berasal dari peneliti sendiri dengan persetujuan dosen pembimbing.

*Data testing* yang diperoleh pada penelitian ini berbeda dengan proses yang ada pada *data sampling*, cara yang digunakan *include* dalam sistem yang nantinya akan menggunakan **library tweepy** yang ada pada pemrograman bahasa python.

Pengambilan *data sampling* dan *data testing* dilakukan dengan cara berbeda. Hal tersebut dikarenakan untuk pengambilan *data sampling* dibutuhkan data 1 tahun kebelakang dan dilakukan secara manual yang dapat dilakukan untuk mengambil data sesuai jangka waktu 1 tahun tersebut, akan tetapi pada pengambilan *data testing* dengan menggunakan *library tweepy* hanya dapat mengambil data dengan batas jangka waktu 7 hari kebelakang.

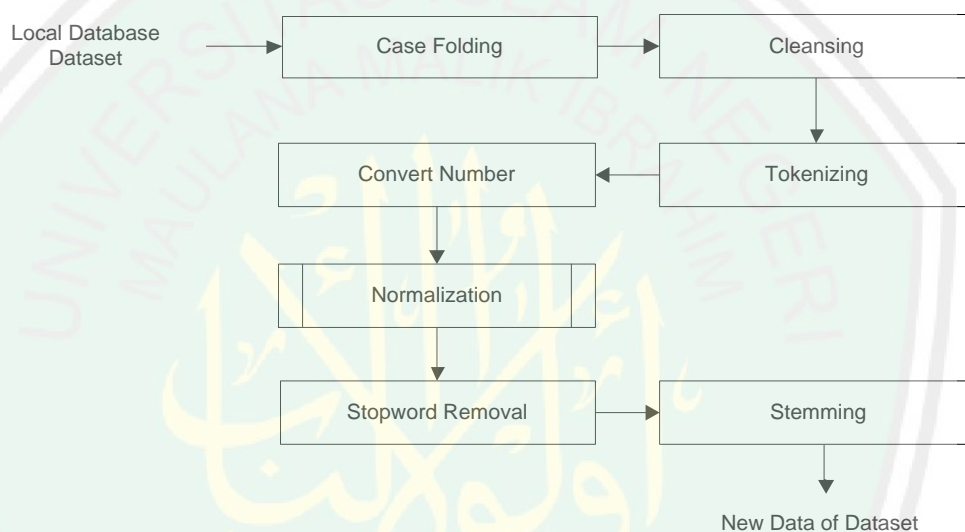
#### 3.4. *Preprocessing*

Proses ini dapat dikatakan sebagai kunci kualitas pada sistem penelitian ini, alasan utamanya adalah kualitas *output* analisis yang dihasilkan pada setiap penelitian *data mining* dipengaruhi oleh data yang dimasukkan. Dokumen yang ada pada data mentah masih terbilang kotor atau tidak sempurna, hal itu



dikarenakan terdapat *incomplete*, *noisy* ataupun *inconsistent*. Untuk mendapatkan dokumen yang berkualitas, data mentah nantinya akan diproses sedemikian rupa dengan harapan agar kualitas dari sistem ini dapat memuaskan.

Sistem pada penelitian ini menggunakan beberapa tahapan pada proses *preprocessing* ini, untuk lebih jelasnya gambar 3.3 akan menjelaskan setiap alur proses dari tahapan ini.



**Gambar 3.3 Blok Diagram Alir *Preprocessing***

#### 3.4.1. *Case Folding*

Pada proses tahapan *case folding* ini dilakukan perubahan semua huruf kapital pada dokumen menjadi huruf kecil. Tahapan ini dilakukan untuk menghindari redudansi kata, yaitu dua kata dianggap berbeda dikarenakan memiliki perbeda huruf kapital dan huruf kecil meskipun dua kata tersebut adalah kata yang sama. Contoh hasil tahapan ini dapat dilihat pada tabel 3.2.

Tabel 3.2 Contoh Tahapan *Case Folding*

<i>Input Process</i>	<i>Output Process</i>
Tolong ya di perbaiki pelayanannya Pesawat kok delaynya gak aturan Saya beli citilink 3x ga pernah tepat waktu	tolong ya di perbaiki pelayanannya pesawat kok delaynya gak aturan saya beli citilink 3x ga pernah tepat waktu

### 3.4.2. *Cleansing*

Tahapan ini dapat dikatakan juga sebagai pembersihan *noise* dalam *dataset*. *Noise* merupakan suatu bentuk data yang nantinya mungkin akan mengganggu proses pengolahan data tersebut. *Noise* yang dimaksud dalam penelitian ini adalah *mention*, *hashtag*, *link* dan karakter simbol maupun tanda baca. *Noise* kategori *mention* diawali dengan karakter ('@'), *hashtag* diawali dengan ('#') dan kategori *link* diawali dengan kata yang berformat ('http:', 'bit.ly'). Ketiga kategori tersebut dalam pembersihannya akan diproses satu kata hingga batas karakter spasi setelah kata tersebut, sedangkan *noise* berkategori karakter simbol (`~!@#$$%^&*()_+ -={}:~>?[];',./`) akan diproses hanya pada karakter simbol tersebut. Pembersihan *noise* dalam proses ini dilakukan dengan mengganti karakter *noise* menjadi karakter spasi (' '). Contoh hasil tahapan ini dapat dilihat pada tabel 3.3.

Tabel 3.3 Contoh Tahapan *Cleansing*

<i>Input Process</i>	<i>Output Process</i>
Tolong ya @Citilink di perbaiki pelayanannya. Saya beli citilink 3x ga pernah tepat waktu ;)	Tolong ya di perbaiki pelayanannya Saya beli citilink 3x ga pernah tepat waktu

### 3.4.3. *Tokenizing*

Tahap *tokenizing* ini melakukan pemotongan kata dalam dokumen menjadi potongan kata tunggal. Kata dalam dokumen yang dimaksud merupakan kata yang dipisah oleh karakter spasi, maka dari itu dalam proses tahapan ini sistem melakukan pengecekan terhadap karakter spasi. Tujuan dari tahapan ini agar nantinya dapat dilakukan penghitungan bobot dari setiap kata yang muncul, hasil yang didapatkan seperti pada tabel 3.4.

**Tabel 3.4 Contoh Tahapan *Tokenizing***

<i>Input Process</i>	<i>Output Process</i>
tolong ya di perbaiki pelayanannya	'tolong' 'ya' 'di' 'perbaiki'
pesawat kok delaynya gak aturan	'pelayanannya' 'pesawat' 'kok'
saya beli citilink 3x ga pernah	'delaynya' 'gak' 'aturan' 'saya' 'beli'
tepat waktu	'citilink' '3x' 'ga' 'pernah' 'tepat' 'waktu'

### 3.4.4. *Convert Number*

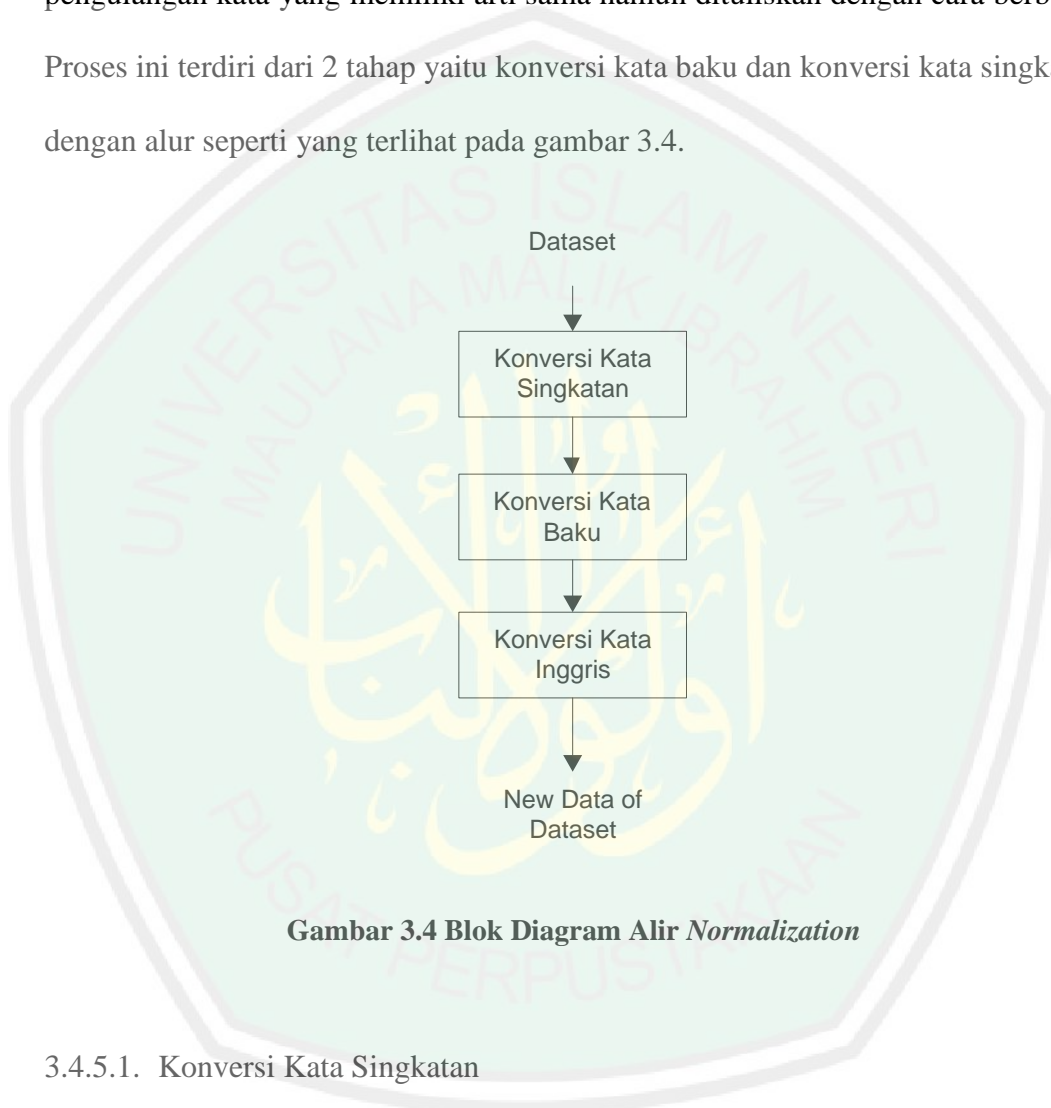
Tahap ini mengacu pada penelitian sebelumnya (Sunni, 2012) bahwa angka yang muncul di tengah-tengah atau belakang sebuah kata perlu diubah menjadi karakter huruf. Hal itu bertujuan untuk meminimalisir adanya huruf atau kata yang sengaja dirubah menjadi sebuah angka. Contoh hasil tahapan konversi angka ini dapat dilihat pada tabel 3.5.

**Tabel 3.5 Contoh Tahapan *Convert Number***

<i>Input Process</i>	<i>Output Process</i>
dear lionair delay kok nambah2	dear lionair delay kok nambah nambah
kaya orang maen di rental ps	kaya orang maen di rental ps

### 3.4.5. Normalization

Pada proses *normalization* ini dilakukan perubahan beberapa kata yang dianggap dapat memudahkan identifikasi terhadap suatu kata dan meminimalisir pengulangan kata yang memiliki arti sama namun dituliskan dengan cara berbeda. Proses ini terdiri dari 2 tahap yaitu konversi kata baku dan konversi kata singkatan dengan alur seperti yang terlihat pada gambar 3.4.



**Gambar 3.4** Blok Diagram Alir *Normalization*

#### 3.4.5.1. Konversi Kata Singkatan

Banyak dari pengguna twitter dalam beropini pada jejaring twitter masih menggunakan kata singkatan, hal itu disebabkan bisa jadi dikarenakan twitter membatasi hanya 140 karakter per tweet dan maksud dari opini pengguna twitter tersebut lebih dari batas tersebut. Pada proses ini dilakukan perubahan kata singkatan menjadi kata baku bahasa Indonesia dengan contoh hasil seperti pada tabel 3.6.

**Tabel 3.6 Contoh Tahapan Konversi Kata Singkatan**

<i>Input Process</i>	<i>Output Process</i>
thx capt ismail citiilink. setelah gagal mendarat 1x, akhirnya mendarat dgn baik. smua bertepuk tangan stlh cuaca buruk dan turbulensi hebat	thanks captain ismail citiilink. setelah gagal mendarat 1x, akhirnya mendarat dengan baik. semua bertepuk tangan setelah cuaca buruk dan turbulensi hebat

#### 3.4.5.2. Konversi Kata Baku

Tahap ini mempunyai tugas untuk mengkonversi logat kata bahasa daerah jawa menjadi kata berbahasa Indonesia, banyak orang yang menuliskan kata dengan menggunakan cara pengucapan/penulisan pada twitter dan salah satunya yaitu menggunakan logat bahasa jawa seperti contoh dalam tabel 3.7.

**Tabel 3.7 Contoh Tahapan Konversi Kata Baku**

<i>Input Process</i>	<i>Output Process</i>
@Citolink seneng bisa terbang bareng lagi bersama citilink..thanks citilink.. pic.twitter.com/WdTdpUibDF	@Citolink senang bisa terbang bareng lagi bersama citilink..thanks citilink.. pic.twitter.com/WdTdpUibDF

#### 3.4.5.3. Konversi Kata Inggris

Tahap ini bertugas melakukan perubahan kata yang berbahasa inggris menjadi kata berbahasa Indonesia, dikarenakan banyak orang yang sering mencampurkan kata bahasa Indonesia dengan bahasa inggris dalam opininya seperti contoh dalam tabel 3.8.

Tabel 3.8 Contoh Tahapan Konversi Kata Inggris

<i>Input Process</i>	<i>Output Process</i>
thanks captain ismail citiilink. setelah gagal mendarat 1x, akhirnya mendarat dengan baik. semua bertepuk tangan setelah cuaca buruk dan turbulensi hebat	terima kasih kapten ismail citiilink. setelah gagal mendarat 1x, akhirnya mendarat dengan baik. semua bertepuk tangan setelah cuaca buruk dan turbulensi hebat

#### 3.4.6. *Stopword Removal*

Tahap ini bertugas menghilangkan kata yang tidak relevan terhadap topik atau kata yang dapat mengganggu keakurasian sistem dalam pencapaian hasil opini. Daftar kata *stopword* telah disimpan dalam *database* sistem dan nantinya akan di bandingkan dengan *dataset*. Jika dalam *dataset* terdapat kata *stopword*, maka kata tersebut akan diganti menjadi karakter spasi. Contoh hasil dari tahapan ini dapat dilihat pada tabel 3.9.

Tabel 3.9 Contoh Tahapan *Stopword Removal*

<i>Input Process</i>	<i>Output Process</i>
ingatkan saya kalau lionair itu maskapai paling tidak usah dipilih utk liburan atau dinas, bisa ya seenaknya reschedule jam keberangkatan	ingatkan lionair usah dipilih liburan dinas, bisa seenaknya reschedule jam keberangkatan

### 3.4.7. *Stemming*

Pada tahap *stemming* ini menerapkan pengetahuan aturan dasar bahasa Indonesia dimana pendekatan algoritma dilakukan dengan:

- a) Kata yang terdiri dari 3 huruf atau kurang tidak dapat berisi afiks, sehingga tidak ada proses *stemming* yang dilakukan pada kata yang pendek.
- b) Afiks tidak pernah dilakukan pengulangan dalam penghapusannya, jadi dalam satu kali jalan harus dapat memproses semua afiks yang ada.
- c) Penggunaan batasan konfiks untuk menghindari kombinasi afiks yang tidak valid. Daftar kombinasi afiks yang tidak valid dapat dilihat pada tabel 2.1 pada bab 2.
- d) Jika karakter telah dikembalikan setelah awalan dihapus, dilakukan pengodean ulang jika diperlukan.

*Stemming* ini nantinya akan menjadi proses akhir dalam *preprocessing* dokumen, dimana output kata yang dihasilkan menjadi kata dasar.

### 3.5. Pembobotan WIDF

Proses ini merupakan proses yang mendukung untuk tahap klasifikasi selanjutnya dimana dibutuhkan nilai skor dalam setiap kata yang nantinya akan dihitung bobotnya. Sesuai dengan rumus 2.1, metode WIDF secara sederhana dapat dicontohkan seperti pada tabel 3.10, dimana  $d$  (kolom) adalah teks dan  $t$  (row) adalah *term* yang dicari.

**Tabel 3.10 Contoh Koleksi Data**

	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$
...	...	...	...	...	...
$t_x$	2	5	3	2	4
$t_y$	3	2	3	2	3
...	...	...	...	...	...

Angka yang ada pada tabel merupakan data frekuensi kata ( $t$ ) pada dokumen ( $d$ ), pada contoh kasus  $d_2$ , dapat dihitung nilai bobotnya dengan rumus 2.1 dan menghasilkan perhitungan sebagai berikut:

$$WIDF(d_2) = \frac{5}{2 + 5 + 3 + 2 + 4} = 0.3125$$

Bobot dari  $t_x$  pada  $d_2$  dapat dihitung dengan membagi jumlah  $t_x$  pada  $d_2$  dengan jumlah keseluruhan  $t_x$  pada semua kumpulan dokumen. Dengan kata lain WIDF merupakan bentuk normalisasi *term frequency* dari semua kumpulan dokumen. Dari perhitungan di atas didapatkan hasil pembobotan seperti pada tabel 3.11.

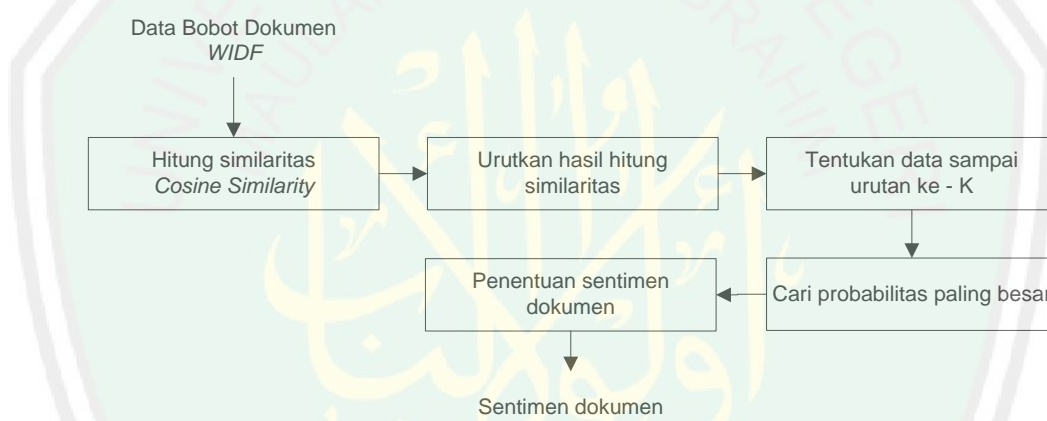
**Tabel 3.11 Hasil Bobot Koleksi Data**

	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$
...	...	...	...	...	...
$t_x$	0.125	0.3125	0.1875	0.125	0.25
$t_y$	0.2308	0.1538	0.2308	0.1538	0.2308
...	...	...	...	...	...



### 3.6. *K-Nearest Neighbor*

Dalam tahap klasifikasi ini digunakan metode *k-Nearest Neighbor*. Proses pada tahap ini dilakukan pengenalan pola pada *dataset sampling* yang sudah diklasifikasikan dalam label positif atau negatif. Kemudian model *dataset* tersebut digunakan pada teks yang belum berlabel dan yang akan diprediksi nilai opininya. Klasifikasi ini nantinya akan menghasilkan prediksi teks ke dalam label positif atau negatif. Alur tahapan klasifikasi KNN ini ditunjukkan pada gambar 3.5 berikut ini.



**Gambar 3.5 Blok Diagram Alir *K-Nearest Neighbor***

Metode ini menggunakan algoritma *cosine similarity* untuk pengukuran jarak dokumen. Pada proses perhitungan similaritas dengan menggunakan rumus 2.2 yang telah dijabarkan pada bab sebelumnya. Data uji nantinya akan dibandingkan dengan setiap data sampel yang ada dengan *input* hasil dari pembobotan kata yang dilakukan sebelum tahap ini. Dari gambar blok diagram 3.5 tahap pertama adalah mendapatkan *input* data bobot dokumen WIDF yang dimisalkan terdapat 5 data latih seperti pada tabel 3.12.

Tabel 3.12 Contoh Bobot WIDF Dokumen Latih

kata	$W(d1,t)$	$W(d2,t)$	$W(d3,t)$	$W(d4,t)$	$W(d5,t)$
sukses	1	0	0	0	0
waktu	0.333333	0	0	0.333333	0.333333
nyaman	0.5	0	0	0.5	0
alhamdulillah	0.5	0	0	0.5	0
kesal	0	1	0	0	0
telat	0	0.333333	0.333333	0	0.333333
panas	0	0.5	0	0	0.5
rugi	0	0.5	0	0	0.5
tiket	0	0.5	0.5	0	0
jadwal	0	1	0	0	0
bagasi	0	0	1	0	0
hilang	0	0	0.5	0	0.5
ganggu	0	0	1	0	0
rusak	0	0	0.5	0	0.5
batal	0	0	0.5	0	0.5
gembira	0	0	0	1	0
pantun	0	0	0	1	0

Nilai 0 pada suatu dokumen pada tabel 3.12 dapat diartikan bahwasannya dalam dokumen tersebut tidak terdapat kata yang ada pada kolom sebelah paling kiri. Dengan keseluruhan nilai yang ada, maka dilanjutkan dengan mengetahui nilai bobot dan opini yang telah diberikan pada setiap dokumen. Nilai bobot

dokumen dapat diketahui dengan menjumlahkan seluruh bobot kata, seperti halnya contoh perhitungan bobot pada dokumen yang pertama dibawah ini.

$$W(d_1) = 1 + 0.333 + 0.5 + 0.5 = 2.333$$

Pada tabel 3.13 ditunjukkan keseluruhan nilai bobot dokumen yang telah dilakukan proses perhitungan dan opini setiap dokumen.

**Tabel 3.13 Contoh Bobot dan Opini Dokumen Latih**

Dokumen	Bobot	Opini
D1	2.333	P
D2	3.833	N
D3	4.333	N
D4	3.333	P
D5	3.167	N

Selanjutnya dihitung nilai bobot kata dan bobot dokumen uji yang di proses seperti halnya pada dokumen latih dan didapatkan hasil nilai pada tabel 3.14.

**Tabel 3.14 Contoh Bobot WIDF Dokumen Uji**

kata	$W(x,t)$
sukses	0
waktu	0.25
nyaman	0.333
alhamdulillah	0
kesal	0
telat	0
panas	0

<b>kata</b>	<b><math>W(x,t)</math></b>
rugi	0.333
tiket	0
jadwal	0
bagasi	0
hilang	0
ganggu	0
rusak	0
batal	0
gembira	0.5
pantun	0

Dari bobot kata pada tabel 3.14, maka dapat diketahui nilai bobot dokumen uji tersebut yaitu 1.417 yang didapatkan dengan cara menjumlahkan seluruh nilai bobot kata.

Dilakukan perkalian bobot tiap kata yang ada pada dokumen latihan dengan bobot yang ada pada dokumen uji untuk mendapatkan nilai bobot pada dokumen, contoh untuk menghitung bobot baru pada D1 maka perkalian yang dilakukan yakni  $(1*0)$ ,  $(0.333*0.25)$ ,  $(0.5*0.333)$ ,  $(0.5*0)$ , ... . Bobot baru yang dihasilkan dapat dilihat pada tabel 3.15.

**Tabel 3.15 Contoh Bobot Baru Kata**

<b><math>W(d1,t)</math></b>	<b><math>W(d2,t)</math></b>	<b><math>W(d3,t)</math></b>	<b><math>W(d4,t)</math></b>	<b><math>W(d5,t)</math></b>
0	0	0	0	0
0.083333	0	0	0.083333	0.083333

$W(d1,t)$	$W(d2,t)$	$W(d3,t)$	$W(d4,t)$	$W(d5,t)$
0.166667	0	0	0.166667	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0.166667	0	0	0.166667
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0.5	0
0	0	0	0	0

Dari bobot kata pada tabel 3.15 tersebut, dihitung nilai bobot dengan menjumlahkan nilai bobot kata tiap dokumennya. Hasil nilai bobot dokumen dapat dilihat pada tabel 3.16.

**Tabel 3.16 Contoh Bobot Baru Dokumen Baru**

<b>Dokumen</b>	<b>Bobot</b>
D1	0.25
D2	0.167

Dokumen	Bobot
D3	0
D4	0.75
D5	0.25

Setelah nilai bobot yang diperlukan untuk perhitungan *cosine similarity* telah didapatkan, maka selanjutnya dihitung nilai kemiripan data dengan rumus 2.2 yang telah dijelaskan sebelumnya. Dicontohkan perhitungan untuk menghitung nilai kemiripan data pada D1 seperti berikut.

$$WIDF(d_2) = \frac{0.25}{\sqrt{1.417} \times \sqrt{2.333}} = 0.1375$$

Dimana nilai bobot baru D1 dibagi dengan perkalian akar kuadar bobot dokumen uji dan akar kuadrat bobot dokumen D1. Dengan demikian didapatkan nilai kemiripan data seluruh dokumen dengan rumus tersebut yang ditunjukkan pada tabel 3.17

**Tabel 3.17 Contoh Nilai Kemiripan Data Dokumen**

<i>Dataset</i>	Nilai Kemiripan Data
D1	0.1375
D2	0.0715
D3	0
D4	0.3451
D5	0.118

Tahap selanjutnya dilakukan pengurutan hasil perhitungan kemiripan data tersebut dari yang tertinggi hingga yang terendah yang ditunjukkan pada tabel 3.18.

**Tabel 3.18 Contoh Pengurutan Nilai Kemiripan Data**

Nomor	<i>Dataset</i>	Nilai Kemiripan Data
1	D4	0.3451
2	D1	0.1375
3	D2	0.0715
4	D5	0.118
5	D3	0

Tentukan nilai  $k$  sebagai parameter yang akan membatasi ketetanggaan terdekat. Dalam contoh ini parameter  $k=3$ , sehingga berdasarkan nilai tersebut ketetanggaan terdekat dengan data dapat dilihat pada tabel 3.19.

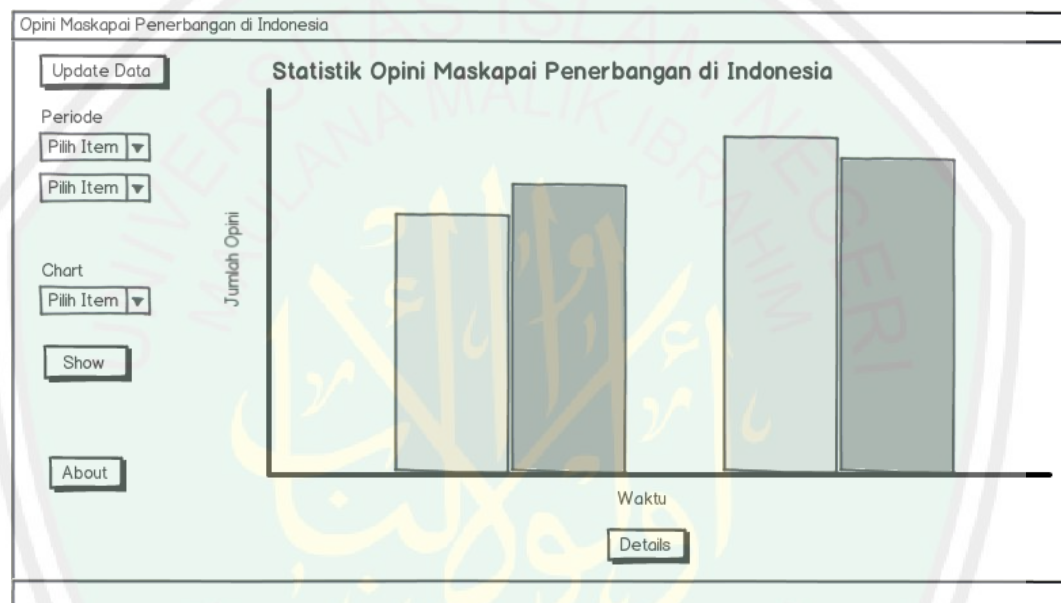
**Tabel 3.19 Contoh Hasil Pembatasan Ketetanggaan**

Nomor	<i>Dataset</i>	Nilai Kemiripan Data	Opini
1	D4	0,3415	P
2	D1	0,1375	P
3	D2	0,0715	N

Dapat dilihat dengan parameter  $k=3$ , data yang memiliki ketetanggaan terdekat ada pada  $D4$ (Positif),  $D1$ (Positif) dan  $D2$ (Negatif) dengan probabilitas opini yang terbanyak adalah positif. Maka dapat disimpulkan opini pada data uji tersebut adalah positif dikarenakan perbandingan nilai positif dan negatif yaitu 2:1.

### 3.7. Hasil Analisis Sistem

Sistem ini akan menghasilkan keluaran berupa grafik opini setiap *data testing* yang akan ditampilkan dalam tampilan GUI. Tampilan GUI yang dirancang berfungsi sebagai implementasi dari sistem akurasi dari metode yang digunakan menyesuaikan dengan tujuan yang dibuat. Pada gambar 3.6 ditampilkan kurang lebihnya rancangan tampilan aplikasi yang akan dibuat.



Gambar 3.6 Rancangan Tampilan GUI

### 3.8. Analisis Sistem

Analisa pada penelitian ini berfokus pada metode klasifikasi KNN yang membutuhkan nilai  $k$  sebagai penentu jarak kedekatan data sampel dan data uji. Nilai  $k$  ini nantinya akan berpengaruh pada hasil nilai keakurasian sistem ini, maka dari itu sebelum sistem ini dibangun nantinya akan ditentukan nilai  $k$  yang paling baik. Penentuan nilai  $k$  dalam penelitian ini akan dicoba dengan nilai 3, 5 dan 9. Untuk mengetahui mana yang kinerjanya paling baik dalam sistem ini,



maka *data sampling* nantinya akan dijadikan juga sebagai *data testing* dan nantinya akan diketahui nilai  $k$  mana yang akan dipakai dalam sistem ini.

Selain itu, sebelum masuk pada proses klasifikasi otomatis, dilakukan pencarian kata kunci yang terkait dengan sekumpulan tweet opini baik opini negatif maupun positif. Kata kunci ini terutama untuk menggambarkan aspek apa yang mendapat opini. Contoh kata kunci pada kumpulan tweet opini negatif yakni “tunda”, “batal” dan “asap”. Sedangkan kata kunci untuk opini positif diantaranya “rute”, “alhamdulillah” dan “halus”. Kata kunci tersebut berdasarkan pemilihan kata yang biasa digunakan dalam komentar pada twitter dan berdasarkan analisa penulis terhadap komentar hasil dari *crawling* yang telah dilakukan.

### 3.9. Sumber Data

Sumber data adalah segala sesuatu yang dapat memberikan informasi mengenai data. Berdasarkan sumbernya, data dibedakan menjadi dua, yaitu *data primer* dan *data sekunder*. Adapun sumber data dalam penelitian ini adalah sebagai berikut:

- 1) *Data primer* yaitu data yang dibuat untuk menyelesaikan permasalahan yang sedang ditanganinya. Data dalam penelitian ini dikumpulkan sendiri langsung dari <http://www.twitter.com/> berupa tweet dengan *query* ‘lionair’ dan ‘citolink’.
- 2) *Data sekunder* yaitu data yang telah dikumpulkan untuk maksud selain menyelesaikan masalah yang sedang dihadapi dan data ini dapat ditemukan dengan cepat. Dalam penelitian ini yang menjadi sumber data sekunder adalah literatur, artikel, jurnal serta situs di internet yang berkenaan dengan penelitian *text mining* dan *sentiment classification*.

## BAB IV

### UJI COBA DAN PEMBAHASAN

#### 4.1. Implementasi

Metode yang digunakan pada penelitian ini diimplementasikan menggunakan bahasa Java pada platform Java SE Development Kit(JDK) 1.7.0 dan IDE Netbeans 7.3 yang menggunakan MySQL sebagai *database server*. Aplikasi ini dibangun di atas platform Microsoft Windows 7 32bit dengan spesifikasi processor Intel Core i3-2330M dan memory 6144MB RAM dan menggunakan koneksi internet di jaringan 3.5G berkecepatan hingga 7.2 Mbps.

Implementasi algoritma dilakukan dengan membuat fungsi-fungsi dari tahapan yang telah dipaparkan pada bab 3 yang akan ditampilkan hasil disetiap langkahnya beserta potongan script yang penting.

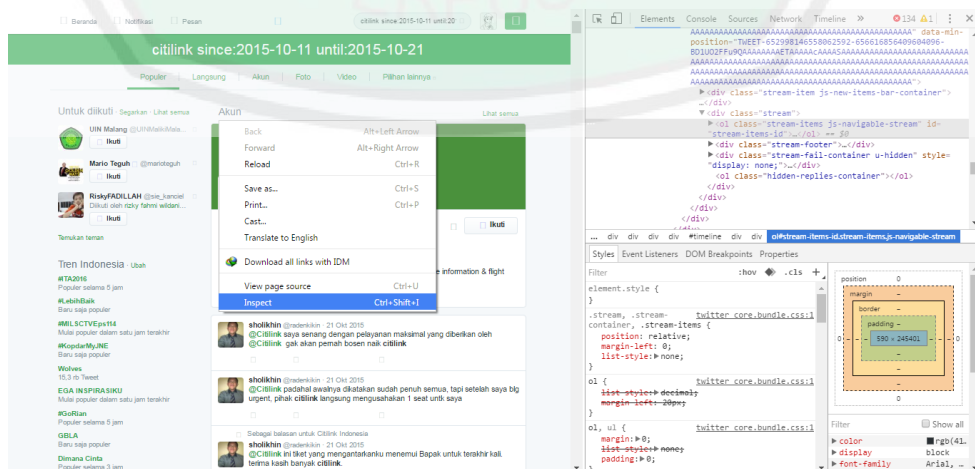
##### 4.1.1. Pengumpulan *Data Sampling*

Pengumpulan *data sampling* dilakukan dengan persiapan aplikasi browser google chrome dan internet yang stabil. Tahap pertama yang dilakukan adalah masuk sebagai *user* dalam *website* resmi twitter <http://twitter.com/> dengan memanfaatkan fitur pencarian di twitter menggunakan contoh *query* “*citilink* since:2015-09-01 until:2015-09-11” atau bisa juga dilakukan dengan menggunakan *link* “[https://twitter.com/search?q=citilink since%3A2015-09-01 until%3A2015-09-11&src=typd&lang=in](https://twitter.com/search?q=citilink%20since%3A2015-09-01%20until%3A2015-09-11&src=typd&lang=in)”. Dari tahap pencarian tersebut akan menghasilkan data *query* “*citilink*” dimulai dari tanggal 1 Oktober 2015 hingga 11 Oktober 2015 seperti contoh gambar 4.1.



Gambar 4.1 Tampilan Pencarian Query

Tahap kedua yang dilakukan adalah melakukan muat halaman web browser secara utuh, dalam artian lain adalah menampilkan secara lengkap tampilan dengan cara menarik layar browser hingga batas akhir sesuai kriteria waktu yang digunakan. Dapat diketahui batas akhir pada tahap ini adalah hingga browser tidak dapat ditarik kebawah lagi dengan syarat tanggal sudah sesuai dengan input yang digunakan.



Gambar 4.2 Tampilan Inspect Elemen Web

Tahap ketiga adalah melakukan *inspect element* pada web browser tersebut yang nantinya akan memunculkan tag “<ol class=...” yang dapat dilihat dalam gambar 4.2. dimana *tag* tersebut merupakan *tag* yang berisi data tweet-tweet yang telah di tampilkan dalam web browser. Salin *tag* tersebut dengan fitur *copy element* agar tag tersalin hingga akhir penutup *tag* dan tempel pada *file* yang berekstensi .txt.

*File* .txt tersebut akan diproses untuk mengambil data yang dibutuhkan yaitu waktu dan tweet dengan cara fungsi *cleansing* yang telah dibuat. Fungsi *cleansing* disini dibuat dengan tahapan mencari *tag* waktu dan dilanjutkan dengan *tag* tweet yang nantinya memberikan keluaran *file* berekstensi .txt yang berisi data waktu dan tweet yang akan dimasukkan dalam *database server* dan hasil pengkategorian dari *data sampling* mulai 1 September 2015 hingga 31 Agustus 2016 dapat dilihat dalam tabel 4.1.

**Tabel 4.1 Rincian Jumlah Data Sampling**

<i>Query</i>	<i>Opini Positif</i>	<i>Opini Negatif</i>	<i>Jumlah Opini</i>
Citilink	1185	1319	2504
Lionair	519	1319	1838

#### 4.1.2. Pengumpulan *Data Testing*

Data yang digunakan sebagai *data testing* diambil secara otomatis pada aplikasi ini yang telah menyediakan fitur untuk mengambil data pada jejaring sosial twitter menggunakan paket tweepy pada pemrograman bahasa Python. Penggunaan fitur ini dilakukan dengan beberapa kebutuhan pada sistem komputer

yang akan digunakan seperti koneksi internet, aplikasi python dan penggunaan kode akses token oleh *user*.

Dalam fitur ini memiliki beberapa tahapan untuk mendapatkan *output* data yang diinginkan, tahapan awal adalah pengecekan koneksi internet. Koneksi dalam hal ini dilakukan dengan cara pengecekan koneksi terhadap jejaring media sosial twitter, apabila koneksi pada sistem komputer dapat terhubung maka akan dilanjutkan dalam tahap selanjutnya. Apabila koneksi tidak terhubung, maka sistem aplikasi akan otomatis berhenti dan memberitahukan *user* jika koneksi internet pada sistem komputer tidak terhubung dengan jejaring media sosial twitter. Berikut fungsi untuk melakukan pengecekan koneksi yang ditampilkan pada gambar 4.3.

```
String cmd = "ping -n 1 www.twitter.com";  
Process myProcess = Runtime.getRuntime().exec(cmd);  
myProcess.waitFor();  
if (myProcess.exitValue() == 0) {  
    return true;  
} else {  
    return false;  
}
```

**Gambar 4.3 Fungsi Cek Koneksi Internet**

Selanjutnya akan dilakukan pengecekan aplikasi python pada sistem komputer. Jika aplikasi python masih belum ada pada saat menjalankan fitur ini, maka aplikasi akan meminta konfirmasi *user* untuk melakukan instalasi aplikasi python yang telah disediakan oleh aplikasi yang secara otomatis akan mengeksekusi instalasi aplikasi python. Fungsi untuk pengecekan aplikasi python pada sistem komputer dapat dilihat pada gambar 4.4.

```
String cmd = "python --version";
Process myProcess = Runtime.getRuntime().exec(cmd);
myProcess.waitFor();
if (myProcess.exitValue() == 0) {
    return true;
} else {
    return false;
}
```

**Gambar 4.4 Fungsi Cek Aplikasi Python**

Pada gambar 4.5 dapat dilihat fungsi tersebut untuk menjalankan *file installer* python yang berada pada folder sistem aplikasi. Aplikasi python yang disediakan yaitu python 2.7.10.

```
[1] String cmd = Paths.get("").toAbsolutePath().toString() +
    "\\python-2.7.10.msi";
[2] Runtime.getRuntime().exec("msiexec /i " + cmd);
```

**Gambar 4.5 Fungsi Menjalankan File *Installer* Python**

Setelah aplikasi python telah terinstall maka aplikasi akan melakukan crawling dengan *file* berekstensi .py pada folder sistem yang telah ada dan akan menghasilkan *output* 2 buah *file* berekstensi .csv yakni citilink.csv dan lionair.csv. *File* csv tersebut akan diproses untuk diinputkan kedalam *database server* agar memudahkan pada proses selanjutnya.

#### 4.1.3. Pengolahan Data pada *Database Server*

Penggunaan *database server* pada aplikasi ini sangat diperlukan untuk memudahkan proses pengolahan data didalamnya. *Database server* yang digunakan dalam penelitian ini menggunakan MySQL dan begitu juga bagi *user*

yang akan menggunakan aplikasi ini harus menggunakan *database server* MySQL pada sistem komputernya.

Pada penelitian ini menggunakan 11 tabel dalam *database server*, salah satu contoh tabel pada penelitian ini yaitu tabel ‘acuan’ yang berfungsi sebagai acuan manakah data yang termasuk dalam maskapai citilink ataupun lionair. Berikut struktur tabel acuan pada tabel 4.2. Tabel ini berisi data nama maskapai yang sedang menjadi bahan penelitian pada aplikasi ini yang nantinya akan memiliki beberapa relasi terhadap tabel yang lain.

**Tabel 4.2 Struktur Tabel Acuan**

<i>Field</i>	<i>Type</i>	<i>Extra</i>
maskapai_id	<i>int(11)</i>	<i>auto_increment</i>
maskapai_nama	<i>text</i>	

Tabel selanjutnya yaitu tabel ‘datalatih’ yang difungsikan sebagai tempat data latih baik data dari maskapai citilink maupun lionair. *Data field* yang digunakan mulanya dalam tabel ini yaitu ‘maskapai\_id’, ‘datalatih\_waktu’, ‘datalatih\_tweet’ dan ‘datalatih\_opini’. Untuk *data field* lain nantinya akan diperbarui dalam tahapan pembobotan dan klasifikasi. Struktur tabel datalatih dapat dilihat pada tabel 4.3.

**Tabel 4.3 Struktur Tabel Datalatih**

<i>Field</i>	<i>Type</i>	<i>Extra</i>
datalatih_id	<i>int(8)</i>	<i>auto_increment</i>
maskapai_id	<i>int(11)</i>	
datalatih_waktu	<i>date</i>	

<i>Field</i>	<i>Type</i>	<i>Extra</i>
datalatih_tweet	<i>text</i>	
datalatih_opini	<i>char(1)</i>	
datalatih_bobot	<i>double</i>	
datalatih_panjang	<i>int(4)</i>	
opini_tiga	<i>char(1)</i>	
opini_lima	<i>char(1)</i>	
opini_sembilan	<i>char(1)</i>	

Pada *field* ‘datalatih\_bobot’ dan ‘datalatih\_panjang’ nantinya akan diberikan data sesuai dengan perhitungan pada bagian pembobotan WIDF serta *field* ‘opini\_tiga’, ‘opini\_lima’ dan ‘opini\_sembilan’ digunakan sebagai tampungan hasil opini yang dihasilkan oleh sistem menggunakan algoritma *cosine similarity* untuk mengetahui tingkat akurasi dari setiap *range* yang digunakan.

Pada tabel 4.4 terdapat struktur tabel ‘datauji’ dimana tabel ini akan digunakan sebagai tampungan data uji yang dikumpulkan menggunakan *crawling* python pada aplikasi ini. Data yang ada pada tabel ini nantinya akan ditampilkan dalam grafik waktu agar user lebih mudah untuk membaca statistik opini maskapai pada aplikasi.

**Tabel 4.4 Struktur Tabel Datauji**

<i>Field</i>	<i>Type</i>	<i>Extra</i>
datauji_id	<i>int(8)</i>	<i>auto_increment</i>
maskapai_id	<i>int(11)</i>	
datauji_waktu	<i>date</i>	



<i>Field</i>	<i>Type</i>	<i>Extra</i>
datauji_tweet	<i>text</i>	
datauji_opini	<i>char(1)</i>	

Sistem aplikasi ini menggunakan perhitungan bobot kata agar dapat menghasilkan opini yang sesuai dengan klasifikasi kNN. Dapat dilihat pada tabel 4.5 dan tabel 4.6, kedua tabel tersebut merupakan struktur yang nantinya akan digunakan sebagai penampung hasil dari perhitungan bobot kata yang dilakukan menggunakan metode WIDF dengan rumus 2.1 pada bab sebelumnya.

**Tabel 4.5 Struktur Tabel dtlatih\_kata**

<i>Field</i>	<i>Type</i>	<i>Extra</i>
dtlatih_kata_id	<i>int(8)</i>	<i>auto_increment</i>
maskapai_id	<i>int(11)</i>	
dtlatih_kata_kata	<i>text</i>	
dtlatih_kata_frekuensi	<i>int(8)</i>	

**Tabel 4.6 Struktur Tabel dtlatih\_bobotkata**

<i>Field</i>	<i>Type</i>	<i>Extra</i>
dtlatih_bobotkata_id	<i>int(8)</i>	<i>auto_increment</i>
maskapai_id	<i>int(11)</i>	
dtlatih_kata_id	<i>int(8)</i>	
datauji_datalatih_id	<i>int(8)</i>	
dtlatih_bobotkata_bobot	<i>double</i>	

#### 4.1.4. Proses *Preprocessing* Dokumen

Pada setiap data tweet sebelum masuk dalam penilaian bobot, terlebih dahulu dilakukan *preprocessing* untuk menjadikan data tersebut lebih mudah di proses dan menghilangkan *noisy* yang ada agar menghasilkan sistem yang memuaskan. Proses ini memiliki beberapa tahapan yaitu *cleansing*, *case folding*, *tokenizing*, *convert number*, *normalization*, *stopword removal* dan *stemming*.

##### a. *Case Folding*

Tahapan *case folding* dalam penelitian ini mengubah huruf kapital menjadi huruf kecil, fungsi yang dibuat dapat dilihat pada gambar 4.6.

```
dataUji = dataUji.toLowerCase();
```

**Gambar 4.6 Fungsi *Case Folding* Dokumen Tweet**

##### b. *Cleansing*

Tahapan *cleansing* dilakukan untuk membersihkan data tweet dari angka, tanda baca, *link*, *hashtag* dan *mention* atau dengan kata lain hanya akan menggunakan data berupa huruf yang diolah dalam penelitian ini. Kata yang termasuk dalam daftar kata yang akan dihapus, nantinya akan diganti dengan karakter spasi. Fungsi tahapan ini ditunjukkan pada gambar 4.7.

```
if ((dataUji.contains("@") || dataUji.contains(",")) == true) {
    dataUji = dataUji.replaceAll("@.*?(?=$)", "");
    dataUji = dataUji.replaceAll(",", " ");
}
}
```

**Gambar 4.7 Fungsi *Cleansing* Dokumen Tweet**

Pada gambar 4.7 dimisalkan untuk menghapus *mention* dan tanda baca koma, untuk halnya *mention* (“@”), *link* (“http:”, “https:”, “pic.twitter.com”) dan *hashtag* (“#”) akan dihapus seluruh kata yang mengikuti tanda baca tersebut. Selain itu untuk tanda baca yang lain seperti titik(“.”), koma(“,”) dan lain sebagainya hanya akan dihilangkan tanda baca tersebut tanpa kata yang mengikuti.

### c. *Tokenizing*

*Tokenizing* memisahkan setiap kata dengan acuan karakter spasi, fungsi yang digunakan dapat dilihat pada gambar 4.8.

```
String []token = dataUji.split("[\\s'"]");
resize(token);
```

**Gambar 4.8 Fungsi *Tokenizing* Dokumen Tweet**

```
int n = 0;
for (int i = 0; i < token.length; i++) {
    if (token[i].equals("")) {
        n = n + 1;
    }
}
List<String> list = new ArrayList<String>(Arrays.asList(token));
list.removeAll(Arrays.asList(""));
token = list.toArray(token);
token = Arrays.copyOf(token, (token.length - n));
```

**Gambar 4.9 Fungsi *Resize* Dokumen Tweet**

Dalam fungsi ini, tahapan yang dilakukan dimulai dengan memisahkan kata dan dilanjutkan dengan penghapusan data dalam array yang berisi data kosong atau spasi dan dilakukan pengurutan ulang agar panjang array sesuai dengan data yang ada.

#### d. Convert Number

Konversi angka ini hanya diproses jika pada sebuah kata tersebut terdapat angka 2. Proses yang dilakukan yakni mengganti angka 2 tersebut dengan kata sebelum angka 2 tersebut. Berikut fungsi dari konversi angka 2 pada gambar 4.10.

```
int n = 0;
for (int i = 0; i < token.length; i++) {
    if (token[i].equals("")) {
        n = n + 1;
    }
}
List<String> list = new ArrayList<String>(Arrays.asList(token));
list.removeAll(Arrays.asList(""));
token = list.toArray(token);
token = Arrays.copyOf(token, (token.length - n));
```

**Gambar 4.10 Fungsi *Convert Number* Dokumen Tweet**

Dilanjutkan dengan fungsi *resize* array untuk pengurutan ulang agar panjang array sesuai dengan data yang ada seperti pada gambar 4.11.

```
List<String> list = new ArrayList<String>(Arrays.asList(token));
for (int i = 0; i < token.length; i++) {
    if (token[i].contains(" ")) {
        String[] tmp = token[i].split("[\\s]");
        list.set(i, tmp[0]);
        list.add(i + 1, tmp[1]);
    }
}
token = (String[]) list.toArray(new String[list.size()]);
```

**Gambar 4.11 Fungsi *Resize* Dokumen Tweet**

#### e. Normalization

Tahapan ini mengganti kata yang memiliki arti sama dengan kata yang lain, kata yang dimaksud disini adalah kata singkatan, kata baku dan kata inggris. Salah satu fungsi tahap normalisasi ini dapat dilihat pada gambar 4.12 dimana kata yang terindikasi sebagai kata singkatan yang terdapat dalam *database server* pada

kolom 'singkatan\_kata', akan diganti dengan kata pengganti yang ada pada kolom 'singkatan\_hasil'.

```

for (int i = 0; i < token.length; i++) {
    Statement stat = sambung.createStatement();
    String sql = "select * from dt_singkatan where singkatan_kata='" +
token[i] + "'";
    ResultSet rs = stat.executeQuery(sql);
    while (rs.next()) {
        String singkatanHasil = rs.getString("singkatan_hasil");
        if (singkatanHasil != null) {
            token[i] = singkatanHasil;
        }
    }
}
rs.close();
stat.close();
}

```

**Gambar 4.12 Fungsi Normalisasi Kata Singkatan Dokumen Tweet**

*Database* pada tahapan ini menggunakan data yang dibuat sendiri menyesuaikan dengan *data sampling* yang telah ada. Fungsi ini menggunakan *database* dengan struktur 3 kolom seperti pada tabel 4. .

**Tabel 4.7 Struktur Database dt\_singkatan**

<i>Field</i>	<i>Type</i>	<i>Extra</i>
singkatan_id	<i>int(4)</i>	<i>auto_increment</i>
singkatan_kata	<i>text</i>	
singkatan_hasil	<i>text</i>	

Untuk tahap normalisasi kata baku dan kata inggris, memiliki alur yang sama dengan normalisasi kata singkatan di atas, perbedaannya hanya pemanggilan tabel dan kolom yang menyesuaikan tabel pada *database server*.

#### f. *Stopword Removal*

Tahapan ini menghilangkan kata yang sekiranya tidak dibutuhkan oleh sistem yang jika tidak dihilangkan akan mempengaruhi sistem. Fungsi ini menggunakan method yang dijelaskan pada gambar 4.13.

```

for (int i = 0; i < token.length; i++) {
    if (token[i].length() < 3) {
        token[i] = "";
    }
    try {
        Statement stat = sambung.createStatement();
        String sql = "select * from dt_stopword where stopword_kata='" +
token[i] + "'";
        ResultSet rs = stat.executeQuery(sql);
        while (rs.next()) {
            String stopwordKata = rs.getString("stopword_kata");
            if (stopwordKata != null) {
                token[i] = "";
            }
        }
        rs.close();
        stat.close();
    } catch (Exception e) {
        System.out.println("Error stopwordRemoval!!\n" + e);
    }
}
resize(token);

```

**Gambar 4.13 Fungsi *Stopword Removal* Dokumen Tweet**

Tahapan ini menggunakan *database* sebagai tempat tampungan kata yang nantinya akan dihapus dengan struktur kolom pada tabel 4.8. Jika pada data tweet ditemukan kata pada kolom *stopword\_kata*, nantinya akan diubah dengan kata spasi dan dilanjutkan *resize* data untuk menyesuaikan panjang array data.

**Tabel 4.8 Struktur *Database dt\_stopword***

<b>Field</b>	<b>Type</b>	<b>Extra</b>
stopword_id	<i>int(4)</i>	<i>auto_increment</i>
stopword_kata	<i>text</i>	

### g. *Stemming*

*Stemming* merupakan konversi kata ke dalam bentuk kata dasar. Dalam penelitian ini tahap *stemming* menggunakan algoritma nazief adriani yang memerlukan adanya kamus kata dasar. Kamus kata dasar dalam penelitian ini berjumlah 28528 kata. Salah satu contoh tahapan didalamnya yakni penghapusan infleksional suffiks, pada gambar 4.14 ditunjukkan fungsi tersebut.

```

if (kata.endsWith("lah") || kata.endsWith("kah") ||
kata.endsWith("tah") || kata.endsWith("pun")) {
    kata = kata.substring(0, kata.length() - 3);
}
if (kata.endsWith("ku") || kata.endsWith("mu")) {
    kata = kata.substring(0, kata.length() - 2);
} else if (kata.endsWith("nya")) {
    kata = kata.substring(0, kata.length() - 3);
}
cekKamus(kata)

```

**Gambar 4.14 Fungsi Penghapusan Infleksional Suffiks**

Langkah pertama dilakukan dengan cara pengecekan akhiran kata “lah”, “kah”, “tah” dan “pun”, jika ditemukan akhiran tersebut pada kata yang diproses maka akhiran tersebut dihilangkan. Selanjutnya dilakukan pengecekan kembali untuk akhiran kata “ku”, “mu” dan “nya” sebagai tanda kepemilikan pada suatu kata. Jika ditemukan akhiran tersebut, maka akhiran tersebut dihapus dan dilanjutkan dengan pengecekan kata yang telah diproses pada kamus kata dasar dalam *database server*. Jika kata telah ditemukan dalam kamus kata dasar, proses berhenti dan jika tidak maka dilanjutkan dengan tahapan-tahapan *stemming* selanjutnya.

#### 4.1.5. Pembobotan WIDF

Tahap selanjutnya setelah *preprocessing* adalah pembobotan WIDF. Pembobotan ini dilakukan dengan cara menghitung TF(*term frequency*) dari masing-masing kata pada dokumen dan seluruh dokumen. Implementasi dari perhitungan TF dilakukan dengan fungsi yang dibuat pada gambar 4.15. Fungsi perhitungan frekuensi kata pada dokumen akan dilakukan dengan fungsi 4.15 dengan menyesuaikan variabel yang ada.

```
Collections.frequency(kumpulanKataDokumen, string);
```

**Gambar 4.15 Fungsi Bobot TF Dokumen**

Variabel *kumpulanKataDokumen* berisi data seluruh kata yang ada pada suatu dokumen dan sedangkan untuk variabel *string* yaitu kata yang akan diproses dan dicari nilai frekuensinya pada dokumen tersebut. Hasil pada tahap ini dapat dilihat pada tabel 4.9.

**Tabel 4.9 Daftar Nilai TF Dokumen**

maskapai_id	dokumen_id	kata	frekuensi
1	1	nyaman	2
1	1	tepat	1
2	34	protes	1
2	36	telat	1

Dalam proses pembobotan ini tidak hanya dilakukan perhitungan frekuensi pada suatu dokumen melainkan juga dilakukan perhitungan frekuensi pada keseluruhan dokumen yang ada. Untuk melakukan hal tersebut, dapat dilakukan



dengan fungsi yang ada pada gambar 4.15 dengan sedikit perubahan pada variabel yang digunakan.

```

for (String string : kataSample) {
    try {
        Statement stat = sambung.createStatement();
        String sql = "insert into `dtlatih_kata` (maskapai_id,
dtlatih_kata_kata, dtlatih_kata_frekuensi)VALUES (" + maskapaiID + ",
'" + string + "', " + Collections.frequency(kumpulanKataSample,
string) + "));";
        stat.executeUpdate(sql);
        stat.close();
    } catch (Exception e) {
        System.out.println("Error insertKata!!\n" + e);
    }
}

```

**Gambar 4.16 Fungsi Bobot TF Seluruh Dokumen**

Variabel yang dimaksud disini yaitu variabel kumpulanKataDokumen diganti dengan variabel lain yang berisi data kata pada keseluruhan dokumen dan sedikit tambahan *source code* seperti pada gambar 4.16. Setelah dijalankan fungsi tersebut nantinya akan menghasilkan data keluaran seperti pada tabel 4.10.

**Tabel 4.10 Daftar Nilai TF Seluruh Dokumen**

dtlatih_kata_id	maskapai_id	kata	frekuensi
1	1	nyaman	46
2	1	tepat	8
3	2	protes	10
4	2	telat	29

Perhitungan bobot dilanjutkan jika kedua nilai TF sudah didapatkan, dengan rumus 2.1 nantinya akan menghasilkan keluaran berupa nilai bobot kata pada

setiap dokumen yang ada. Hasil yang didapatkan nantinya seperti yang ada pada tabel 4.11.

**Tabel 4.11 Hasil Perhitungan WIDF**

<b>dtlatih_bobot_id</b>	<b>maskapai_id</b>	<b>kata</b>	<b>frekuensi</b>
1	1	nyaman	0.043
2	1	tepat	0.125
3	2	protes	0.1
4	2	telat	0.034

#### 4.1.6. Kemiripan Data dengan *Cosine Similarity*

Penentuan opini dilakukan dengan cara menghitung *cosine similarity* antara dokumen uji dengan masing-masing dokumen latih. Perhitungan ini didasarkan pada tahap sebelumnya yaitu pembobotan kata dengan metode WIDF. Pada implementasinya, hal ini dilakukan dengan alur membandingkan kedekatan antara matriks dokumen dengan matriks masing-masing dokumen.

Dokumen uji yang akan ditentukan opininya, dilakukan proses *preprocessing* dan pembobotan seperti halnya yang dilakukan pada dokumen latih. Dokumen uji disimpan pada *database server* sebelum dilakukan tahap-tahap penentuan opini. Contoh dokumen uji sebagai berikut:

“Alhamdulillah nyaman dan waktunya tepat @Citilink QG.986 SUB-BDO.”

Dilakukan tahap *preprocessing* pada dokumen tersebut dan kata yang dihasilkan alhamdulillah, nyaman, tepat, waktu. Kemudian dilakukan perhitungan bobot kata menggunakan metode WIDF yang menghasilkan seperti pada tabel 4.12.

Tabel 4.12 Daftar Bobot Dokumen Uji

dtlatih_bobot_id	maskapai_id	kata	bobot
1	1	alhamdulillah	0.25
2	1	nyaman	0.333
3	1	waktu	0.333
4	1	tepat	0.5

Langkah selanjutnya dihitung jumlah perkalian bobot uji dengan tiap dokumen latih. Nilai *cosine similarity* sendiri didapatkan dengan cara membagi jumlah perkalian bobot dengan perkalian vektor dokumen uji dengan vektor dokumen latih.

Implementasi perhitungan perkalian bobot dokumen uji dengan tiap dokumen latih dapat dilihat dalam tabel 4.13, dijelaskan bahwasannya terdapat 5 dokumen latih yang nantinya akan menghasilkan nilai kedekatan tiap dokumen tersebut dengan dokumen uji.

Tabel 4.13 Daftar Bobot *Query* Dokumen Latih

R(Q,d1)	R(Q,d2)	R(Q,d3)	R(Q,d4)	R(Q,d5)
0.083	0	0	0.083	0.083
0.167	0	0	0.167	0
0	0.167	0	0	0.167
0	0	0	0.5	0

Setelah didapatkan keluaran nilai seperti pada tabel 4.11, dihitung jumlah bobot tiap dokumen seperti pada tabel 4.14.

**Tabel 4.14 Daftar Jumlah Bobot *Query* Dokumen Latih**

<b>R(Q,d1)</b>	<b>R(Q,d2)</b>	<b>R(Q,d3)</b>	<b>R(Q,d4)</b>	<b>R(Q,d5)</b>
0.25	0.167	0	0.75	0.25

Langkah selanjutnya dihitung perkalian vektor dokumen uji dengan tiap dokumen latih menggunakan algoritma *cosine similarity*, fungsi yang digunakan dapat dilihat pada gambar 4.17.

```
Math.sqrt (bobotLatihDoc.get (jj)) * Math.sqrt (bobotUjiDoc)
```

**Gambar 4.17 Fungsi Perkalian Vektor**

Perkalian vektor antara dokumen uji dan latih menggunakan fungsi sqrt, dimana variabel bobotLatihDoc berisi nilai bobot dokumen latih dan variabel bobotUjiDoc berisi nilai dokumen uji. Setelah dilakukan perhitungan, didapatkan nilai dari tiap dokumen seperti pada tabel 4.15.

**Tabel 4.15 Bobot Perkalian Vektor**

<b>R(Q,d1)</b>	<b>R(Q,d2)</b>	<b>R(Q,d3)</b>	<b>R(Q,d4)</b>	<b>R(Q,d5)</b>
1.818	2.330	2.477	2.173	2.118

Perhitungan *cosine similarity* ini nantinya akan menghasilkan nilai kemiripan antara matriks vektor dokumen uji dan matriks masing-masing dokumen latih. Semakin besar nilai yang dihasilkan maka semakin tinggi pula nilai kemiripannya dengan nilai berkisar dari 0 sampai dengan 1. Berikut hasil perhitungan *cosine simlairity* pada tabel 4.16.

**Tabel 4.16 Bobot Nilai Jarak Dokumen**

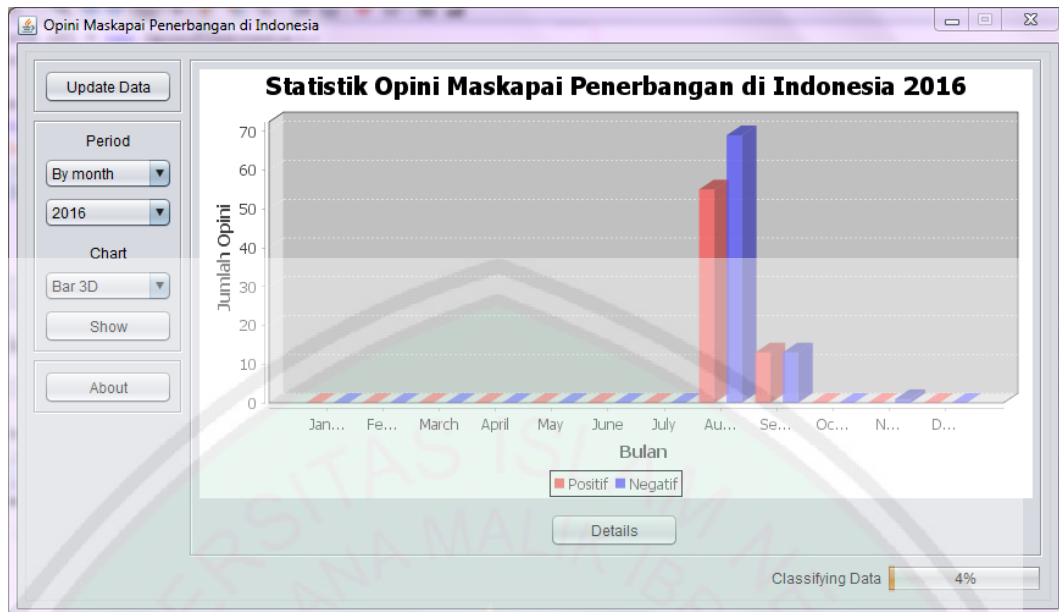
<b>R(Q,d1)</b>	<b>R(Q,d2)</b>	<b>R(Q,d3)</b>	<b>R(Q,d4)</b>	<b>R(Q,d5)</b>
0.137	0.07	0	0.345	0.118

Berdasarkan nilai *cosine similarity* tersebut akan didapatkan hasil perankingan dokumen sesuai dengan tingkat kemiripan dokumen latih terhadap dokumen uji diurutkan berdasarkan dokumen yang memiliki nilai paling tinggi.

Penentuan hasil opini pada dokumen uji juga berdasarkan *range* yang digunakan pada algoritma ini, jika *range* yang digunakan adalah 3 maka diambil 3 peringkat teratas dokumen latih yang telah dilakukan perankingan. Opini yang diambil adalah opini yang memiliki frekuensi paling besar dalam *range* yang digunakan.

#### 4.1.7. Desain dan Implementasi GUI

Desain GUI diharapkan dapat memudahkan *user* dalam menggunakan sistem aplikasi ini. Nantinya akan dijelaskan kegunaan dari tiap komponen yang ada pada aplikasi ini. Rancangan tampilan aplikasi ditunjukkan pada gambar 4.18.



Gambar 4.18 Tampilan GUI 1

Penjelasan dari tampilan GUI 1 di atas antara lain:

a) *Button Update Data*

Berfungsi untuk mengambil data dokumen uji dan dilakukan klasifikasi opini pada dokumen tersebut.

b) *TextArea Periode*

Berfungsi sebagai acuan waktu dalam penampilan chart statistik. Terdapat 2 Combo Box pada fungsi ini, Combo Box 1 berisi pilihan 'by Month', 'by Quarterly' dan 'by Year'. Combo Box 2 berisi pilihan tahun yang ada pada *database server*.

c) *TextArea Chart*

Berfungsi sebagai acuan pada tampilan chart statistik. Terdapat 1 Combo Box yang berisi pilihan Chart yakni 'Bar 3D', 'Line 3D' dan 'Stacked Area'.

d) *Button Show*

Berfungsi untuk menampilkan chart statistik sesuai seleksi periode dan chart yang dipilih.

e) *Button About*

Berfungsi menampilkan informasi terkait tentang aplikasi Statistik Opini Maskapai Penerbangan di Indonesia.

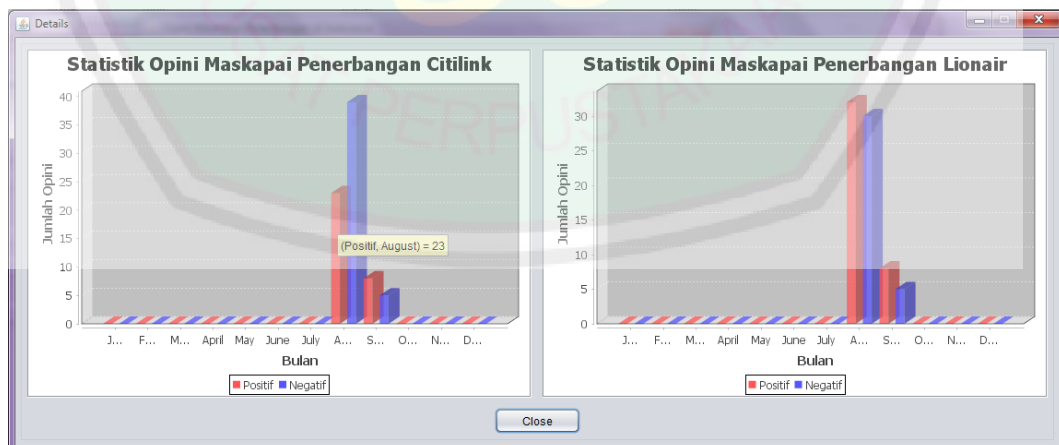
f) *Chart Statistik*

Berfungsi menampilkan chart untuk mempermudah user dalam membaca statistik.

g) *Button Details*

Berfungsi menampilkan data statistik opini yang telah di saring sesuai dengan maskapai yang digunakan dalam penelitian.

Tampilan GUI lainnya yakni tampilan lanjutan dari button details yang menampilkan grafik chart dengan batas data tiap maskapai. Tampilan tersebut dapat dilihat pada gambar 4.19.



Gambar 4.19 Tampilan GUI 2

Penjelasan dari tampilan GUI 2 di atas antara lain:

a) *Button Close*

Berfungsi untuk keluar dari tampilan GUI ini dan kembali ke tampilan GUI awal.

b) *Chart Statistik*

Berfungsi menampilkan chart untuk mempermudah *user* dalam membaca statistik dengan parameter masing-masing maskapai.

#### 4.2. Pengujian Sistem

Pengujian pada tahap ini dilakukan dengan melakukan tes akurasi terhadap sistem. Tes akurasi dilakukan dengan cara menggunakan seluruh data dokumen latih sebagai data uji dengan total data berjumlah 4.342.

Evaluasi pengujian akurasi sistem dilakukan dengan menggunakan *range* 3, 5 dan 9 pada algoritma *cosine similarity* yang digunakan. Setiap dokumen latih nantinya akan dilakukan perhitungan dokumen relevansi terhadap setiap dokumen latih yang ada sesuai dengan maskapai data tersebut.

**Tabel 4.17 Hasil Pengujian Sistem Range 3**

Maskapai		Positif	Negatif	Akurasi	Error
Citilink	Positif	1086	98	90.858%	9.142%
	Negatif	132	1187		
Lion Air	Positif	351	79	89.163%	10.837%
	Negatif	35	1025		
				90.01%	9.99%



Skenario pengujian pertama yakni penggunaan range 3 pada algoritma *cosine similarity* yang digunakan. Untuk mendapatkan nilai akurasi digunakan perhitungan menggunakan rumus 2.3 pada pembahasan sebelumnya. Hasil pengujian sistem dengan *range* 3 pada tabel 4.17 membuktikan bahwa masih terdapat error pada masing-masing maskapai. Sedangkan jika di konversi pada perbandingan setiap opini, maka nilai akurasi untuk opini positif sebesar 86.674% dan opini negatif sebesar 93.345%.

Selanjutnya pada skenario pengujian kedua, menggunakan *range* 5 pada implementasinya. Hasilnya dapat dilihat pada tabel 4.18, dimana nilai akurasi pada kedua maskapai lebih rendah atau menurun jika dibandingkan dengan pengujian yang pertama dengan menggunakan *range* 3. Untuk perbandingan setiap opini, nilai akurasi untuk opini positif sebesar 81.941% dan opini negatif sebesar 91.129%.

**Tabel 4.18 Hasil Pengujian Sistem Range 5**

Maskapai		Positif	Negatif	Akurasi	Error
Citilink	Positif	1051	133	87.636%	12.364%
	Negatif	178	1141		
Lion Air	Positif	323	107	85.436%	14.564%
	Negatif	45	1015		
				86.536%	13.464%

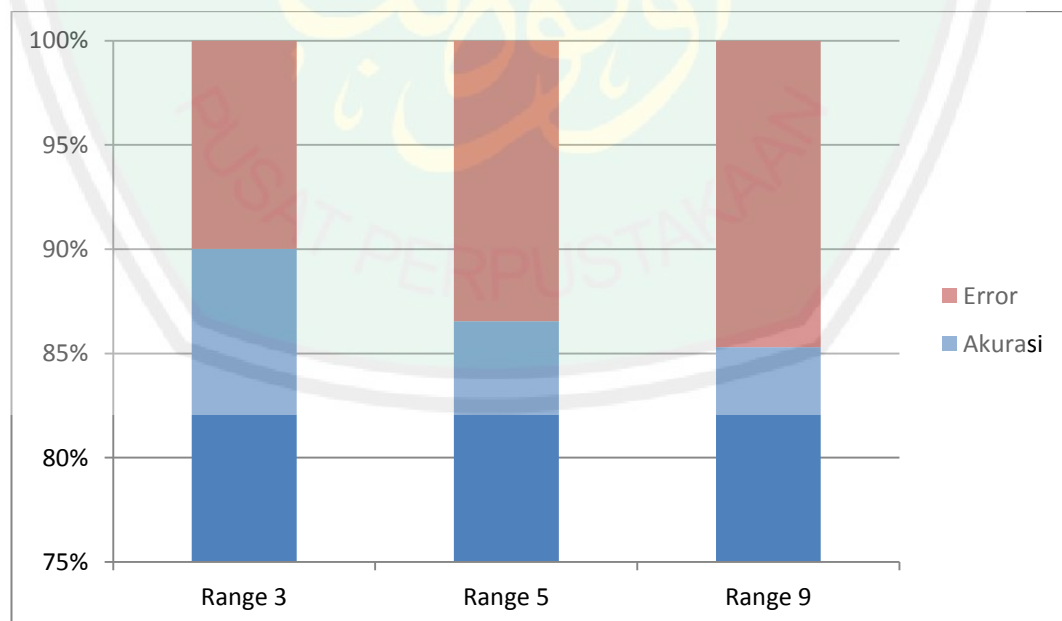
Pengujian tahap ketiga, digunakan range 9 sebagai acuan kedekatan tiap dokumen. Hasil pada tabel 3.19 menunjukkan bahwa tingkat nilai akurasi sistem lebih rendah dari pengujian pertama maupun pengujian kedua. Untuk

perbandingan setiap opini, nilai akurasi untuk opini positif sebesar 80.071% dan opini negatif sebesar 90.522%.

**Tabel 4.19 Hasil Pengujian Sistem Range 9**

Maskapai		Positif	Negatif	Akurasi	Error
Citilink	Positif	1048	136	86.713%	13.287%
	Negatif	199	1120		
Lion Air	Positif	308	122	83.88%	16.12%
	Negatif	41	1019		
				85.297%	14.703%

Hasil pengujian sistem yang diperoleh dapat dikatakan bahwasannya setiap *range* pastinya memiliki nilai error dengan nilai yang berbeda-beda. Terbukti pada setiap *range* memiliki nilai akurasi yang lebih besar daripada nilai error.



**Gambar 4.20 Grafik Perbandingan Range**

Pada gambar 4.20 dapat dilihat bahwa semakin kecil *range* maka tingkat akurasi yang didapatkan menjadi semakin baik. Pada tahap kali ini didapatkan *range* 3 yang menjadi tingkat akurasi paling tinggi pada setiap maskapai yang nantinya akan diimplementasikan pada aplikasi penentuan opini maskapai.

Penggunaan sistem ini juga memperhitungkan waktu yang dibutuhkan oleh sistem dengan beberapa tahapan yang ada untuk melakukan pembaruan data uji sistem. Berikut spesifikasi waktu yang dibutuhkan oleh sistem:

- a) Cek koneksi : 2 detik
- b) Cek python : 2 detik
- c) *Crawling* Data Uji : ± 12 menit
- d) Membaca Data : ± 8 menit
- e) Klasifikasi : Waktu yang dibutuhkan menyesuaikan total data, persatu data uji memerlukan waktu ± 2 menit.

Untuk waktu yang dibutuhkan oleh proses *crawling* data uji, dalam pengujian ini dicoba dalam kecepatan internet 7.2 Mbps dan kemungkinan proses waktu yang dibutuhkan dalam proses ini dapat berbeda jika menggunakan koneksi yang berbeda pula.

#### 4.3. Integrasi Islam

Sistem ini dibuat sebagai bahan evaluasi pelayanan yang ada pada maskapai penerbangan di Indonesia. Pelayanan yang baik akan menjadikan orang yang menggunakan pelayanan tersebut menjadi bahagia dan tidak takut untuk menggunakan pelayanan tersebut. Selain itu dalam hidup ini bukan hanya untuk diri sendiri melainkan hakikat hidup adalah menjadi abdi yang dapat berguna

sebanyak-banyaknya bagi orang lain sebagai manifestasi khaira ummah. Syariat Islam menilai bahwa perbuatan atau pelayanan terbaik seseorang kepada orang lain pada hakikatnya ia telah berbuat baik untuk dirinya sendiri, sebagaimana dijelaskan pada penggalan QS. Al-Isra': 7 yang berbunyi:

إِنَّ أَحْسَنَكُمْ أَحْسَنُكُمْ لَأَنْفُسِكُمْ ... ﴿٧﴾

*Artinya: "jika kamu berbuat baik, (berarti) kamu berbuat baik bagi dirimu sendiri"*

Seseorang yang disertai tanggung jawab oleh negara kemudian tidak dilaksanakan dengan baik sesuai standar pelayanan yang telah ditentukan, Allah akan murka kepadanya sehingga kelak tidak mendapatkan perhatian Allah di hari kiamat. Nabi Muhammad bersabda "Barangsiapa disertai urusan manusia lalu menghindar melayani kamu yang lemah dan mereka yang memerlukan bantuan, maka kelak di hari kiamat Allah tidak akan mengindahkannya." (HR. Imam Ahmad).

Hal tersebut menjelaskan bahwasannya memperbaiki pelayanan publik yang salah satunya yaitu maskapai penerbangan menjadi kebaikan antar sesama manusia demi menciptakan kondisi yang nyaman dalam penggunaan pelayanan yang ada. Selain itu terdapat juga dalam QS. Al-Hujurat: 21 yang berbunyi:

يَأْتِيهَا الَّذِينَ ءَامَنُوا أَجْتَنِبُوا كَثِيرًا مِّنَ الظَّنِّ إِنَّ بَعْضَ الظَّنِّ إِثْمٌ وَلَا تَجَسَّسُوا  
وَلَا يَغْتَب بَّعْضُكُم بَعْضًا ۚ أَنُحِبُّ أَحَدُكُمْ أَن يَأْكُلَ لَحْمَ أَخِيهِ مَيْتًا

فَكَرِهْتُمُوهُ ۚ وَاتَّقُوا اللَّهَ ۚ إِنَّ اللَّهَ تَوَّابٌ رَّحِيمٌ ﴿٢١﴾

*Artinya: “Hai orang-orang yang beriman, jauhilah kebanyakan purba-sangka (kecurigaan), karena sebagian dari purba-sangka itu dosa. dan janganlah mencari-cari keburukan orang dan janganlah menggunjingkan satu sama lain. Adakah seorang diantara kamu yang suka memakan daging saudaranya yang sudah mati? Maka tentulah kamu merasa jijik kepadanya. dan bertakwalah kepada Allah. Sesungguhnya Allah Maha Penerima taubat lagi Maha Penyayang.”*

Dalam surah Al-Hujurat ayat 12 Allah SWT melarang berpikiran negatif atau *negative thinking* artinya Allah menyukai orang-orang yang memiliki cara pandang/pensikapan yang baik terhadap permasalahan yang dihadapinya. Menurut pandangan Islam, ada tingkatan penyikapan berpikir positif terhadap suatu keadaan yakni yang pertama adalah qona’ah, yaitu menerima apa yang dianugerahkan Allah sebagai suatu kewajiban, baik itu positif maupun negatif.

Kedua, istiqomah yang secara harfiah berarti “tegak berdiri” atau “tidak bergeser” atau dengan kata lain konsisten. Para ulama mengkaitkannya dengan tetap berpegang teguh kepada aturan agama. Ketiga, tawakal yaitu memasrahkan hasil suatu ikhtiar atau usaha kepada Allah. Dari hal tersebut dapat dijelaskan bahwsannya perbuatan yang harus dihindari oleh orang-orang yang beriman dan salah satunya adalah menggujing seperti hujatan, cercaan dan makian. Akan tetapi opini yang ditemukan masih banyak makian yang ditujukan kepada maskapai terkait, hal tersebut menandakan bahwa masih ada orang yang berprasangka negatif pada suatu kejadian yang artinya masih belum memilik rasa husnudzan kepada Allah atas segala kejadian yang terjadi.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Berdasarkan aplikasi yang telah dibuat dan hasil yang didapat dari serangkaian uji coba, maka dapat ditarik kesimpulan bahwa penggunaan metode WIDF dan metode k-NN lebih baik menggunakan *range* 3 dibandingkan dengan *range* 5 dan *range* 9. Nilai tingkat akurasi pada *range* 3 untuk sentimen positif sebesar 86.674% dan sentimen negatif sebesar 93.345%.

#### 5.2. Saran

Beberapa saran yang diusulkan untuk pengembangan penelitian selanjutnya setelah dilakukan penelitian ini adalah sebagai berikut:

- 1) Adanya peningkatan koleksi kamus sehingga diharapkan dapat meningkatkan nilai akurasi pada sistem.
- 2) Dapat dikembangkan dengan penambahan teknik *POST tagging* atau *decision rule*.

## DAFTAR PUSTAKA

- Aliandu, P. (2013). *Twitter Used by Indonesian President: An Sentiment Analysis of Timeline*. Department of Informatics, Faculty of Engineering, Widya Mandira Catholic University. Information Systems International Conference (ISICO).
- Asian, J. (2007). *Effective Techniques for Indonesian Text Retrieval*. School of Computer Science and Information Technology, RMIT University.
- Barawi, M. H. & Seng, Y. Y. (2013). *Evaluation of Resources Creations by Using Sentiment Analysis*. The 9<sup>th</sup> International Conference on Cognitive Science. Elsevier Publisher Inc.
- Davidov, D., Tsur, O. & Rappoport, A. (2010). *Enhanced Sentiment Learning Using Twitter Hastags and Smileys*. Institute of Computer Science, The Hebrew University.
- Feldman, R. & Sanger, J. (2007). *The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York.
- Han, J. & Kamber, M. (2000). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Hotho, A., Nurnberger, A. & Paaß, G. (2005). *A Brief Survey of Text Mining*. Computational Linguistics and Language Technology.
- Indraswari, A. A. (2015). *Perbandingan Jejaring Sosial dan Analisis Sentimen Pada Bank di Indonesia Untuk Kepentingan Social Customer Relationship Management*. Manajemen Bisnis Telekomunikasi dan Informatika, Fakultas Ekonomi dan Bisnis Universitas Telkom, Bandung.
- Khamar, K. (2013). *Short Text Classification Using kNN Based on Distance Function*. International Journal of Advanced Research in Computer and Communication Engineering (pp. 1916-1919). IJARCCCE.
- Krisandi, N., Helmi & Prihandono, B. (2013). *Algoritma K-Nearest Neighbor Dalam Klasifikasi Data Hasil Produksi Kelapa Sawit Pada Pt. Minamas Kecamatan Parindu*. Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster).
- Kumar, A. & Sebastian, T. M. (2012). *Sentiment Analysis on Twitter*. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3.
- Kurniawan, H. (2012). *Sistem Penentuan Kualitas Air pada Depot Air Minum Menggunakan Metode K-Nearest Neighbor*. Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru.

- Letierce, J., Passant, A. & Decker, S. (2010). *Understanding How Twitter is Used To Spread Scientific Messages*. Digital Enterprise Research Institute National University of Ireland, Galway.
- Liu, B. (2007). *Web Data Mining*. ACM Computing Classification, Springer Berlin Heidelberg. ISBN-10 3-540-37881-2.
- Liu, B. (2012). *Sentiment Analysis and Subjectivity*. Synthesis Lectures on Human Language Technologies, USA. Editor: Graeme Hirst Morgan & Claypool Publishers.
- Novantirani, A. (2015). *Analisis Sentimen pada Twitter Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine*. Teknik Informatika, Fakultas Informatika, Universitas Telkom, Bandung.
- Nugraha, M. W.A. (2014). *Sentiment Analysis Pada Review Film Dengan Menggunakan Metode K-Nearest Neighbor*. Program Studi Teknik Informatika, Fakultas Teknik Universitas Widyatama, Bandung.
- Purnomo, J., Firdaus, Y. & Hidayati, H. (2010). *Analisis Perbandingan Beberapa Metode Pembobotan Kata Terhadap Performansi Ketgorisasi Teks*. Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom.
- Samuel, Y., Delima, R. & Rachmat, A. (2014). *Implementasi Metode K-Nearest Neighbor dengan Decision Rule untuk Klasifikasi Subtopik Berita*. Program Studi Teknik Informatika, Universitas Kristen Duta Wacana, Yogyakarta. Jurnal Informatika, Vol. 10 No. 1, Juni 2014: 1-15.
- Suh, B., Hong, L., Pirolli, P. & Chi, E. H. (2010). *Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network*. Palo Alto Research Center Inc.
- Sunni, I. & Widyantoro, D. H. (2012). *Analisis Sentimen dan Ekstraksi Topik Penentu Sentimen pada Opini Terhadap Tokoh Publik*. Jurnal Sarjana Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika.
- Sussolaikah, Kelik. & Alwa, Aslan. (2016). *Sentiment Analysis Terhadap Acara Televisi Mata Najwa Berdasarkan Opini Masyarakat Pada Microblogging Twitter*. Konferensi Nasional Teknologi Informasi dan Komunikasi 2016.
- Visa, S., Ramsay, B., Ralescu, A. & Knaap, E. V. D. (2011). *Confusion Matrix-based Feature Selection*.
- Witten, I.H. (2005). *Text Mining*. Practical handbook of internet computing, edited by M.P. Singh, pp. 14-1 - 14-22. Chapman & Hall/CRC Press, Boca Raton, Florida.



Zaki, M. J. & Meira JR, W. (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, New York. ISBN 978-0-521-76633-3.

Zhao, Y. (2013). *R and Data Mining: Examples and Case Studies*. Elsevier Publishers Inc.

