

TESIS

**PENGUKURAN KINERJA OPTIMASI ALGORITMA BAT PADA
ALGORITMA NAIVE BAYES, DECISION TREE DAN K-NN UNTUK
SENTIMEN ANALISIS DI LINI MASA TWITTER**



Disusun oleh:

Nama : Candra Adipradana
NIM : 18.52.1125
Konsentrasi : Informatics Technopreneurship

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2021

TESIS

**PENGUKURAN KINERJA OPTIMASI ALGORITMA BAT PADA
ALGORITMA NAIVE BAYES, DECISION TREE DAN K-NN UNTUK
SENTIMEN ANALISIS DI LINI MASA TWITTER**

**THE OPTIMATION MEASUREMENT OF BAT ALGORITHM FOR NAIVE
BAYES, DECISION TREE AND K-NN ALGORITHM FOR THE SENTIMENT
ANALYSIS OF THE TWITTER USERS**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Candra Adlpradana
NIM : 18.52.1125
Konsentrasi : Informatics Technopreneurship

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2021

HALAMAN PENGESAHAN

**PENGUKURAN KINERJA OPTIMASI ALGORITMA BAT PADA
ALGORITMA NAIVE BAYES, DECISION TREE DAN K-NN UNTUK
SENTIMEN ANALISIS DI LINI MASA TWITTER**

**THE OPTIMATION MEASUREMENT OF BAT ALGORITHM FOR NAIVE
BAYES, DECISION TREE AND K-NN ALGORITHM FOR THE SENTIMENT
ANALYSIS OF THE TWITTER USERS**

Dipersiapkan dan Disusun oleh

Candra Adipradana

18.52.1125

Telah Dujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 5 Januari 2021

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 5 Januari 2021

Rektor

Prof. Dr. M. Suvanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

**PENGUKURAN KINERJA OPTIMASI ALGORITMA BAT PADA
ALGORITMA NAIVE BAYES, DECISION TREE DAN K-NN UNTUK
SENTIMEN ANALISIS DI LINI MASA TWITTER**

**THE OPTIMATION MEASUREMENT OF BAT ALGORITHM FOR NAIVE
BAYES, DECISION TREE AND K-NN ALGORITHM FOR THE SENTIMENT
ANALYSIS OF THE TWITTER USERS**

Dipersiapkan dan Disusun oleh

Candra Adlpradana

18.52.1125

Telah Diuji dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 5 Januari 2021

Pembimbing Utama

Anggota Tim Penguji

Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

Prof. Dr. Bambang Soedjono, WA.
NIK. 555126

Pembimbing Pendamping

Dr. Arief Servanto, S.Si, M.T
NIK. 190302036

Anggit Dwi Hartanto, M.Kom
NIK. 190302163

Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 5 Januari 2021

Direktur Program Pascasarjana

Dr. Kusri, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Candra Adipradana
NIM : 18.52.1125
Konsentrasi : Informatics Technopreneurship

Menyatakan bahwa Tesis dengan judul berikut:
**Pengukuran Kinerja Optimasi Algoritma Bat Pada Algoritma Naive Bayes,
Decision Tree Dan K-NN Untuk Sentimen Analisis Di Lini Masa Twitter**

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Anggit Dwi Hartanto, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 5 Januari 2021

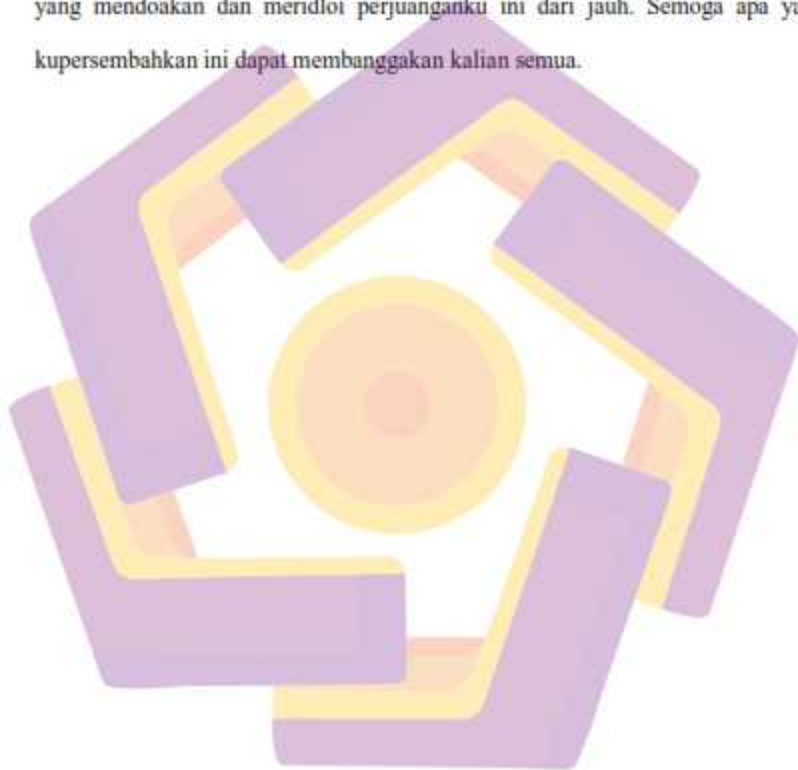
Yang Menyatakan,



Candra Adipradana

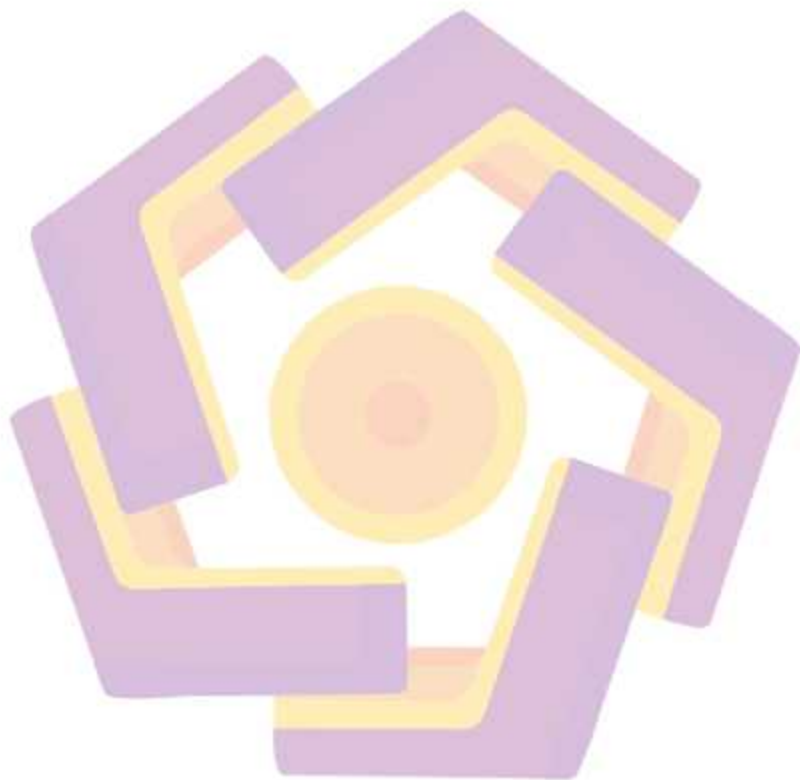
HALAMAN PERSEMBAHAN

Ibuku yang telah berpulang ke Rahmatullah pada 22 Desember 2019, Semoga ridlo dan doamu menghantarkanku meraih gelar Magister ini serta ayah yang mendoakan dan meridloi perjuanganku ini dari jauh. Semoga apa yang kupersembahkan ini dapat membanggakan kalian semua.



HALAMAN MOTTO

"Ubah Sudut Pandangmu Agar Terasa Ikhlasnya"



KATA PENGANTAR

Puji syukur Alhamdulillah kehadiran Allah Swt, atas segala rahmat-NYA sehingga penulis dapat menyelesaikan Tesis yang berjudul “PENGUKURAN KINERJA OPTIMASI ALGORITMA BAT PADA ALGORITMA NAIVE BAYES, DECISION TREE DAN K-NN UNTUK SENTIMEN ANALISIS DI LINI MASA TWITTER”.

Laporan Tesis ini disusun untuk memenuhi salah satu syarat akademis dalam menyelesaikan Program Studi S2 Teknik Informatika Program Pascasarjana Universitas AMIKOM Yogyakarta.

Pada kesempatan ini, penulis menyampaikan terima kasih kepada :

1. Allah Swt, yang telah memberikan semua kebutuhan sehingga terselesaikannya laporan Tesis ini.
2. Ayah, (alm.) Umi dan semua saudara atas semua dukungan, doa dan kesabarannya sehingga penulis dapat menyelesaikan Tesis ini.
3. Bunda dan anak-anak yang senantiasa mendoakan dan terus memberikan dukungan
4. Bapak Imam Suhaimi, S.Pd, M.Pd, Calon Doktor, Rektor Universitas Kahuripan Kediri yang memberikan dorongan dan inspirasi agar penulis mantab mengambil kuliah S2 ini serta dorongan agar segera menyelesaikan laporan Tesis ini.

5. Prof. Dr. Ema Utami, S.Si., M.Kom, Professor inspirator saya, yang memberikan wawasan dan dukungan atas terselesaikannya Tesis ini. Semoga sehat selalu Prof.
6. Bapak Anggit Dwi Hartanto, M.Kom, yang telah memberikan ide judul Tesis serta dukungan atas terselesaikannya tugas ini.
7. Semua civitas akademik Universitas Kahuripan Kediri yang cukup memberikan kami sebagai penulis ruang dalam menyelesaikan tugas Tesis ini.

Menyadari bahwa dalam laporan Tesis ini masih terdapat banyak sekali kekurangan, penulis sangat mengharapkan saran dan kritik yang membangun dari semua pihak yang akan sangat berguna untuk perbaikan dan penyempurnaan laporan Tesis ini.

Yogyakarta, 13 Januari 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR isi.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
INTISARI.....	xvii
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	6
1.6. Hipotesis.....	7
BAB II TINJAUAN PUSTAKA.....	8
2.1. Tinjauan Pustaka.....	8

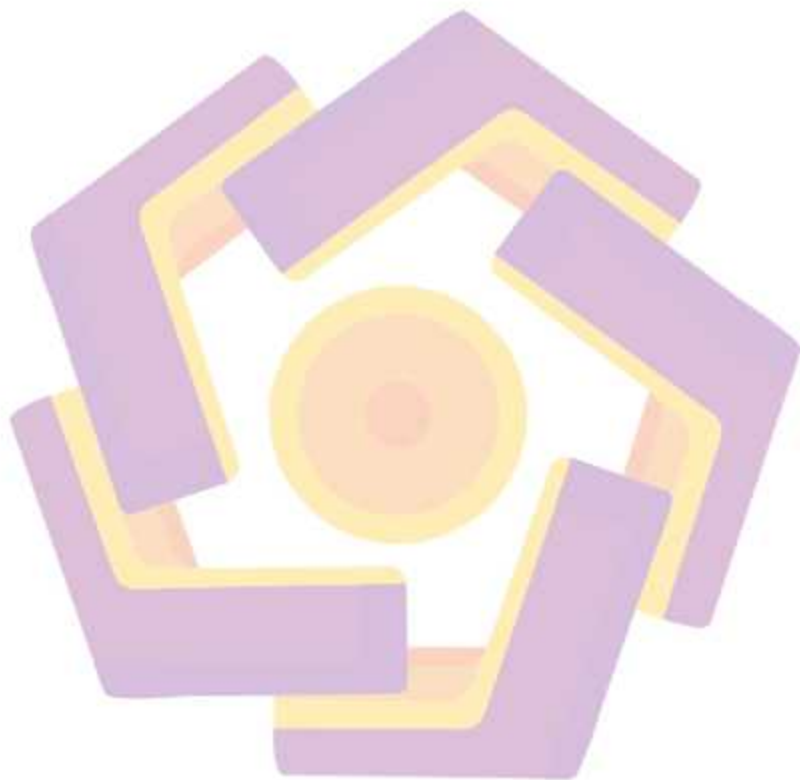
2.2. Keaslian Penelitian.....	12
2.3. Landasan Teori.....	17
BAB III METODE PENELITIAN.....	32
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	32
3.2. Metode Pengumpulan Data.....	33
3.3. Metode Analisis Data.....	37
3.4. Alur Penelitian.....	37
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	55
4.1. Crawling Data Tweet (Twitter search API).....	55
4.2 Analisis Sumber Data.....	55
4.3. PreProcessing Data.....	57
4.4 Klasifikasi dan Akurasi.....	62
BAB V PENUTUP.....	96
5.1. Kesimpulan.....	96
5.2. Saran.....	96
DAFTAR PUSTAKA.....	97

DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian Pengukuran Kinerja Algoritma Naive Bayes, Decision Tree dan K-Nearest Neighbor (K-NN) Untuk Sentimen Analisis.....	12
Tabel 3.1. Contoh Pelabelan Data Tweet.....	41
Tabel 3.2. Simulasi Hasil Pembobotan Kata.....	47
Tabel 3.3 Contoh Hasil Cross Validation Prediksi Dua Kelas.....	49
Tabel 3.4. Confusion Matrix (Gorunescu 2011).....	51
Tabel 4.1. Hasil Crawling data Twitter dari RapidMiner.....	55
Tabel 4.2. Hasil proses Crawling, cleaning dan labeling.....	57
Tabel 4.3. Contoh Hasil Tokenizing, StopWords Removal dan Stemming.....	59
Tabel 4.4. Contoh Dokumen awal.....	60
Tabel 4.5. Contoh Data Tokenizing.....	61
Tabel 4.6. Simulasi perhitungan Pembobotan Kata.....	62
Tabel 4.7. Contoh dokumen yang akan diklasifikasi.....	63
Tabel 4.8. Contoh tabel Training Data.....	63
Tabel 4.9. Naïve Bayes Model.....	64
Tabel 4.10. Training Data Model.....	65
Tabel 4.11. Prediction Model.....	65
Tabel 4.12. Confusion Matrix Tabel.....	66
Tabel 4.13. Konversi Data String ke Numerik.....	68
Tabel 4.14. Contoh Data Latih/Training Data.....	69

Tabel 4.15. Contoh Data Uji/Testing Data.....	70
Tabel 4.16. Contoh K-NN Model	70
Tabel 4.18. K-NN Model Prediction.....	72
Tabel 4.19. Confusion Matrix dan Akurasi.....	72
Tabel 4.20. Data Latih Decision Tree	74
Tabel 4.21. Data Uji/Testing Data Decision Tree.....	75
Tabel 4.22. Data Latih/Training Data Kata Jabat	76
Tabel 4.23. Data Uji/Testing Data Kata Jabat.....	77
Tabel 4.24. Tabel Hasil Statistik pada Decission Tree	79
Tabel 4.25 Identifikasi kelelawar pada tabel klasifikasi kata	82
Tabel 4.26 Hasil perhitungan frekuensi	83
Tabel 4.27 Hasil perhitungan kecepatan kelelawar.....	84
Tabel 4.28 Hasil loudness/kebisingan kelelawar	85
Tabel 4.29 Hasil <i>pulse emission rate</i> kelelawar.....	85
Tabel 4.30 Hasil update posisi kelelawar.....	86
Tabel 4.31 Hasil perhitungan fungsi obyektif / <i>fitness</i> posisi kelelawar.....	87
Tabel 4.32 Rekap fungsi obyektif/ <i>fitness</i> posisi kelelawar pada seluruh sentimen	88
Tabel 4.33 Akurasi Naïve Bayes dengan 100 Bat.....	89
Tabel 4.34 Akurasi Decision Tree dengan 100 Bat	91
.....	91
Tabel 4.35 Akurasi KNN dari berbagai nilai K	92
Tabel 4.36 Perbandingan hasil optimasi	93

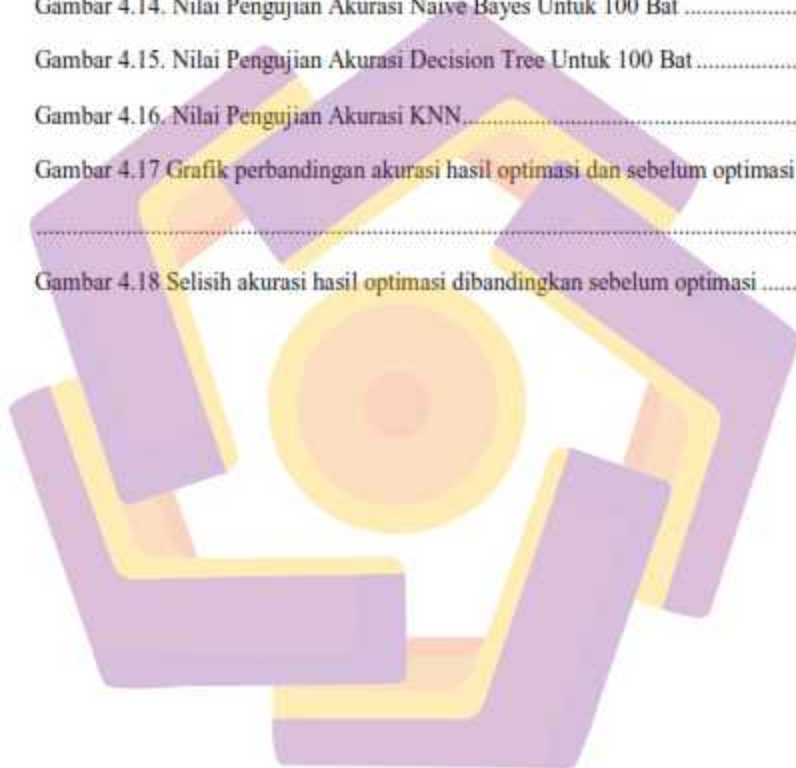
Tabel 4.37. Perbandingan selisih hasil optimasi dibandingkan sebelum optimasi 94



DAFTAR GAMBAR

Gambar 3.1 Skema Web Crawling.....	34
Gambar 3.2. Alur Penelitian.....	38
Gambar 3.3. Tahap PreProcessing Data.....	40
Gambar 3.4. Contoh penerapan <i>Cleansing</i>	41
Gambar 3.5. Contoh penerapan <i>case folding</i>	42
Gambar 3.6. Contoh penerapan tokenizing.....	43
Gambar 3.7. Contoh penerapan <i>Filtering/Stop Word Removal</i>	44
Gambar 3.8. Contoh penerapan <i>Stemming</i>	44
Gambar 3.9. Representasi 10 folds cross validation.....	50
Gambar 3.10. Confussion matrix model dalam aplikasi rapid miner.....	52
Gambar 3.11. Hasil akurasi dari Confussion Matrix.....	52
Gambar 3.12. Flowchart Algoritma Bat.....	54
Gambar 4.1. Contoh dari isi kamus slangword.....	59
Gambar 4.2 klasifikasi Naive Bayes dan akurasinya.....	67
Gambar 4.3 Hasil akurasi dari Naive Bayes.....	68
Gambar 4.4. Layar proses Training dan Testing dengan klasifikasi K-NN.....	73
Gambar 4.5. Hasil akurasi dari K-NN.....	73
Gambar 4.6 Hasil performance vektor dari K-NN.....	74
Gambar 4.7. Desain Decision Tree.....	78
Gambar 4.8. Layar proses Training dan Testing dengan Decission Tree.....	78
Gambar 4.9 Hasil akurasi dari Decission Tree.....	79

Gambar 4.10. Hasil Tree pada Decission Tree	79
Gambar 4.11. Bentuk model pada Decission Tree.....	80
Gambar 4.12. Word cloud penelitian	80
Gambar 4.13. Grafik class penelitian.....	81
Gambar 4.14. Nilai Pengujian Akurasi Naïve Bayes Untuk 100 Bat	90
Gambar 4.15. Nilai Pengujian Akurasi Decision Tree Untuk 100 Bat.....	91
Gambar 4.16. Nilai Pengujian Akurasi KNN.....	93
Gambar 4.17 Grafik perbandingan akurasi hasil optimasi dan sebelum optimasi	94
.....	94
Gambar 4.18 Selisih akurasi hasil optimasi dibandingkan sebelum optimasi	95



INTISARI

Media sosial merupakan suatu media komunikasi yang sangat efektif pada era digital saat ini. Dari sekian media sosial, Twitter merupakan media sosial yang paling banyak digunakan. Banyak tweet yang masuk pada Twitter mendorong timbulnya penelitian dibidang text mining. Salah satu cabang dari text mining adalah sentimen analisis. Sentiment analisis pada penelitian ini dibentuk dari 3 algoritma klasifikasi yaitu Naïve Bayes, Decission Tree dan K-NN. Pada prakteknya, hasil dari 3 algoritma klasifikasi tersebut sering kali menghasilkan tingkat akurasi yang sangat rendah. Algoritma Bat/kelelawar merupakan algoritma yang mampu mengoptimasi hasil dari akurasi dari algoritma Naïve Bayes, Decission Tree, K-NN.

Dalam penelitian ini, dibuat 2 skenario penelitian yaitu pertama, menghitung akurasi dari algoritma Naïve Bayes, Decission Tree dan K-NN. Kedua, mengoptimasi hasil klasifikasi dari 3 algoritma tersebut dengan metode algoritma Bat yang kemudian diujikan kembali nilai akurasinya. Pada skenario pertama persentase dihasilkan dari nilai akurasi Naïve Bayes sebesar 33,58, Decission Tree senilai 32,82 dan K-NN senilai 33,61. Pada skenario kedua, dengan menggunakan salah satu fungsi obyektif yaitu $f(x) = x^2$ dihasilkan nilai Naïve Bayes 39,01, Decission Tree 76,63 dan K-NN 66,15.

Dari hasil 3 uji optimasi Algoritma Klasifikasi ini didapatkan bahwa keseluruhan fungsi obyektif dari algoritma Bat ini semuanya mampu meningkatkan nilai akurasi data dari sebelum dioptimasi. Dari sekian pengujian didapatkan bahwa algoritma Decision Tree memiliki nilai rata-rata penambahan optimasi tertinggi yaitu 43,81%.

Kata kunci: Naïve Bayes, Decission Tree, K-NN, Algoritma Kelelawar, optimasi

ABSTRACT

Social media is a very effective communication media in today's digital era. Of the social media, Twitter is the most widely used social media. Many tweets entered on Twitter have encouraged research in the field of text mining. One of the branches of text mining is sentiment analysis. Sentiment analysis in this study was formed from 3 classification algorithms, namely Naïve Bayes, Decision Tree and K-NN. In practice, the results of the 3 classification algorithms often produce very low levels of accuracy. Bat algorithm is an algorithm that can optimize the results from the accuracy of the Naïve Bayes algorithm, Decision Tree, K-NN. In this study, two research scenarios were made: first, calculating the accuracy of the Naïve Bayes algorithm, Decision Tree and K-NN. Second, optimizing the classification results of the 3 algorithms with the Bat algorithm method, which then re-tested the accuracy value.

In the first scenario the percentage is generated from the accuracy value of Naïve Bayes of 33,58, Decision Tree of 32,82 and K-NN of 33,61. In the second scenario, using one of the objective functions, namely $f(x) = x^2$, the Naïve Bayes value is obtained 39,01, Decision Tree 76,63 and K-NN 66,15.

From the results of 3 the optimization test of classification Algorithm, it was found that the overall objective functions of the Bat algorithm were all able to increase the data accuracy value from before optimization. From all the tests, it was found that the Decision Tree algorithm has the highest average value of optimization increment, namely 43,81 %.

Keywords: Naïve Bayes, Decision Tree, K-NN, Bat Algorithm, optimization

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pengguna internet di Indonesia tercatat pada tiap tahun terus mengalami peningkatan dibandingkan pada tahun sebelumnya. Sesuai dengan hasil polling Indonesia yang bekerja sama dengan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) bahwa jumlah pengguna internet di Indonesia tumbuh 10,12 persen pada tahun tersebut. Dari total populasi sebanyak 264 juta jiwa penduduk Indonesia, sebanyak 171,17 juta jiwa atau sekitar 64,8 persen sudah terhubung ke internet (Pratomo, 2019). Diantara penggunaan internet, media sosial menjadi salah satu media favorit yang sering digunakan para penggunanya (Pertiwi 2018).

Salah satu manfaat dari media sosial adalah sebagai media komunikasi massa sehingga mampu memberikan popularitas kepada siapa saja yang muncul di media massa (Morissan 2013). Dari sekian media sosial yang ada, Twitter merupakan media sosial yang paling banyak digunakan (Iskandar 2015). Menurut data internal Twitter pada kuartal pertama 2018, pengguna Twitter Indonesia tumbuh sebanyak 11 persen, sementara angka global hanya 10 persen.

Pada kuartal kedua, jumlah pengguna Twitter Indonesia tumbuh 31 persen, sedangkan pertumbuhan global hanya 11 persen. Pada kuartal ketiga, Twitter Indonesia mencatat pertumbuhan pengguna aktif harian sebanyak 33 persen, naik tajam dibandingkan dengan pertumbuhan global yang hanya 9 persen, puncaknya

pada kuartal keempat 2018, rerata pertumbuhan pengguna Twitter Indonesia sebesar 41 persen, sedangkan pertumbuhan global tetap 9 persen (Sianipar 2019).

Adanya suatu media sosial seperti Twitter mendorong orang melakukan penelitian untuk mengetahui apa yang sedang dipikirkan oleh orang kebanyakan. Oleh sebab itu dikembangkan suatu teknik klasifikasi untuk mengelompokkan suatu opini yang sedang berkembang, baik yang ada di media sosial maupun media massa yang lain. Sebuah analisa sentimen atas opini yang berkembang di media twitter dapat dikelompokkan menjadi kategori opini positif, negatif ataupun netral (Kriyantono 2012).

Manfaat dari pengklasifikasian ini adalah mampu meneliti tren dan memperkirakan produk yang dihasilkan. Hal ini juga bisa digunakan untuk mempelajari pandangan individu, perilaku, perasaan terhadap orang lain, diri sendiri, masalah yang berkembang serta kegiatan yang selama ini sudah dilakukan. Informasi ini menjadi hal yang menarik bagi para pengambil keputusan sebab mampu memastikan apa yang sedang dipikirkan oleh orang lain. (Basari, et al. 2012).

Teknik klasifikasi ini kemudian berkembang menjadi suatu algoritma klasifikasi. Dari sekian algoritma yang sering digunakan tersebut adalah Naive Bayes Classifier, Decision Tree dan K-Nearest Neighbor (K-NN). (Rozlini, Munirah dan Noorhaniza 2016). Naive Bayes Classifier sendiri merupakan algoritma yang memiliki pendekatan yang cukup sederhana dan efektif. Naive Bayes sendiri lebih tepat apabila diterapkan pada data yang besar serta memiliki atribut yang tidak relevan (Husin, et al. 2017). Pada penelitian ke media Twitter

sebelumnya, Naive Bayes Classifier menghasilkan Nilai Precision 74,5 persen, Recall 73,5 persen, Accuracy 73,9 persen (Kumar, et al. 2017).

Decision Tree merupakan algoritma klasifikasi yang mempunyai input berupa *training sample* dan *sample*. Algoritma ini dapat digunakan untuk memudahkan dalam pengambilan keputusan dengan memproyeksikan data-data yang ada kedalam bentuk pohon keputusan, berdasarkan nilai *entropy* dan *gain* yang dimiliki masing-masing atribut data (Sumarmo 2017). Pada penelitian (Kumar, et al. 2017) di media sosial Twitter, algoritma Decision Tree menghasilkan Nilai Precision 52,6 persen, Recall 53 persen, Accuracy 52,9 persen.

K-NN merupakan algoritma yang mengklasifikasikan suatu obyek berdasarkan data pembelajaran yang paling dekat dengan obyek (Rasenda, Lubis dan Ridwan 2020). Pada penelitian sebelumnya di media Twitter, K-NN mampu menghasilkan Nilai Precision 62,8 persen, Recall 62,9 persen, Accuracy 62,9 persen (Kumar, et al. 2017).

Dalam penerapannya, algoritma klasifikasi ini seringkali kurang menghasilkan solusi yang baik bagi masalah yang dihadapi, baik dari pengumpulan data, Preprocessing, Fitur Ekstraksi, Optimasi Fitur, Klasifikasi dan Optimasi. Untuk itu adanya suatu optimasi atas algoritma klasifikasi diharapkan mampu menghasilkan tingkat solusi yang lebih baik. (Bansal dan Kaur 2018). Ada beberapa algoritma optimasi yang memungkinkan dapat digunakan dalam mengoptimasi algoritma klasifikasi salah satunya yaitu Algoritma Bat atau Bat Algorithm (BA). (Rozlini, Munirah dan Noorhaniza 2016).

Algoritma Bat ini merupakan suatu algoritma metaheuristik baru yang terinspirasi dari kebiasaan echolocation seekor Kelelawar (Rizk-Allah dan Hassanien 2017). Secara umum Algoritma Bat meniru tingkah laku kelelawar dalam mencari makanan dan dapat membedakan jenis-jenis serangga walaupun dalam kegelapan total. Algoritma Bat merupakan metode optimasi metaheuristik yang dapat digunakan sebagai feature selector dalam kasus klasifikasi. Algoritma Bat memberikan performa yang lebih baik bila dibandingkan dengan beberapa algoritma lain seperti Algoritma Genetika, Particle Swarm Optimization dan Geometric Particle Swarm Optimization (Chandra, Widiartha dan Muliantara 2016).

Oleh sebab itu, pada penelitian ini akan dilakukan pengujian optimasi atas hasil klasifikasi analisa sentimen dari algoritma Naive Bayes, Decision Tree dan K-NN dengan menggunakan Algoritma Bat. Hal ini nantinya bertujuan untuk menentukan apakah Algoritma Bat layak menjadi algoritma pengoptimasi untuk algoritma klasifikasi pada sentimen analisis atau tidak.

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah

- a. Berapa tingkat akurasi yang dihasilkan tiap kelompok sentimen dengan menggunakan Algoritma Naive Bayes, Decision Tree dan K-NN ?
- b. Berapa nilai akurasi yang dihasilkan pada tiap kelompok sentimen apabila Algoritma Naive Bayes, Decision Tree dan K-NN dioptimasi dengan Algoritma Bat ?

- c. Apakah Algoritma Bat layak dijadikan sebagai Algoritma optimasi untuk Sentimen Analisis bagi Algoritma Naive Bayes, Decision Tree dan K-NN ?

1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah

- a. Obyek penelitian adalah opini pengguna twitter pada pemilihan kepala daerah.
- b. Algoritma yang digunakan dalam penelitian ini adalah Naive Bayes Classifier, Decision Tree dan K-Nearest Neighbor (K-NN).
- c. Optimasi pada penelitian ini menggunakan algoritma Bat.
- d. Media sosial yang digunakan sebagai penelitian ini adalah Twitter.
- e. Kurun waktu pengambilan data pada Twitter dari bulan April 2020 hingga Juni 2020.
- f. Pengelompokan opini pada Twitter dibedakan menjadi sentimen positif, negatif dan netral.
- g. Penelitian ini tidak mengambil data faktor pendorong netizen dalam mengeluarkan sentimen.
- h. Penelitian ini tidak meneliti buzzer yang melakukan tweet di twitter.
- i. Penelitian ini hanya digunakan untuk menilai tingkat kelayakan kualitas algoritma optimasi (Algoritma Bat) terhadap algoritma klasifikasi (Naive Bayes, Decision Tree dan K-NN).

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

Mengetahui tingkat kelayakan optimasi menggunakan Algoritma Bat terhadap Algoritma Naive Bayes, Decision Tree dan K-Nearest Neighbor (K-NN) pada analisa sentimen pemilihan kepala daerah di media sosial Twitter.

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah :

- a. Dapat menjadi pedoman penelitian dalam menganalisa suatu sentimen publik pada media sosial Twitter.
- b. Dapat menjadi bahan acuan menghitung tingkat akurasi penggunaan suatu metode penelitian pada data mining apabila dilakukan dengan cara optimasi dan tanpa optimasi.
- c. Dapat menjadikan Algoritma Bat sebagai algoritma optimasi alternatif untuk menganalisa sentimen dengan menggunakan algoritma klasifikasi seperti Naive Bayes Classifier, Decision Tree dan K-Nearest Neighbor (K-NN).

1.6. Hipotesis

Dari penelitian ini dibuat suatu hipotesis untuk memperkirakan hasil yang akan didapat yaitu bahwa analisa sentimen dengan menggunakan Model Algoritma Naive Bayes, Decision Tree dan K-Nearest Neighbor (K-NN) diharapkan mampu memiliki nilai prediksi keakurasian diatas 71,94% (Rusdianan 2019) serta jika dioptimasi dengan Algoritma Bat dapat memperoleh nilai diatas 82,79 % (Chaudhary dan Khumar 2016). Diharapkan hal ini dapat menunjukkan tingkat keberhasilan suatu kelayakan algorithma optimasi Kelelawar terhadap hasil dari algoritma klasifikasi.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian-penelitian terdahulu yang relevan, pernah dilakukan, dan dijadikan bahan rujukan dalam penelitian ini adalah sebagai berikut :

Penelitian Rozlini, Munirah dan Noorhaniza, Melakukan pengujian tingkat Melakukan pengujian tingkat efektivitas pengolahan data untuk Algoritma Bat melalui tipe data dan ukuran atribut dalam berbagai dataset menggunakan algoritma Naive Bayes, Decision Tree dan K-NN. Pada penelitian ini didapatkan bahwa suatu tipe data memiliki hubungan dengan kinerja klasifikasi dibandingkan nilai pengurangan atributnya. Meskipun persentase pengurangan atributnya tinggi tetapi hal ini menghasilkan kinerja klasifikasi yang lebih rendah. Namun dalam penelitian ini tidak membahas tentang analisis sentimen serta tools yang digunakan (2016).

Penelitian Khurana dan Sanjib Kumar Sahu, mendalami tentang sentimen analisis pada data twitter menggunakan optimasi Algoritma Bat dengan algoritma klasifikasi yang digunakannya adalah Support Vector Machine. Penelitian ini mendalami tentang bagaimana meningkatkan efisiensi pada algoritma Support Vector Machine serta melakukan pendekatan analisa sentimen dari algoritma tersebut. Penelitian ini menerangkan bagaimana sebuah optimasi Algoritma Bat digunakan namun tidak membahas algoritma klasifikasi yang lain seperti Naive

Bayes Classifier, Decision Tree dan K-Nearest Neighbor (K-NN) serta teknik validasi menggunakan Cross Validation (2018).

Penelitian Bilal, Shahid dan Amin Khan, tentang klasifikasi sentimen berdasarkan algoritma Naive Bayes, Decision Tree dan K-NN. Dalam penelitian ini diterangkan teknik penerapan algoritma klasifikasi Naive Bayes, Decision Tree dan K-NN pada analisa sentimen dimana menghasilkan Naive Bayes memiliki kinerja yang terbaik dalam akurasi, lebih tinggi ketepatannya, lebih tinggi recall dan nilai dari F-Measure jika dibandingkan dengan Decision Tree dan K-NN. Tetapi dari penelitian ini tidak menggunakan RapidMiner Tools dan tidak melakukan optimasi Algoritma Bat serta pengujian validasi (2016).

Penelitian Trisna, dkk., Menganalisa dokumen teks yang mengandung unsur non-tekstual berupa emoji. Algoritma yang digunakan adalah Naive Bayes dengan pembaruan berupa penambahan pembobotan non-tekstual(emoji). Hasil dari pembobotan tekstual dan non tekstual yang dinormalisasi dengan metode min-max menghasilkan sentimen positif-negatif. Data Twitter tentang sebanyak 900 data *tweet*. Dari hasil pengujian akurasi, diperoleh 68,52% untuk kondisi pembobotan tekstual, 75,93% untuk pembobotan non-tesktual, dan 74,81% untuk kondisi penggabungan dengan nilai konstanta 0,5 untuk tekstual dan 0,5 untuk non-tekstual. Dari hasil pengujian pengaruh pembobotan non-tesktual disimpulkan bahwa pembobotan non-tekstual berpengaruh terhadap akurasi dan pengklasifikasian, dengan komposisi konstanta pengali terbaik ketika $a=0,4$ dan $\beta=0,6$ sampai dengan $a=0,1$ dan $\beta=0,9$ (2017).

Penelitian Matharasi, P. Bavithra dan Dr. A.Senthilrajan, Membangun analisa Sentimen dengan menggunakan pendekatan Unigram dari Naive Bayes. Pengembangan Naive Bayes Classifier yang digunakan untuk membaca kumpulan data dengan kategori dan struktur yang ditentukan dalam file excel (2017).

Penelitian Gururaj K S, Dr. K. Thippeswamy, Merancang model probabilistik untuk menganalisis tingkat kegagalan pemilihan dalam sistem pemilihan dengan melibatkan satu model Baye terkuat yang disebut sebagai Naive Bayes Model. Penelitian ini digunakan untuk mencoba memprediksi alasan kegagalan di tingkat pemungutan suara dengan model Probabilitas Naive Bayes dengan mempertimbangkan empat parameter utama (2018).

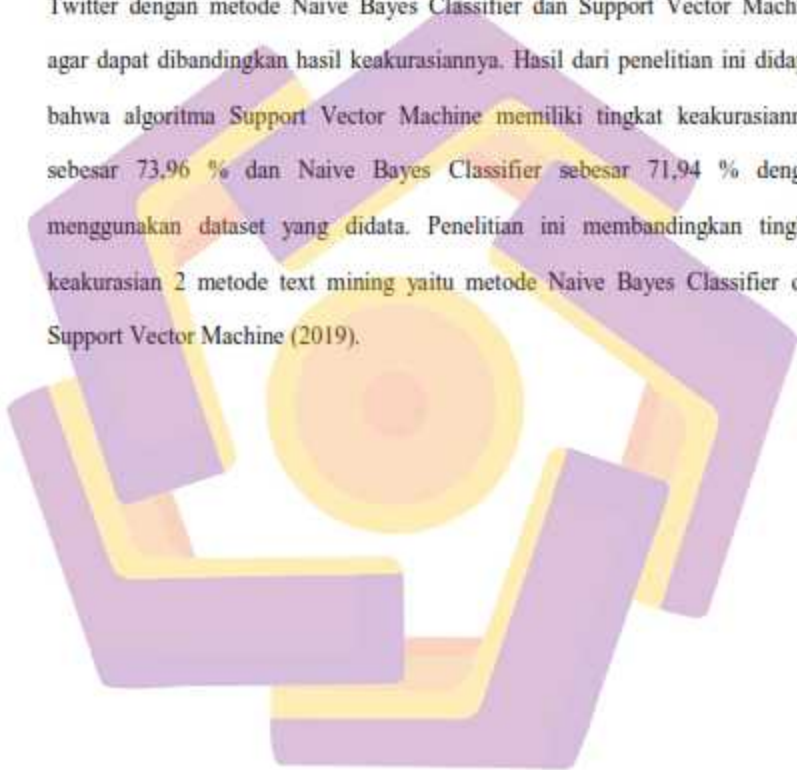
Penelitian Saputro, Notodiputro, Khairil Anwar, dan Indahwati, Mendeskripsikan hasil penambangan sentimen dan menentukan metode terbaik untuk memprediksi kelas sentimen. Persentase sentimen positif di 17 provinsi lebih besar dari sentimen negatif dan netral. Metode C5.0 menghasilkan prediksi yang lebih baik daripada Naive Bayes (2018).

Penelitian Chaudhary & Khumar, Untuk meningkatkan akurasi klasifikasi yaitu dengan mengoptimasi fitur yang telah diseleksi untuk mengurangi ukuran bagian fitur subset dan kompleksitas komputasi Algoritma Bat dan Binary Bat Algorithm dengan algoritma SVM mampu meningkatkan akurasi, fitur perulangan dan pengganggu telah dihapus dari set fitur dan terhitung. Algoritma Bat dan Binary Bat Algorithm telah memberikan hasil yang lebih baik (2016).

Penelitian Singh, Gupta, dan Singh, Menganalisa dan membandingkan implikasi signifikan opini tweet yang diungkapkan dalam kalimat positif dan

negatif. Penelitian ini menggunakan algoritma Naive Bayes dengan sebuah library Python yang dinamakan Textblob dan Twython package. Namun Penelitian ini belum mengungkap tweet yang menggunakan kalimat yang sifatnya netral (2017).

Penelitian Rusdian dan Rosiya, Membandingkan hasil ekstraksi API Twitter dengan metode Naive Bayes Classifier dan Support Vector Machine agar dapat dibandingkan hasil keakurasiannya. Hasil dari penelitian ini didapat bahwa algoritma Support Vector Machine memiliki tingkat keakurasiannya sebesar 73,96 % dan Naive Bayes Classifier sebesar 71,94 % dengan menggunakan dataset yang didata. Penelitian ini membandingkan tingkat keakurasi 2 metode text mining yaitu metode Naive Bayes Classifier dan Support Vector Machine (2019).



2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian Pengukuran Kinerja Algoritma Naive Bayes, Decision Tree dan K-Nearest Neighbor (K-NN) Untuk Sentimen Analisis

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	The Effectiveness of Bat Algorithm for Data Handling in Various Applications	Rozlini Mohamed, Munirah Mohd. Yusof, Noorhaniza Wahid, 2016 6th IEEE International Conference on Control System, Computing and Engineering, 25-27 November 2016, Penang, Malaysia	Melakukan pengujian tingkat efektivitas pengolahan data untuk Algoritma Bat melalui tipe data dan ukuran atribut dalam berbagai dataset menggunakan algoritma Naive Bayes, Decision Tree dan K-NN	<ul style="list-style-type: none"> - Ciri khas tipe data ditemukan hampir disemua data yang menggunakan Algoritma Bat sebagai seleksi fitur - Ketidaktepatan tipe data dan nilai atribut tidak berhubungan dengan persentase pengurangan atribut dan performa klasifikasinya, artinya persentase pengurangan atribut tinggi tetapi performa hasil klasifikasinya rendah - Suatu tipe data memiliki hubungan dengan kinerja klasifikasi dibandingkan nilai pengurangan atributnya. Meskipun persentase pengurangan atributnya tinggi tetapi hal ini menghasilkan kinerja klasifikasi yang lebih rendah 	<ul style="list-style-type: none"> - Tidak menggunakan teknik validasi - Tidak menggunakan aplikasi pemodelan - Tidak menunjukkan secara detail tools yang digunakan - Tidak membahas sentimen analisis 	<ul style="list-style-type: none"> - Penelitian berikutnya menggunakan algoritma Naive Bayes, Decision Tree dan K-Nearest Neighbor dengan optimasi Algoritma Bat pada sentimen analisis - obyek penelitian adalah pemilihan kepala daerah - Tools yang digunakan adalah RapidMiner - Penelitian selanjutnya menggunakan pengujian validasi dengan Cross Validation.
2	Bat Inspired Sentiment Analysis of Twitter Data	Himja Khurana, Sanjib Kumar Sahu, Progress in Advanced Computing and Intelligent Engineering, Springer Nature Singapore Pte Ltd, 2018	Menerapkan sebuah pendekatan supervised learning untuk membuat sentimen analisis	Hasil empiris menunjukkan study tentang bagaimana meningkatkan efisiensi pada algoritma Support Vector Machine	<ul style="list-style-type: none"> - Tidak menggunakan algoritma Naive Bayes classifier, Decision Tree dan K-NN 	<ul style="list-style-type: none"> - Penelitian selanjutnya menggunakan naive bayes classifier, Decision Tree dan K-NN - Pengujian validasi dengan teknik Cross Validation.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Sentiment classification of Roman-Urdu opinions using Naive Bayesian, Decision Tree and K-NN classification techniques	Muhammad Bilal, Huma Israr, Muhammad Shahid, Amin Khan, Journal of King Saud University – Computer and Information Sciences (2016) 28, 330–344	Menerapkan algoritma Naive Bayes, Decision Tree dan K-Nearest Neighbor untuk menilai sentimen dengan menggunakan bahasa Urdu	<ul style="list-style-type: none"> - Penelitian ini mengacu pada penambahan opini menggunakan bahasa Roman Urdu dengan menggunakan 3 algoritma klasifikasi - Data training berisi 150 opini positif dan 150 negatif - Hasilnya Naive Bayes memiliki kinerja yang terbaik dalam akurasi, lebih tinggi ketepatannya, lebih tinggi recall dan nilai dari F-Measure jika dibandingkan dengan Decision Tree dan K-NN 	<ul style="list-style-type: none"> - Tidak melakukan optimasi - Tidak menggunakan RapidMiner - Tidak ada pengujian validasi 	<ul style="list-style-type: none"> - Tools yang digunakan adalah RapidMiner - Melakukan optimasi menggunakan Algoritma Bat
4	Analisis Sentimen Tentang Opini Pilkada DKI 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes dan Pembobotan Emoji	Agnes Rossi Trisna Lestari, Rizal Setya Perdana, M. Ali Fauzi, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 1 No. 2, hlm 1718-1724, e-ISSN : 2548-964X, http://j-ptiik.ub.ac.id , 12 Desember 2017	Menganalisa dokumen teks yang mengandung unsur non-tekstual berupa emoji. Algoritma yang digunakan adalah Naive Bayes dengan pembaruan berupa penambahan pembobotan non-tekstual (emoji). Hasil dari pembobotan tekstual dan non tekstual yang dinormalisasi dengan metode min-max menghasilkan sentimen positif-negatif	Data Twitter tentang sebanyak 900 data <i>tweet</i> . Dari hasil pengujian akurasi, diperoleh 68,52% untuk kondisi pembobotan tekstual, 75,93% untuk pembobotan non-tekstual, dan 74,81% untuk kondisi penggabungan dengan nilai konstanta 0,5 untuk tekstual dan 0,5 untuk non-tekstual. Dari hasil pengujian pengaruh pembobotan non-tekstual disimpulkan bahwa pembobotan non-tekstual berpengaruh terhadap akurasi dan pengklasifikasian, dengan komposisi konstanta pengali terbaik ketika $\alpha=0,4$ dan $\beta=0,6$ sampai dengan $\alpha=0,1$ dan $\beta=0,9$.	<ul style="list-style-type: none"> - Penelitian pada emoji tidak menghasilkan sentimen netral - Penelitian ini diharapkan mampu menampilkan bentuk emoji yang bersifat netral 	<ul style="list-style-type: none"> - Penelitian berikutnya menggunakan algoritma Naive Bayes, Decision Tree dan K-NN dengan optimasi Algoritma Bat dengan software pengujian pada RapidMiner tools - Penelitian selanjutnya menggunakan teknik confusion matrik dalam memvalidasi tingkat akurasi penilaiannya

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Sentimen Analysis of Twitter Using Naive Bayes With Unigram Approach	P. Bavithra Matharasi, Dr. A.Senthilrajan, International Journal Of Scientific and Research Publication, Vol. 7, Issue 5, ISSN : 2250-3153, 5 May 2017	Membangun analisa Sentimen dengan menggunakan pendekatan Unigram dari Naive Bayes	Pengembangan Naive Bayes Classifier yang digunakan untuk membaca kumpulan data dengan kategori dan struktur yang ditentukan dalam file excel.	<ul style="list-style-type: none"> - Penelitian ini justru menghasilkan (secara umum) suatu analisa yang kurang dapat dihandle oleh algoritma Bayes - Penelitian ini tidak membahas tentang PIlkada - Penelitian ini tidak melakukan pengukuran tingkat validasi - Tidak membahas Decision Tree dan K-NN 	<ul style="list-style-type: none"> - Penelitian selanjutnya menggunakan algoritma Naive Bayes, Decision Tree dan K-NN dengan optimasi Algoritma Bat dengan software pengujian pada RapidMiner tools - Penelitian selanjutnya menggunakan pengujian validasi dengan teknik Cross Validation
6	Naive Bayes Model for Analysis of Voting Rate Failure in Election System	Gururaj K S, Dr. K. Thippeswamy, International Journal on Recent and Innovation Trends in Computing and Communication (IJRI TCC), Vol. 6, Issue 5, ISSN : 2321-8109, 171-173, May 2018,	Merancang model probabilistik untuk menganalisis tingkat kegagalan pemilihan dalam sistem pemilihan dengan melibatkan satu model baye terkuat yang disebut sebagai Naive Bayes Model	Penelitian ini digunakan untuk mencoba memprediksi alasan kegagalan di tingkat pemungutan suara dengan model Probabilitas Naive Bayes dengan mempertimbangkan empat parameter utama.	<ul style="list-style-type: none"> - Tidak Menggunakan Teknik validasi - Tidak menganalisa sentimen pemilih - Tidak membandingkan dengan metode Decision Tree dan K-NN - Tidak melakukan optimasi algoritma 	<ul style="list-style-type: none"> - Penelitian selanjutnya menggunakan algoritma Naive Bayes, Decision Tree dan KNN dengan optimasi Algoritma Bat dengan software pengujian pada RapidMiner tools - Penelitian selanjutnya menggunakan pengujian validasi dengan teknik Cross Validation

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7	Study Of Sentiment Of Governor's Election Opinion in 2018	Agung Eddy Suryo Saputro, Khairil Anwar Notodiputro, Indahwati, International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Vol. 4, Issue 11, ISSN : 2395-1990, DOI: 10.32628/ijstet21841124, November-Desember 2018	Mendeskripsikan hasil penambangan sentimen dan menentukan metode terbaik untuk memprediksi kelas sentimen.	<ul style="list-style-type: none"> - Persentase sentimen positif di 17 provinsi lebih besar dari sentimen negatif dan netral. - Metode C5.0 menghasilkan prediksi yang lebih baik daripada Naive Bayes 	<ul style="list-style-type: none"> -Tidak menggunakan teknik validasi - Tidak menggunakan aplikasi pemodelan - Tidak menggunakan algoritma KNN 	<ul style="list-style-type: none"> - Penelitian berikutnya menggunakan algoritma Naive Bayes, Decision Tree dan K-NN dengan optimasi Algoritma Bat dengan software pengujian pada RapidMiner tools - Penelitian selanjutnya menggunakan pengujian validasi dengan teknik Cross Validation.
8	Sentiment Analysis using Nature-Inspired Algorithm	Shweta Chaudhary, A Dissertation On Master of Technology In Software Engineering, Roll No. 2K14/SWE/17, 2016	Mencrapkan Algoritma Bat pada Algoritma SVM untuk meningkatkan nilai akurasi	<ul style="list-style-type: none"> - Untuk meningkatkan akurasi klasifikasi yaitu dengan mengoptimasi fitur yang telah diseleksi untuk mengurangi ukuran bagian fitur subset dan kompleksitas komputasi - Algoritma Bat dan Binary Bat Algorithm dengan algoritma SVM mampu meningkatkan akurasi - fitur perulangan dan pengganggu telah dihapus dari set fitur dan terhitung. - Algoritma Bat dan Binary Bat Algorithm telah memberikan hasil yang lebih baik 	<ul style="list-style-type: none"> -Tidak menggunakan algoritma Naive Bayes Classifier, Decision Tree dan KNN 	<ul style="list-style-type: none"> - Penelitian selanjutnya menggunakan Naive Bayes Classifier, Decision Tree dan K-NN - Pengujian validasi dengan teknik Cross Validation.

Tabel 2.1. Matriks literatur review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
9	Sentiment Analysis Of Twitter User Data On Punjab Legislative Assembly Election, 2017	Akhilesh Kumar Singh, Deepak Kumar Gupta, Raj Mohan Singh; International Journal Of Modern Education and Computer Science (IJMECS); Vol. 9, No.9, pp. 60-68, 2017, DOI: 10.5815/ijmeecs.2017.09.07	Menganalisa dan membandingkan implikasi signifikan opini tweet yang diungkapkan dalam kalimat positif dan negatif.	Popularitas tweet dapat dikespresikan pada tingkat yang berbeda, apakah diungkapkan dalam bentuk opini dengan kalimat positif ataukah negatif.	<ul style="list-style-type: none"> - Penelitian ini belum mengungkap tweet yang menggunakan kalimat yang sifatnya netral. - Kelemahan pada penggunaan berbagai bahasa lokal - Kurangnya penerapan pada strategi kualifikasi 	<ul style="list-style-type: none"> - Penelitian berikutnya menggunakan algoritma Naive Bayes, Decision Tree dan K-NN dengan optimasi Algoritma Bat dengan software pengujian pada RapidMiner tools - Penelitian selanjutnya menggunakan teknik Cross Validation dalam memvalidasi tingkat akurasi penilaiannya
10	Analisa Sentimen Terhadap Tokoh Publik Menggunakan Metode Naive Bayes Classifier dan Support Vector Machine, 2019	Deni Rusdian, Didi Rosiyadi, Journal of Computer Engineering System and Science (CESS); Vol. 4 No. 2; p-ISSN : 2502-7131, e-ISSN: 2502-714x, 2 Juli 2019	Membandingkan hasil ekstraksi API Twitter dengan metode Naive Bayes Classifier dan Support Vector Machine agar dapat dibandingkan hasil keakurasiannya	Hasil dari penelitian ini didapat bahwa algoritma Support Vector Machine memiliki tingkat keakurasi sebesar 73,96 % dan Naive Bayes Classifier sebesar 71,94 % dengan menggunakan dataset yang didapat.	<ul style="list-style-type: none"> - Penelitian ini tidak meneliti tentang pemilihan kepala daerah - Pemilihan tokoh public bersifat acak sehingga kurang fokus - Hanya menguji tingkat akurasi tanpa ada proses validasi nilai akurasi. - Tidak menggunakan algoritma, Decision Tree dan K-NN 	<ul style="list-style-type: none"> - Penelitian selanjutnya menggunakan algoritma Naive Bayes, Decision Tree dan K-NN dengan optimasi Algoritma Bat dengan software pengujian pada RapidMiner tools Serta menggunakan teknik Cross Validation dalam memvalidasi tingkat akurasi penilaiannya

2.3. Landasan Teori

2.3.1. Text Mining

Text mining juga dikenal sebagai *data mining* teks (Hearst 1997) atau penemuan pengetahuan dari database tekstual (Feldman & Dagan 1995). Sesuai dengan buku *The Text Mining Handbook* (Feldman & Sanger 2007), *text mining* dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining*. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen.

Jadi, sumber data yang digunakan dalam *text mining* adalah sekumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari *text mining* antara lain yaitu pengkategorisasian teks dan pengelompokkan teks (Triawati, et al. 2009). *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian/pengelompokkan dan menganalisa *unstructured data* dalam jumlah besar, dalam hal ini data yang akan digunakan adalah data yang diambil dari Twitter.

Dalam memberikan solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti *Data Mining*, *Information Retrieval*, Statistik dan Matematik, *Machine Learning*, *Linguistic*, *Natural Language Processing* dan *Visualization*. Kegiatan riset untuk *text mining* antara lain ekstraksi dan penyimpanan teks, *preprocessing* akan konten teks, pengumpulan data.

2.3.2. Term Weighting TF-IDF / Pembobotan Kata

Pada data yang kelasnya besar, skema yang paling sukses dan secara luas digunakan untuk pemberian bobot term adalah skema pembobotan atau *Term Weighting TF-IDF*. Kelemahan scoring dengan Jaccard coefficient adalah tidak disertakannya frekuensi suatu term dalam suatu dokumen, maka diperlukan scoring dengan kombinasi *Term Weighting TF-IDF*.

Hal yang perlu diperhatikan dalam pencarian informasi dari koleksi dokumen yang heterogen adalah pembobotan term. Term dapat berupa kata, frase atau unit hasil indexing lainnya dalam suatu dokumen yang dapat digunakan untuk mengetahui konteks dari dokumen tersebut, maka untuk setiap kata tersebut diberikan indikator, yaitu *term weight*.

1. Term Frequency (TF)

TF (Term Frequency) adalah frekuensi dari kemunculan sebuah term dalam dokumen yang bersangkutan. Semakin besar jumlah kemunculan suatu term (TF tinggi) dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar.

Pada Term Frequency (TF), terdapat beberapa jenis formula yang dapat digunakan :

1. TF biner (binary TF), hanya memperhatikan apakah suatu kata atau term ada atau tidak dalam dokumen, jika ada diberi nilai satu (1), jika tidak diberi nilai nol (0).

2. TF Murni (raw TF), nilai TF diberikan berdasarkan jumlah kemunculan suatu term di dokumen. Contohnya, jika muncul lima (5) kali maka kata tersebut akan bernilai lima (5).
3. TF Logaritmik, hal ini untuk menghindari dominansi dokumen yang mengandung sedikit term dalam query, namun mempunyai frekuensi yang tinggi. Rumus TF logaritmik ini ditunjukkan pada Rumus 1 dibawah ini :

$$TF = \begin{cases} 1 + \log_{10}(f_{t,d}), & f_{t,d} > 0 \\ 0, & f_{t,d} = 0 \end{cases} \dots\dots\dots 1$$

Dimana nilai $f_{t,d}$ adalah frekuensi term (t) pada document (d). Jadi jika suatu kata atau term terdapat dalam suatu dokumen sebanyak 5 kali maka diperoleh bobot = $1 + \log(5) = 1.699$. Tetapi jika term tidak terdapat dalam dokumen tersebut, bobotnya adalah nol (0).

4. TF Normalisasi, menggunakan perbandingan antara frekuensi sebuah term dengan nilai maksimum dari keseluruhan atau kumpulan frekuensi term yang ada pada suatu dokumen. Rumus TF Normalisasi ini ditunjukkan pada Rumus 2 dibawah ini :

$$TF = 0.5 + 0.5 x \left[\frac{f_{t,d}}{\max\{f_{t',d+t'} \mid d \in d\}} \right] \dots\dots\dots 2$$

2. Inverse Document Frequency (IDF)

IDF (Inverse Document Frequency) merupakan sebuah perhitungan dari bagaimana term didistribusikan secara luas pada koleksi dokumen yang bersangkutan. IDF menunjukkan hubungan ketersediaan sebuah term dalam seluruh dokumen. **Semakin sedikit jumlah dokumen yang mengandung term**

yang dimaksud, maka nilai IDF semakin besar. Sedangkan untuk Inverse Document Frequency (IDF) dihitung dengan menggunakan formula yang ditunjukkan pada rumus 3 dibawah ini :

$$IDF_j = \log\left(\frac{D}{df_j}\right) \dots\dots\dots 3$$

Dimana **D** adalah jumlah semua dokumen dalam koleksi sedangkan **df_j** adalah jumlah dokumen yang mengandung term (**t_j**). Jenis formula **TF** yang biasa digunakan untuk perhitungan adalah **TF murni (raw TF)**. Dengan demikian rumus umum untuk **Term Weighting TF-IDF** adalah penggabungan dari formula perhitungan **raw TF** dengan formula **IDF** dengan cara mengalikan nilai **TF** dengan nilai **IDF** seperti yang ditunjukkan pada rumus 4 dan 5 dibawah ini :

$$w_{ij} = tf_{ij} \times idf_j \dots\dots\dots 4$$

$$w_{ij} = tf_{ij} \times \log\left(\frac{D}{df_j}\right) \dots\dots\dots 5$$

Dimana **w_{ij}** adalah bobot term (**t_j**) terhadap dokumen(**d_i**). Sedangkan **tf_{ij}** adalah jumlah kemunculan term (**t_j**) dalam dokumen (**d_i**). **D** adalah jumlah semua dokumen yang ada dalam database dan **df_j** adalah jumlah dokumen yang mengandung term (**t_j**) (minimal ada satu kata yaitu term (**t_j**)).

Berapapun besarnya nilai **tf_{ij}**, apabila **D = df_j**, maka akan didapatkan hasil 0 (nol), dikarenakan hasil dari log 1, untuk perhitungan IDF. Untuk itu dapat ditambahkan nilai 1 pada sisi IDF, sehingga perhitungan bobotnya seperti ditunjukkan pada rumus 6 dibawah ini :

$$w_{ij} = t f_{ij} \times \log\left(\frac{v}{df_j}\right) + 1 \dots\dots\dots 6$$

2.3.3. Algoritma Optimasi

Algoritma adalah setiap prosedur komputasi yang terdefinisi dengan baik dengan mengambil beberapa nilai, atau seperangkat nilai-nilai, sebagai masukan dan menghasilkan beberapa nilai, atau seperangkat nilai-nilai sebagai output. Algoritma juga dipandang sebagai alat bantu dalam memecahkan masalah (Cormen, et al. 2001).

Istilah Optimasi diartikan sebagai upaya atau cara untuk memperoleh hasil yang terbaik (Badan Pengembangan dan Pembinaan Bahasa 2016). Algoritma yang optimal adalah algoritma yang mampu mengurutkan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dalam langkah-langkah terbatas, yang paling baik dan menguntungkan, yang menggambarkan prosedur komputasi tertentu untuk mencapai hubungan antara input dan output.

Algoritma Optimasi (Optimization Algorithms) dapat didefinisikan sebagai algoritma untuk menemukan nilai x sedemikian hingga nilai $f(x)$ bernilai sekecil atau sebesar mungkin untuk suatu fungsi f yang diberikan, yang dalam hal ini mungkin dapat disertai dengan beberapa batasan pada nilai x . Nilai x sendiri dapat berupa skalar atau vektor dari nilai-nilai kontinu maupun diskrit.

Algoritma optimasi sedikit berbeda dengan algoritma pencarian (search algorithm). Jika pada algoritma pencarian terdapat suatu kriteria tertentu yang menyatakan apakah elemen x , merupakan solusi atau bukan. Sebaliknya pada algoritma optimasi mungkin tidak terdapat kriteria tersebut, melainkan hanya

fungsi-fungsi objektif yang menggambarkan bagus tidaknya suatu konfigurasi yang diberikan.

Suatu algoritma dikatakan optimal yaitu jika tidak ada algoritma lain yang sekelasnya yang dapat melakukan lebih sedikit operasi dasar pada worstcase-nya. atau algoritma dianggap paling optimal adalah apabila dalam worstcase-nya melakukan perbandingan 2 entri dalam list yang paling sedikit. Cara menentukan optimalitas algoritma adalah dengan menentukan batas bawah dari jumlah operasi yang diperlukan untuk menyelesaikan masalah, jika ada algoritma yang mampu melakukan operasi sejumlah batas maka algoritma tersebut berarti optimal.

Salah satu cara memeriksa optimalitas suatu algoritma adalah membuat algoritma yang efisien. Efisiensi suatu algoritma ditentukan oleh 2 hal, yaitu :

1. Kecepatan (Waktu)

Dalam segi kecepatan, faktor-faktor yang mempengaruhinya adalah banyak langkah, tipe data dan operator-operator.

2. Space memori (Alokasi Memori)

Space (alokasi) memori dipengaruhi oleh struktur data dinamis, procedure/function call dan rekursif. (Hasad 2016).

2.3.4. Sentimen Analisis

Informasi tekstual secara umum dapat dibagi menjadi informasi fakta dan opini (Liu, 2010). Fakta adalah ekspresi obyektif terhadap suatu benda, kejadian dan kepunyaan benda tersebut. Opini bisaanya berupa ekspresi subyektif yang

menggambarkan sentimen, penilaian, atau perasaan seseorang terhadap suatu benda, kejadian atau kepunyaan dari benda tersebut. (Pang dan Lilian 2008) menjelaskan bahwa analisis sentimen adalah bagian dari pekerjaan yang meninjau segala sesuatu berhubungan dengan pendapat komputasi, sentimen dan subjektivitas teks. Ditambahkan oleh (Dave, et al. 2003) bahwa analisis sentimen adalah alat untuk memproses koleksi hasil pencarian yang bertujuan dengan mencari atribut suatu produk (kualitas, fitur, dll) dan proses memperoleh hasil pendapatnya.

Tugas dasar dalam analisis sentimen adalah mengelompokkan polaritas dari teks yang ada dalam dokumen, apakah pendapat yang dikemukakan dalam dokumen bersifat positif, negatif atau netral (Dave, et al. 2003). Penelitian mengenai analisis sentimen telah berkembang sejak tahun 2003 dan merupakan bagian dari *text mining* yang merupakan penelitian komputasi berdasarkan sentimen, *emoticon*, pendapat, komentar dan setiap ekspresi yang diungkapkan oleh teks. Analisis sentimen difokuskan untuk *review* klasifikasi berdasarkan polaritas.

Berdasarkan klasifikasi, analisis sentimen dibagi menjadi dua kelompok utama. Yaitu dokumen klasifikasi ke pendapat atau fakta, atau dikenal sebagai klasifikasi subjektivitas (*subjectivity classification*) dan dokumen klasifikasi ke dalam positif atau negatif, atau dikenal sebagai analisis sentimen. Hal ini adalah proses yang penting untuk menentukan dokumen yang memiliki opini dan dokumen yang menyimpulkan opini bernilai positif, negatif maupun netral.

2.3.5. Naive Bayes

Naive Bayes Classifier merupakan sebuah metode klasifikasi dengan probabilitas sederhana yang mengaplikasikan Teorema *Bayes* dengan asumsi ketidaktergantungan (independen) yang tinggi. Penggunaan metode *Naive Bayes Classifier* pada penelitian ini didasarkan pada banyaknya dataset yang dipakai sehingga membutuhkan suatu metode yang mempunyai performansi yang cepat dalam pengklasifikasian serta keakuratan yang cukup tinggi (Larose 2006).

Keuntungan penggunaan *Naive Bayes Classifier* adalah metode ini hanya membutuhkan jumlah data pelatihan (*training data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian (Kao dan Poteet 2007). Metode *Naive Bayes Classifier* menempuh dua tahap dalam proses klasifikasi teks, yaitu tahap pelatihan dan tahap klasifikasi. Pada tahap pelatihan dilakukan proses terhadap sampel data yang sedapat mungkin dapat menjadi representasi data tersebut. Selanjutnya adalah penentuan probabilitas *prior* bagi tiap kategori berdasarkan sampel data. Pada tahap klasifikasi ditentukan nilai kategori dari suatu data berdasarkan *term* yang muncul dalam data yang diklasifikasi. Teorema *Naive Bayes* dapat dinyatakan dalam persamaan 7 dibawah ini :

$$P(X_k | Y) = \frac{P(Y|X_k)}{\sum_i P(Y|X_i)} \dots\dots\dots 7$$

Dimana, keadaan *Posterior* (Probabilitas X_k di dalam Y) dapat dihitung dari keadaan *prior* (Probabilitas Y di dalam X_k dibagi dengan jumlah dari semua probabilitas Y di dalam semua X_i). Untuk dapat mengklasifikasikan suatu *tweet*, dalam penelitian ini penulis menggunakan metode *Naive Bayes Classifier* untuk

klasifikasi teks, seperti yang dilakukan (Ricci, et al. 2011) pada persamaan 8 dibawah ini.

$$P(v_1|C = c) = \frac{\text{CountTerms}(v_1, \text{docsv}(c))}{\text{AllTerms}(\text{docs}(c))} \dots\dots\dots 8$$

Dimana v_1 dalam penelitian ini adalah satu kata tertentu dalam *tweet*, sedangkan $\text{CountTerms}(v_1, \text{docsv}(c))$ menunjuk pada jumlah kemunculan suatu kata berlabel c ("positif" atau "negatif" atau "netral"). $\text{AllTerms}(\text{docs}(c))$ menunjuk pada jumlah semua kata berlabel c yang ada pada dataset. Untuk menghindari adanya nilai nol pada probabilitas, maka diberlakukan *Laplace (add-one) smoothing*. Tujuan daripada *smoothing* adalah untuk mengurangi probabilitas dari hasil/keluaran yang terobservasi, dan juga sekaligus meningkatkan/menambah probabilitas hasil/keluaran yang belum terobservasi (Arguello, J, 2013), sehingga persamaan menjadi seperti yang ditunjukkan pada rumus 9 dibawah ini :

$$P(v_1|C = c) = \frac{\text{CountTerms}(v_1, \text{docsv}(c))}{\text{AllTerms}(\text{docs}(c)) + |V|} \dots\dots\dots 9$$

Dimana $|V|$ menunjuk pada jumlah semua kata dalam *tweet* yang ada di dataset.

2.3.6. Decision Tree

Decision Tree dibuat dengan memisahkan ruang vektor dari fitur dokumen kedalam bagian-bagian terpisah secara interaktif. Pada Setiap tahapan, suatu pemisahan bagian melakukan perbaikan atas kesalahan pada bagian data training

serta didasarkan dari strategi penguasaan serta bagian cabang yang dibuat. Untuk data yang baru, sebuah label diprediksi dengan membaca bagian cabang dari titik pusat sebagaimana nilai pada fitur dari data yang baru.

Decision Tree sangatlah mudah untuk diterjemahkan, sebagaimana struktur suatu cabang yang dapat digambarkan secara grafis, dan kita dapat mengikuti arah yang menurun pada cabang yang merujuk pada input variabel untuk mengurangi waktu yang dibutuhkan pada tahap pelatihan (Sharma dan Dey 2012).

Decision Tree classifier menghasilkan sebuah hirarki ruang ketidaktepatan dari data training dimana suatu kondisi dari nilai atribut digunakan untuk memilahkan data. Suatu kondisi atau predikat adalah keberadaan satu atau lebih suatu kata. Pembagian data dikatakan selesai hingga titik daun berisi nilai minimum dari record sehingga dapat digunakan untuk menunjukkan maksud dari pengelompokan (Pradhan, Vala dan Balani 2016).

2.3.7. K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) Classifier adalah suatu instan yang didasarkan pada pengelompokan pada label klas dari data training yang sesuai dengan data uji. Selanjutnya, hal ini tidak membentuk sebuah model yang terbuka pada klas. Sebuah instan dikelompokkan dari kesamaan data dengan tetangganya, dengan suatu instan yang dikelompokkan dari sebuah persamaan opini dari tetangganya, dengan instan yang dikelompokkan ke dalam klas yang umum digunakan pada nilai K tetangga terdekat (k adalah sebuah nilai positif). Jika $k=1$

kemudian instan didaftarkan berdasarkan kesamaan per klas dari tetangga terdekatnya. Memberikan sebuah data uji d , KNN menemukan K tetangga terdekat diantara data training. Kesamaan skor masing-masing data tetangga terdekat terhadap data uji digunakan sebagai bobot klas data tetangga. Bobot jumlah dalam KNN yang dikelompokkan dapat ditulis dengan persamaan yang ditunjukkan pada rumus 10 dibawah ini :

$$Score(d, c) = \sum_{d_j \in KNN(d)} sim(d, d_j) \delta(d_j, c_l) \dots\dots\dots 10$$

Tingkatan KNN(d) menunjukkan data set k tetangga terdekat atas data d . Jika d_j menjadi milik c_l , $\delta(d_j, c_l) = 1$, atau 0. Sebuah data uji d , seharusnya menjadi menjadi milik klas yang memiliki hasil lebih tinggi dari jumlah bobot. (Sharma dan Dey 2012). K-Nearest Neighbor juga digunakan untuk pengklasifikasian data secara luas pada kelompok Tweet. Banyak peneliti menggunakan K-NN untuk mengklasifikasikan tweets. K-NN merupakan algoritma paling sederhana. K-NN secara instan didasarkan pada algoritma pembelajaran dimana seluruh komputasi ditahan hingga dilakukan pengelompokan. (Jain dan Katkar 2015).

2.3.8. Algoritma Bat

Salah satu algoritma heuristik baru yang diadaptasi dari perilaku echolocation kelelawar dalam mencari makanan. Kemampuan echolocation ini membuat kelelawar dapat membedakan rintangan dengan sumber makanan.

Dalam Algoritma Bat, ada dua hal penting yakni kebisingan dan pulse rate. Kebisingan digunakan untuk mendeteksi jauh dekatnya jarak dengan sumber makanan. Semakin dekat dengan sumber makanan maka tingkat kebisingan akan menurun, sedangkan pulse rate merupakan sinyal yang didapat dari pantulan emisi sonar yang dimiliki kelelawar terhadap rintangan. Semakin dekat kelelawar dengan rintangan maka pulse rate semakin meningkat. Solusi yang lebih minimum berusaha dicari melalui proses movement dan local search.

2.3.8.1 Perilaku Kelelawar

Menurut Chaudhary & Kumar, 2016, Kelelawar adalah mamalia yang menggunakan panca indra nya dalam menemukan mangsanya dan mampu memilih serta mangsa yang terbaik didalam kegelapan malam. Perilaku Kelelawar mampu mengontrol keseimbangan antara eksploitasi dan eksplorasi. Kelelawar menggunakan jenis sonar yang disebut sebagai echolokasi untuk mendeteksi mangsa, menghindari hambatan dan menemukan celah bertengger mereka dikegelapan. Kelelawar ini mengeluarkan sonar yang sangat keras untuk mendengarkan nadi dan gema yang memantul kembali dari benda-benda disekitarnya. Dan tiap spesies memiliki denyut nadi yang berbeda-beda (Khurana & Suhu, 2018).

2.3.8.2. Kualitas Suara Echolocation

Menurut Chaudhary & Kumar, 2016, Meskipun setiap sonar hanya berlangsung beberapa ribu detik (hingga sekitar 8 hingga 10 ms), Namun, ia memiliki frekuensi konstan yang biasanya di wilayah 25kHz hingga 150 kHz. Rentang frekuensi tipikal untuk sebagian besar spesies kelelawar ada di wilayah

tersebut 25kHz dan 100kHz, meskipun beberapa spesies dapat memancarkan frekuensi yang lebih tinggi hingga 150 kHz.

Setiap ledakan ultrasonik biasanya berlangsung 5 hingga 20 ms, dan microbats memancarkan sekitar 10 hingga 20 suara seperti itu meledak setiap detik. Saat berburu mangsa, laju emisi pulsa bisa dipercepat hingga sekitar 200 pulsa per detik ketika mereka terbang di dekat mangsanya. Sangat singkat semburan suara menyiratkan kemampuan fantastis kekuatan pemrosesan kelelawar dari kelelawar. Faktanya, studi menunjukkan waktu integrasi telinga kelelawar biasanya sekitar 300 hingga 400 μ s. Sebagai kecepatan suara di udara biasanya $v = 340$ m / s, panjang gelombang λ ultrasonik suara meledak dengan frekuensi konstan f diberikan oleh seperti pada persamaan 11 dibawah ini :

$$\lambda = v / f \dots\dots\dots 11$$

Ini berada dalam kisaran 2mm hingga 14mm untuk rentang frekuensi tipikal dari 25 kHz hingga 150 kHz. Panjang gelombang seperti itu berada dalam urutan yang sama dengan ukuran mangsanya.

Hebatnya, pulsa yang dipancarkan bisa sekeras 110 dB, dan, untungnya, mereka masuk wilayah ultrasonik. Kerasnya juga bervariasi dari yang paling keras saat mencari mangsa dan ke pangkalan yang lebih tenang saat menuju mangsa. Kisaran pancaran pulsa pendek seperti itu biasanya beberapa meter, tergantung pada frekuensi aktual. Kelelawar dapat mengelola untuk menghindari rintangan sekecil rambut manusia yang tipis. Studi menunjukkan bahwa kelelawar gunakan waktu tunda dari emisi dan deteksi gema, perbedaan waktu antara dua telinga

mereka, dan variasi nyaring gema untuk membangun tiga skenario dimensi sekitarnya.

Mereka dapat mendeteksi jarak dan orientasi dari target, jenis mangsa, dan bahkan kecepatan bergerak mangsa seperti serangga kecil. Memang, penelitian menunjukkan bahwa kelelawar tampaknya dapat membedakan target variasi efek Doppler yang diinduksi oleh tingkat sayap-mengepak dari target serangga. Jelas, beberapa kelelawar memiliki penglihatan yang bagus, dan sebagian besar kelelawar juga sangat indera penciuman yang sensitif.

Pada kenyataannya, mereka akan menggunakan semua indera sebagai kombinasi memaksimalkan deteksi efisien mangsa dan navigasi yang lancar. Namun, disini dia hanya tertarik pada ekolokasi dan perilaku yang terkait. Echolocation seperti perilaku kelelawar itu dapat dirumuskan sedemikian rupa sehingga dapat dikaitkan dengan fungsi objektif untuk dioptimalkan, dan ini memungkinkan untuk merumuskan algoritma pengoptimalan yang baru (Khurana & Suhu, 2018).

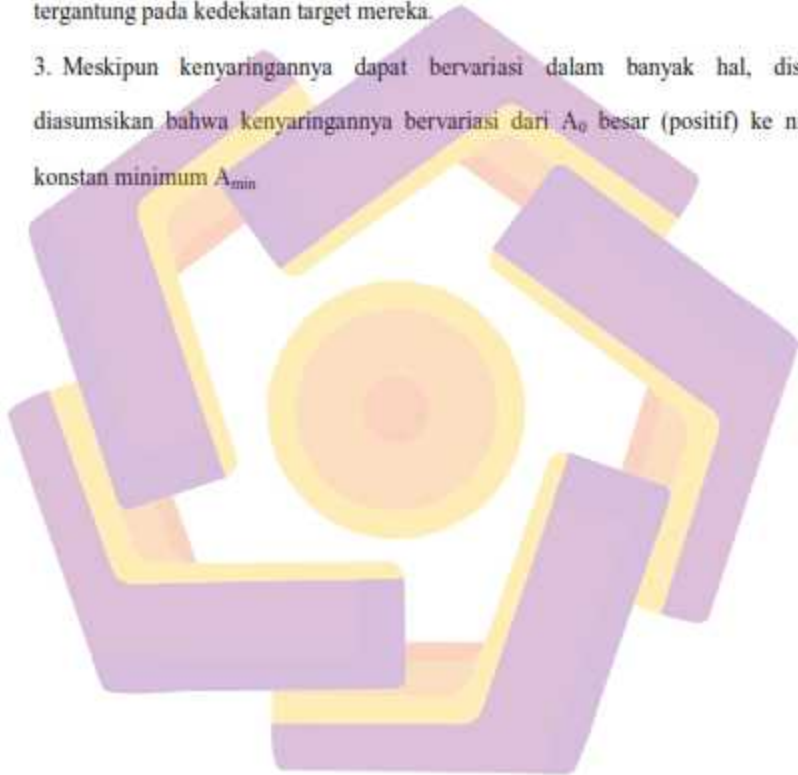
2.3.8.3. *Kondisi awal*

Jika kita mengidealkan beberapa karakteristik ekolokasi kelelawar, kita dapat mengembangkan beragam Algoritma Bat-terinspirasi atau Algoritma Bat. Dalam algoritma dasar kelelawar dikembangkan oleh Xin-She Yang (2010), aturan perkiraan atau ideal yang berikut ini digunakan adalah

1. Semua kelelawar menggunakan ekolokasi untuk merasakan jarak, dan mereka juga 'tahu' perbedaannya antara makanan/mangsa dan latar belakang hambatan dalam beberapa cara ajaib.

2. Kelelawar terbang secara acak dengan kecepatan v_i pada posisi x saya dengan f_{min} frekuensi, bervariasi panjang gelombang dan kenyaringan A_0 untuk mencari mangsa. Mereka dapat secara otomatis menyesuaikan panjang gelombang (atau frekuensi) dari pulsa yang dipancarkan dan sesuaikan laju emisi pulsa $r \in [0, 1]$, tergantung pada kedekatan target mereka.

3. Meskipun kenyaringannya dapat bervariasi dalam banyak hal, disini diasumsikan bahwa kenyaringannya bervariasi dari A_0 besar (positif) ke nilai konstan minimum A_{min}



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Adapun Jenis, Sifat dan Pendekatan pada Penelitian ini adalah

1. Identifikasi Masalah

Pada tahap ini dilakukan proses identifikasi masalah atas hal-hal/tema yang berhubungan dengan tweet para netter.

2. Analisis dan Pemetaan Masalah

Pada tahap ini analisis terhadap dataset serta model algoritma Naive Bayes yang akan diuji serta menghitung validitas nilainya menggunakan Cross Validation.

3. Studi Literatur

Pada tahap ini, penelitian dilakukan dengan mencari literatur-literatur yang berhubungan dengan penelitian terdahulu. Literatur tersebut dapat berupa buku Tesis, Jurnal, Proceeding, Hasil seminar baik tingkat lokal, nasional maupun internasional.

4. Preparasi Data

Pada tahap ini, dilakukan suatu persiapan data sebagai data uji yang akan digunakan dalam penelitian. Pada tahap ini pula data-data dari API Twitter tersebut dikumpulkan.

5. Pengelompokan Data

Pada tahap ini, dilakukan pengelompokan data dari API Twitter yang telah dikumpulkan. Pengelompokan data sesuai dengan kriteria pemetaan masalah yang telah disusun.

6. Pemodelan Data

Pada tahap ini, dilakukan proses pemodelan data pada tools RapidMiner sehingga dapat dihitung nilai akurasi dan persentasenya.

7. Evaluasi

Pada tahap ini, dilakukan evaluasi atas data yang telah dimodelkan sebelumnya. Data-data yang telah dihitung kemudian dilakukan evaluasi untuk menghindari kesalahan perhitungan dan pengelompokan.

8. Hasil Evaluasi

Pada tahap ini, dilakukan dokumentasi atas hasil evaluasi untuk dijabarkan dalam bentuk statistik, tabel serta kesimpulan.

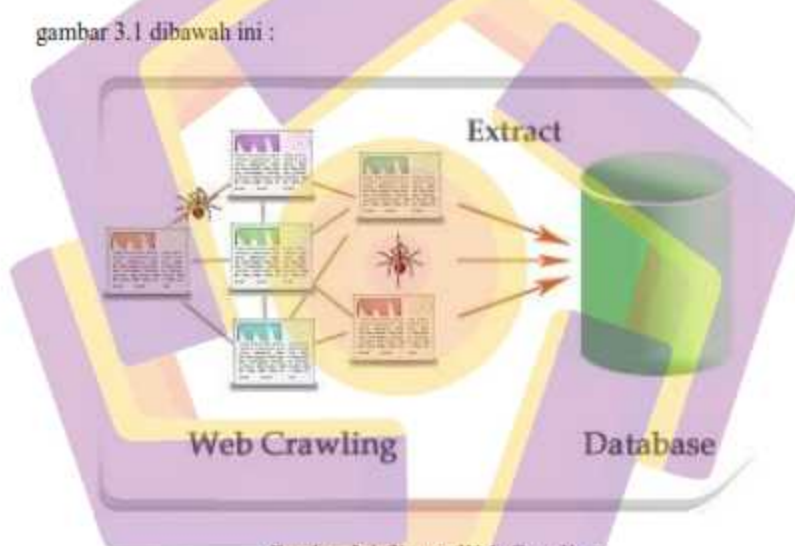
3.2. Metode Pengumpulan Data

Adapun metode pengumpulan data pada penelitian ini adalah

3.2.1. Crawling

Crawling adalah suatu teknik yang digunakan untuk mengumpulkan informasi yang ada dalam web. Crawling bekerja secara otomatis, dimana informasi yang dikumpulkan berdasarkan atas kata kunci yang diberikan oleh pengguna. Alat yang digunakan untuk melakukan crawling disebut dengan crawler.

Crawler berbentuk program yang diprogram dengan algoritma tertentu, sehingga dapat melakukan pemindaian ke halaman-halaman web, sesuai dengan alamat web ataupun kata kunci yang diberikan pengguna. Pada saat melakukan pemindaian, crawler akan membaca teks yang ada, hyperlink dan berbagai tag yang digunakan di halaman web tersebut. Berdasarkan informasi tersebut, crawler akan mengindeks informasinya atau pun menyimpan informasi tersebut ke dalam sebuah file atau ke dalam database. Skema Web Crawling ditunjukkan pada gambar 3.1 dibawah ini :



Gambar 3.1 Skema Web Crawling

Twitter crawling merupakan penerapan teknik crawling pada media sosial Twitter. Teknik ini bertujuan untuk mengumpulkan data tweet dari media sosial Twitter. Dari data yang telah dikumpulkan, dapat dianalisa dan diambil informasi penting yang bermanfaat bagi pengguna. Untuk menerapkan teknik twitter crawling ini, pihak Twitter telah memberikan akses bagi pengguna untuk memanfaatkan Twitter API. Sehingga dengan memanfaatkan Twitter API

tersebut, pengguna bisa dengan mudah memperoleh data-data seperti tweet, data pengguna dan lain-lain. Untuk selanjutnya dikumpulkan dan disimpan dalam sebuah file atau basis data. Dalam penelitian ini Crawling ini akan dilakukan oleh program RapidMiner dan hasil crawling akan disimpan kedalam file excel.

3.2.2. Twitter API

Twitter API, merupakan API (Application Programming Interface) yang disediakan oleh Twitter untuk memfasilitasi pengguna agar dapat berinteraksi dengan data-data yang ada pada Twitter. Data-data ini misalnya adalah tweet, id pengguna, lokasi, waktu pembuatan tweet dan lain-lain. Untuk memanfaatkan Twitter API, pengguna harus menggunakan bahasa server side scripting seperti php, python, R dan lain-lain.

Dengan menggunakan bahasa-bahasa tersebut, pengguna dapat melakukan request kepada Twitter API, dan respon hasilnya dirupakan dalam format JSON. Agar komunikasi pengguna dengan Twitter API aman, maka Twitter menerapkan OAuth atau Open Authorization.

OAuth merupakan protokol terbuka yang memungkinkan pengguna untuk berbagi resource pribadi seperti foto, video, data pengguna dan lain-lain yang tersimpan di suatu situs web, dengan situs lain tanpa memberikan nama pengguna dan kata sandi pengguna tersebut. OAuth mengizinkan pengguna untuk memberikan akses kepada situs pihak ketiga untuk mengakses informasi mereka yang disimpan di penyedia layanan lain tanpa harus membagi izin akses atau keseluruhan data mereka. OAuth bergantung pada tiga set dari token dan secret yang didapat dari server dan client:

1. Consumer key dan consumer secret

Consumer key adalah unique identifier untuk client yang digunakan client untuk melakukan request untuk mendapatkan request tokens.

2. Request token dan request token secret

Request token adalah temporary onetime identifier yang diberikan oleh server untuk tujuan permintaan kepada user untuk melakukan grant permission pada client. Token secret digunakan untuk melakukan sign request untuk mendapatkan sebuah access token.

3. Access token dan access token secret

Access token adalah identifier untuk digunakan oleh client untuk melakukan akses ke resources milik user. Sebuah client bisa melakukan akses ke resources milik user selama token nya masih valid. Server bisa melakukan revoke kapanpun karena sudah expire atau user melakukan revoke secara manual. Secret digunakan untuk sign request ke resources yang di proteksi oleh akses user. Cara untuk mendapatkan consumer key, consumer secret, access token dan access secret adalah sebagai berikut:

1. Client melakukan request ke server menggunakan sebuah consumer key
2. Client menggunakan consumer key untuk mendapatkan sebuah request token dan secret.
3. Client melakukan redirect pada user ke server untuk grant permission untuk client melakukan akses ke resources milik user. Proses ini bisa terjadi jika request token telah diautentikasi.

4. Client melakukan request ke server untuk memberikan access token dan secret. Hasilnya merepresentasikan sebuah identifier dan shared secret yang client nya bisa gunakan untuk mengakses resources atas nama user.

5. Ketika membuat sebuah request untuk akses ke resource terproteksi, client menyertakan Authorization header yang berisi consumer key, access token, signature method dan sebuah signature, timestamp, sebuah nonce, dan untuk opsionalnya adalah versi dari OAuth yang digunakan. (Saputra 2017). Dalam penelitian ini proses OAuth dijalankan melalui program RapidMiner.

3.3. Metode Analisis Data

Adapun metode analisis data pada penelitian ini adalah

1. Analisis konten

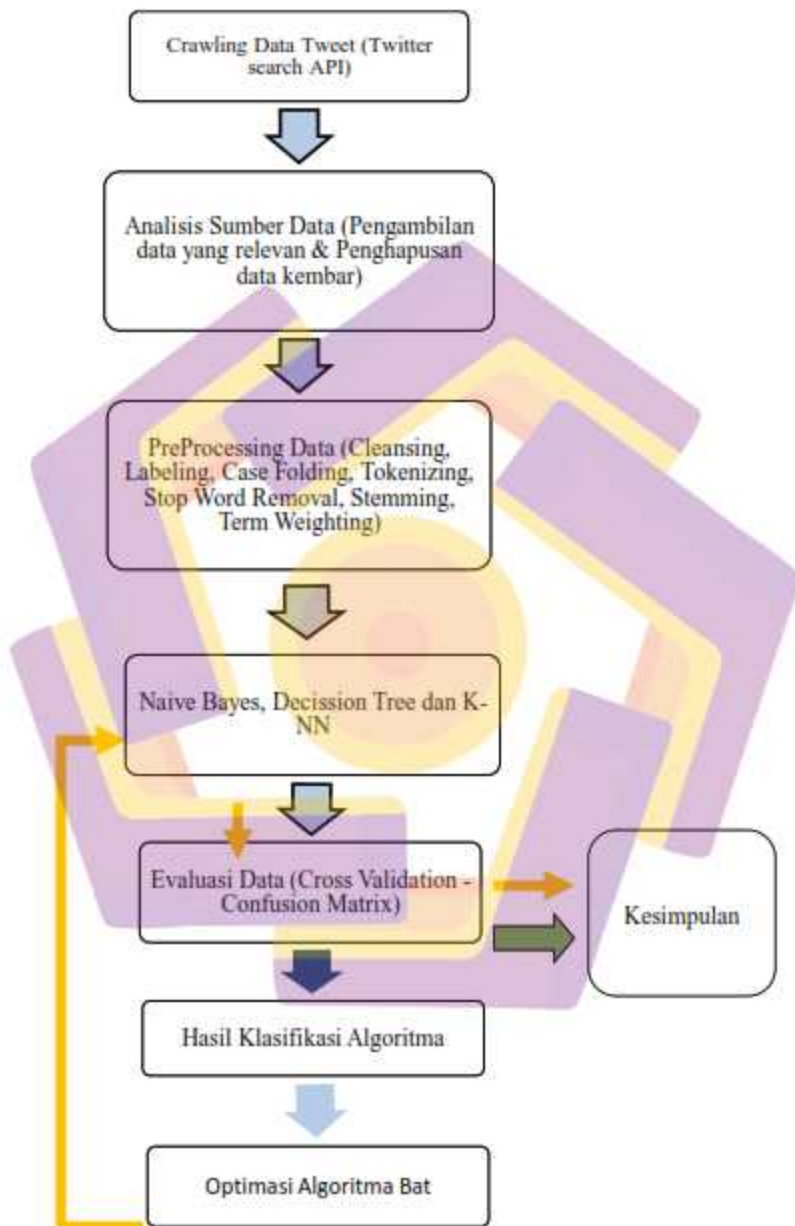
Metode ini membantu memahami keseluruhan komentar yang ada pada tweet. Metode ini membantu mengurai data tekstual yang ada agar dapat dikelompokkan menjadi kelompok sentimen.

2. Analisis naratif

Jenis metode ini berfokus pada cara bagaimana sebuah komentar/ide yang kemudian dikomunikasikan pada hal-hal yang terkait dengan pemilihan kepala daerah. Analisis ini menggunakan metode Naive Bayes, Decision Tree dan K-NN dengan optimasi menggunakan Algoritma Bat serta validasinya menggunakan metode Cross Validation.

3.4. Alur Penelitian

Berikut ini bagan dari alur penelitian seperti yang ditunjukkan pada gambar 3.2 :



Gambar 3.2. Alur Penelitian

Tahapan alur penelitian ini adalah

3.4.1. Crawling Data Tweet (Twitter search API)

Pada tahap ini dilakukan crawling data tweet dengan memanfaatkan Twitter search API (Application Programming Interface) untuk tweet yang berhubungan dengan pemilihan kepala daerah tahun 2020. Langkah awal yang dilakukan adalah Instalasi tools RapidMiner serta membuat pola crawling data twitter dengan akun twitter yang telah terhubung. Setelah menghubungkan akun twitter tersebut, maka selanjutnya melakukan registrasi ke <https://dev.twitter.com/apps/new> untuk mendapatkan kode akses API twitter.

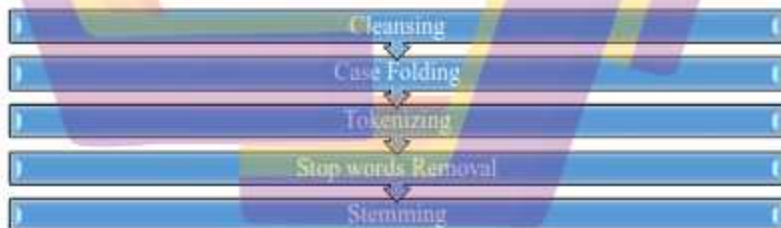
Crawling ini dilakukan dengan menentukan kata kunci apa yang sedang dicari serta menentukan jumlah tweet yang diinginkan. Setelah itu, data hasil *crawling* tersebut di *scrapping* dan disimpan dalam bentuk dokumen microsoft excel dengan extention .xlsx.

3.4.2. Analisis Sumber Data (Penghapusan data kembar dan pemilahan data yang relevan)

Pada tahap analisis sumber data ini dilakukan proses Penghapusan data yang kembar agar menambah tingkat akurasi hasil penelitian ini nantinya. Selanjutnya pada pemilahan data yang relevan adalah melakukan pemilahan kolom yang dibutuhkan pada penelitian ini dan membuang kolom yang tidak relevan. Dalam tahap ini total jumlah data yang dihasilkan sebanyak 3.000 tweet. Selanjutnya dari 3.000 data yang telah dikumpulkan kemudian dibagi menjadi 2 bagian yang terdiri dari 2700 data latih dan 300 data uji tweet.

3.4.3. PreProcessing Data (Cleansing, Labeling, Case Folding, Tokenizing, Stop Word Removal, Stemming)

Pada tahap ini dilakukan proses ekstraksi dokumen, teks pada proses *text mining* ini memiliki resiko noise data yang cukup tinggi serta memiliki struktur teks bahasa Indonesia yang kurang baik. Dalam hal ini data yang berasal dari Twitter memiliki kerumitan yang cukup tinggi karena ketidaksesuaian dengan ejaan yang baku dan kesalahan pada ejaan yang cukup tinggi pada penulisan tweetnya (Go Alec at all, 2009). Cara memahami suatu teks adalah dengan terlebih dahulu menentukan fitur-fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen. Menurut (Paurabek & Pak, 2010) sebelum menentukan fitur-fitur yang mewakili diperlukan tahap Preprocessing yang dilakukan secara umum dalam *text mining* pada dokumen, yaitu cleansing, labeling, case folding, tokenizing, stop words removal, stemming seperti yang ditunjukkan pada gambar 3.3.

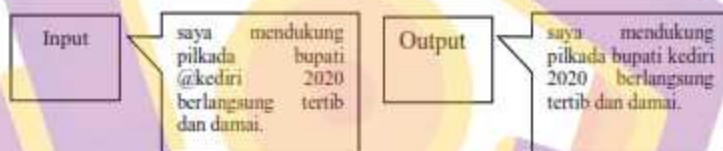


Gambar 3.3. Tahap PreProcessing Data

Adapun tahapan Preprocessing adalah sebagai berikut :

1. Cleansing

Tahap *Cleansing* masih termasuk dalam *preprocess* text dimana dalam tahap ini dilakukan penghapusan semua karakter html atau web yang tidak lagi memiliki makna yang berhubungan dalam pengklasifikasian dokumen. Hal ini disebabkan karena terkadang sebuah tweet menyertakan suatu alamat web didalamnya, sehingga jika tidak dihapus akan mengganggu proses klasifikasi. Pembersihan semua karakter html atau web ini seperti spesial karakter, URL link, username, serta *emoticon*. Berdasarkan (Go Alec at all, 2009), *emoticon* disini dihilangkan karena bisa menurunkan kualitas akurasi data yang signifikan. (Nurhuda, et al. 2013). Contoh karakter yang dihapus seperti @, #, \$, %, & dan lain-lain. Contoh hasil *cleansing* ditunjukkan pada gambar 3.4



Gambar 3.4. Contoh penerapan *Cleansing*

2. Labeling

Pada tahap ini dilakukan proses labeling data dengan cara mengklasifikasikan data tweet ke dalam sentimen positif, negatif dan netral. Proses ini dapat dilakukan secara manual maupun menggunakan aplikasi pihak ketiga seperti *Rosette Text Analytics*. Contoh bentuk pelabelan secara manual ditunjukkan pada tabel 3.1 dibawah ini :

Tabel 3.1. Contoh Pelabelan Data Tweet

Pernyataan	Labeling
Saya mendukung pilkada bupati @Kediri 2020 berlangsung tertib dan damai.	Positif

Calon bupati incumben menggunakan bantuan sembako pemerintah sebagai alat kampanye #boikotpilbup #kediri	Negatif
Dokter sukma berkunjung ke kampung Inggris Pare	Netral

Dari tahap ini nantinya akan didapatkan berapa jumlah tweet positif, negatif dan netral. Langkah selanjutnya, dari 3.000 data yang ada kemudian dibagi menjadi 1.000 tweet data sentimen positif, 1.000 tweet data sentimen negatif dan 1.000 tweet data sentimen netral.

3. Case Folding

Case Folding adalah proses merubah huruf kapital menjadi huruf kecil pada data tweet (Go Alec at all, 2009). Hal ini digunakan untuk mempermudah pencarian. Karena tidak semua tweet menggunakan huruf kapital. Oleh sebab itu lakukan keseragaman bentuk tulisan dalam model *lowercase*. Dalam tahap ini hanya huruf 'a' sampai 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter* seperti tanda baca dan angka. *Delimiter* merupakan karakter yang digunakan untuk membatasi atau memisahkan data yang disajikan dalam *plain text*. Contoh *case folding* ditunjukkan pada gambar 3.5:



Gambar 3.5. Contoh penerapan *case folding*

4. Tokenizing

Pada tahap ini yang dimaksud *Tokenizing* adalah sebuah proses pemotongan string input/tweet berdasarkan tiap kata yang menyusunnya (Feldman & Sanger, 2007). Proses ini memisahkan setiap kata yang menyusunnya. Secara umum

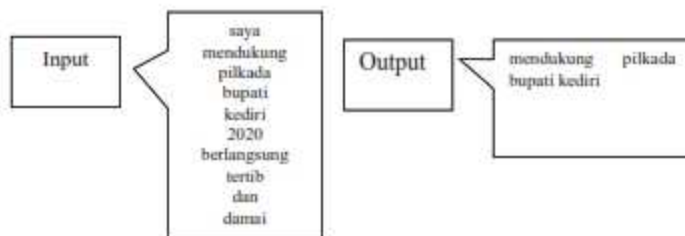
sebuah kata dipisah dengan spasi. Oleh sebab itu karakter spasi menjadi karakter penting dalam penentuan pemisahan suatu kata. Pada saat bersamaan tokenizing juga berfungsi untuk membuang bagian karakter tertentu yang dianggap sebagai tanda baca. Contoh proses *Tokenizing* dapat dilihat pada gambar 3.6 dibawah ini.



Gambar 3.6. Contoh penerapan tokenizing

5. Stop Words Removal

Setelah melalui proses tokenizing dimana telah dipisahkan kata per kata maka dilakukan proses *Stop Words Removal*. *Stop words Removal* adalah langkah penghilangan kata-kata yang tidak terkait dan berkontribusi pada topik data. Kata-kata yang termasuk dalam stop words tersebut memberikan pengaruh yang tidak baik pada proses text mining seperti pada kata-kata "bagaimana", "juga", "agar", "jadi" dan lain-lain (Cahyono, 2017). Pada proses stop words ini dibutuhkan kamus stop words untuk mendata kata-kata apa saja yang harus dibuang. Contoh penerapan *Stop words Removal* dapat dilihat pada gambar 3.7 dibawah ini.



Gambar 3.7. Contoh penerapan *Filtering/Stop Word Removal*

6. Stemming

Sebuah pembuatan indeks dilakukan karena suatu dokumen tidak dapat dikenali langsung oleh suatu *Information Retrival System* (IRS). Oleh karena itu, dokumen tersebut terlebih dahulu perlu dipetakan ke dalam suatu representasi dengan menggunakan teks yang berada didalamnya.

Teknik *Stemming* diperlukan guna memperkecil jumlah indeks yang berbeda dari suatu dokumen serta melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau form yang berbeda karena mendapat imbuhan yang berbeda. Sebagai contoh kata pendidikan, mendidik, terdidik, dididik, akan di *stem* ke root word-nya yaitu didik. Namun, seperti halnya kata making, kinerja stemming juga bervariasi dan sering tergantung pada domain bahasa lokal yang digunakannya.

Proses stemming disini juga melakukan perbaikan atas kata-kata gaul atau slang menjadi kata baku bahasa Indonesia. Untuk itu diperlukan kamus slangword dan kamus stemming. Contoh hasil proses *stemming* seperti yang ditunjukkan pada gambar 3.8 dibawah ini :

Gambar 3.8. Contoh penerapan *Stemming*

Pada tahap ini dilakukan penentuan Data Training dan Data Set. Data Training positif diambil dari kelompok positif, Data Training negatif diambil dari kelompok negatif dan data training netral diambil dari kelompok netral.

7. Term Weighting TF-IDF / Pembobotan Kata

Setelah proses stemming selesai dilakukan maka dilakukan proses Pembobotan Kata/Term Weighting TF-IDF agar tiap kata yang sudah terpisahkan tersebut dapat dihitung. Berikut ini adalah contoh tahapan dari metode **Term Weighting TF-IDF**. Proses ini dimulai dari simulasi proses stop words removal kemudian stemming. Contoh kata hasil terms query (Q) adalah gold, silver, truck Untuk koleksi dokumennya terdapat:

dokumen1(**d1**)= *Shipment of gold damaged in a fire*

dokumen2(**d2**)= *Delivery of silver arrived in a silver truck*

dokumen3(**d3**)= *Shipment of gold arrived in a truck*

Jadi total jumlah dokumen adalah koleksi dokumen (**D**) = 3

Untuk setiap query dan dokumen dalam koleksi, dilakukan pemotongan string berdasarkan tiap kata yang menyusunnya, menghilangkan tanda baca, angka dan stopword. Setelah melalui proses ini, maka kata "of", "in", dan "a" pada ketiga dokumen dihapus lalu di-stemming sehingga didapatkan term-term (documents terms) sebagai berikut:

- *ship - gold - damage - fire - deliver - silver - arrive - truck*

Pada tahap selanjutnya tiap dokumen diwujudkan sebagai sebuah vektor dengan elemen sebanyak term query yang terdapat dalam tiap dokumen yang berhasil dikenali dari tahap ekstraksi dokumen sebelumnya. Vektor tersebut

beranggotakan bobot dari setiap term query yang dihitung berdasarkan metode **Term Weighting TF-IDF**.

Fungsi metode ini adalah untuk mencari representasi nilai dari tiap dokumen dalam koleksi. Dari sini akan dibentuk suatu vektor antara dokumen dan query yang ditentukan oleh nilai bobot term query dalam dokumen.

Semakin besar nilai perhitungan bobot yang diperoleh maka semakin tinggi tingkat similaritas dokumen terhadap query. Contohnya untuk perhitungan **bobot(w)** term query *silver* dalam dokumen2 (**d2**) = *Delivery of silver arrived in a silver truck*, yaitu: jumlah kemunculan term silver dalam dokumen 2 (**d2**) adalah sebanyak dua kali (**tf = 2**), total dokumen yang ada di koleksi sebanyak tiga dokumen (**D**)=3, dari ketiga dokumen dalam koleksi, term silver muncul pada dokumen 2 (**d2**) saja, sehingga total dokumen yang mengandung term silver adalah satu dokumen (**df**)=1, sehingga dapat diperoleh nilai bobot term silver pada dokumen 2 (**d2**) seperti yang ditunjukkan perhitungan dibawah ini :

$$w_{ij} = tf_{ij} \times \left(\log \left(\frac{D}{df_j} \right) + 1 \right)$$

$$w_{ij} = 2 \times \left(\log \left(\frac{3}{1} \right) + 1 \right)$$

$$w_{ij} = 2 \times (0,477 + 1)$$

$$w_{ij} = 2,954$$

Dengan demikian dapat diperoleh nilai **bobot (w)** untuk setiap term pada query dalam masing-masing dokumen seperti yang ditunjukkan pada tabel 3.2 dibawah ini :

Tabel 3.2. Simulasi Hasil Pembobotan Kata

Q	tf			D/df	IDF	IDF+1	W=tf * (IDF+1)			
	d1	d2	d3				d1	d2	d3	
gold	1	0	1	2	1.5	0.176	1.176	1.176	0	1.176
silver	0	2	0	1	3	0.477	1.477	0	2.954	0
truck	0	1	1	2	1.5	0.176	1.176	0	1.176	1.176
								sum(d1)	sum(d2)	sum(d3)
Nilai Bobot setiap Dokumen =								1.176	4.130	2.352

8. Klasifikasi Naive Bayes

Tahap klasifikasi Naive Bayes adalah tahapan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasikan data uji pada kategori yang paling tepat (Feldman & Sanger, 2007). Tahap klasifikasi Naive Bayes ini juga mengasumsikan penggolongan suatu keberadaan atau ketiadaan dari suatu kelas dengan fitur lainnya (Lorosae, et al. 2018). *Naive Bayes Classifier* (NBC) adalah metode classifier berdasarkan probabilitas dan Teorema Bayes dengan asumsi bahwa setiap variabel X bersifat bebas (independence). Dengan kata lain, *Naive Bayesian Classifier* (NBC) mengasumsikan bahwa keberadaan sebuah atribut (variabel) tidak ada kaitannya dengan keberadaan atribut (variabel) yang lain. Rumus *Naive Bayes Classifier* seperti yang ditunjukkan pada persamaan 6 dibawah ini

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \dots\dots\dots 6$$

9. Decision Tree

Decision Tree merupakan metode pencarian dengan alur dari atas ke bawah secara rekursif. Pada setiap tahapnya, atribut sebuah subset data training benar-benar dipertimbangkan, dan bentuk ini membelah menjadi 2 atau lebih subset

berdasarkan sebuah test bobot dari atribut. Ini akan berakhir hingga algoritma mencapai ranting yang diset hanya dapat menerima ketidaksetaraan. Sebuah bagian dari root ke sebuah ranting titik dicocokkan ke sebuah aturan keputusan konjungtive dan pohon keputusan merespon ke rule base ataupun ke sebuah setting aturan keputusan disjunctive. (Fong, et al. 2013).

10. K-Nearest Neighbor

K-NN adalah sebuah tipe pembelajaran berbasis suatu contoh atau pembelajaran sederhana dimana fungsinya hanya melakukan perkiraan pada seluruh maupun sebuah perhitungan dengan menunda perhitungannya hingga ada sebuah pengelompokan. Ini merupakan metode non parametric yang digunakan untuk pengelompokan ataupun untuk regresi.

Dalam hal klasifikasi, output dari klas anggota (paling banyak pada kelompok umum yang bisa dikembalikan), obyek diklasifikasikan menjadi sebuah pendapat mayoritas tetangganya, dengan sebuah obyek yang terdaftar pada suatu klas yang terbanyak diantara K nearest neighbor. Aturan ini memudahkan penguasaan atas data training selama proses pembelajaran dan didaftarkan pada masing-masing query suatu klas yang ditampilkan dengan mayoritas label k-nearest neighbor dalam proses training (Dey, et al. 2016).

11. Cross Validation

Cross Validation ini digunakan untuk menghitung akurasi, presisi, recall dan f-measure. Untuk kasus klasifikasi yang mempunyai proporsi jumlah data yang tidak seimbang akurasi tidak terlalu mencerminkan performa dari sistem yang dibuat. Perhitungan presisi, recall dan f-measure lebih penting untuk

mengetahui performa dari sistem yang dibuat. *Cross Validation* merupakan salah satu tools penting dalam metode visualisasi yang digunakan pada mesin pembelajaran yang biasanya memuat dua kategori atau lebih (Manning, et al. 2009; Horn, 2010). Tabel 3.2 menggambarkan contoh hasil *Cross Validation* prediksi dua kelas.

Tabel 3.3 Contoh Hasil *Cross Validation* Prediksi Dua Kelas

		Actual Class	
		Class-1	Class-2
Predicted Class	Class-1	True Positive	False Negative
	Class-2	False Positive	True Negative

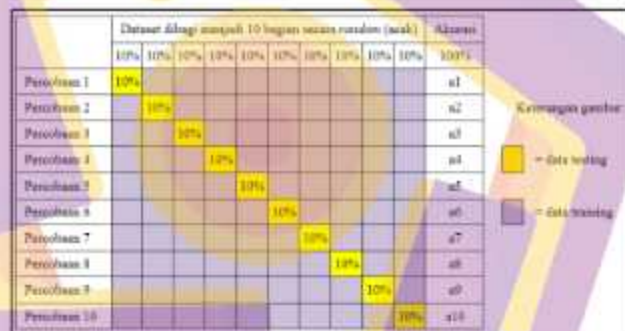
Nilai true positive (TP) dan true negative (TN) adalah hasil klasifikasi yang benar. Nilai false positive (FP) adalah nilai dimana hasilnya diprediksi sebagai class-1 namun sebenarnya merupakan class-2 sedangkan false negative (FN) adalah nilai dimana prediksi mengklasifikasikan sebagai class-2 namun faktanya termasuk dalam klasifikasi class-1. Nilai akurasi *Cross Validation* berdasarkan Tabel 3.1 diperoleh dengan persamaan 7.

$$Accuracy = \frac{TF + TN}{TF + TN + FP + FN} \dots \dots \dots 7$$

Validasi merupakan proses pengujian performa algoritma. Pada umumnya validasi dilakukan dengan mengulang proses perhitungan sampai beberapa kali. Proses validasi dalam penelitian ini menggunakan *cross validation*. *Cross*

validation adalah membagi dataset menjadi dua bagian dengan satu bagian dijadikan data training dan bagian yang lain dijadikan data testing.

Beberapa penelitian membagi data menjadi 10 bagian, 90% dijadikan training dan 10 lainnya digunakan sebagai testing. Proses ini dilakukan berulang sampai dengan 10 kali hingga semua record data mendapatkan bagian menjadi data testing. Proses ini dikenal juga dengan istilah 10 folds cross validation. 10 folds cross validation banyak digunakan peneliti karena terbukti menghasilkan performa algoritma yang lebih stabil. Gambar 3.9 merupakan representasi dari 10 folds cross validation.



Gambar 3.9. Representasi 10 folds cross validation

12. Confusion Matrix

Pengukuran akurasi algoritma Pengukuran akurasi merupakan tahapan untuk membuktikan tingkat performa suatu algoritma terhadap dataset yang digunakan. Dalam penelitian ini digunakan confusion matrix sebagai alat ukur performa algoritma klasifikasi. Confussion matrix atau matrik kebingungan merupakan sebuah perhitungan yang membandingkan dataset dengan hasil klasifikasi sesuai dengan data sebenarnya dengan jumlah keseluruhan data.

Hasil akhir dari matrik ini adalah tingkat akurasi dengan satuan persen (%). Tingkat akurasi ini yang nantinya dijadikan acuan para peneliti terkait performa algoritma klasifikasi tersebut. Confusion Matrix adalah evaluasi dari sebuah klasifikasi data mining yang direpresentasikan menjadi tabel (Gorunescu 2011). Confusion matrix berisi informasi perbandingan label hasil klasifikasi dengan label sebenarnya. Tabel 3.4 menggambarkan confusion matrix dengan dua label yaitu yes dan no.

Tabel 3.4. Confusion Matrix (Gorunescu 2011)

Classification		Predicted class	
		Class: YES	Class: NO
Observed class	Class YES	<i>a</i> True Positive (TP)	<i>b</i> False Negative (FN)
	Class NO	<i>c</i> False Positive (FP)	<i>d</i> True Negative (TN)

Dari tabel 3.4 dapat dihitung tingkat akurasi dari sebuah model algoritma dengan menggunakan persamaan sebagai berikut:

$$Accuracy = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Keterangan:

- a : hasil klasifikasi positif dengan klas sebenarnya positif
- b : hasil klasifikasi negatif dengan klas sebenarnya positif
- c : hasil klasifikasi positif dengan klas sebenarnya negatif
- d : hasil klasifikasi negatif dengan klas sebenarnya negatif

Dalam aplikasi rapid miner proses perhitungan performa algoritma dapat dijelaskan sesuai gambar 3.10 dibawah ini :



Gambar 3.10. Confussion matrix model dalam aplikasi rapid miner

Dalam data training digunakan algoritma KNN dan dalam testing digunakan apply model serta prformance untuk menghasilkan matrik kebingungan atau confussion matrix. Hasil matrik dari aplikasi dapat dilihat pada gambar 3.11 dibawah ini :

Accuracy / AAK = 84,6% (AAK) (AAK)			
	Real kelas positif	Real kelas negatif	Total prediksi
pred kelas positif	78	78	156 (84,6%)
pred kelas negatif	71	427	498 (84,6%)
Total	149	505	654

Gambar 3.11. Hasil akurasi dari Confussion Matrix

13. Algoritma Bat

Dalam Algoritma Bat, secara umum frekuensi f dalam rentang $[f_{min}, f_{max}]$ sesuai dengan kisaran panjang gelombang $[\lambda_{min}, \lambda_{max}]$. Misalnya rentang frekuensi $[20\text{kHz}, 500\text{kHz}]$ sesuai dengan kisaran panjang gelombang dari 0,7 mm hingga 17 mm. Untuk masalah tertentu, kita juga bisa menggunakan panjang gelombang apa saja untuk kemudahan implementasi. Dalam implementasi aktual, kita dapat menyesuaikan rentang dengan menyesuaikan panjang gelombang (atau frekuensi), dan rentang yang dapat dideteksi (atau panjang gelombang terbesar) harus dipilih sedemikian rupa sehingga sebanding dengan ukuran domain yang diminati, dan kemudian dikurangi ke rentang yang lebih kecil.

Selain itu, kita tidak harus menggunakan panjang gelombang sendiri, sebagai gantinya, kita juga dapat memvariasikan frekuensi sambil memperbaiki panjang gelombang λ . Ini adalah karena λ dan f terkait karena fakta λf adalah

konstan. Kami akan menggunakan ini nanti pendekatan dalam implementasi kami. Untuk kesederhanaan, kita dapat mengasumsikan $f \in [0, f_{\max}]$. Kita tahu bahwa frekuensi lebih tinggi panjang gelombang pendek dan menempuh jarak yang lebih pendek.

Untuk kelelawar, rentang tipenya beberapa meter. Laju denyut nadi bisa berada dalam kisaran $[0, 1]$ di mana 0 berarti tidak ada pulsa sama sekali, dan 1 berarti tingkat maksimum emisi pulsa. Berdasarkan perkiraan dan idealisasi ini, langkah dasar dari Algoritma Bat (BA) dapat diringkas sebagai kode pseudo yang ditunjukkan pada Pseudocode Algoritma Bat dibawah ini :

Kondisi awal:

Fungsi obyektif (x) , $x = (x^1 \dots x^n)^t$.

Inisialisasi populasi kelelawar x_i dan v_i , $i = 1, 2 \dots n$.

Tentukan frekuensi pulsa f_i pada x_i , $\forall i = 1, 2 \dots n$.

Inisialisasi denyut nadi r_i dan kenyaringan A_i , $i = 1, 2 \dots n$.

While $t < T$

For each bat x_i , do

Hasilkan solusi baru menggunakan Persamaan (1), (2) dan (3).

If $r \text{ and } > r_i$, maka

Pilih solusi di antara solusi terbaik.

Hasilkan solusi lokal di sekitar solusi terbaik.

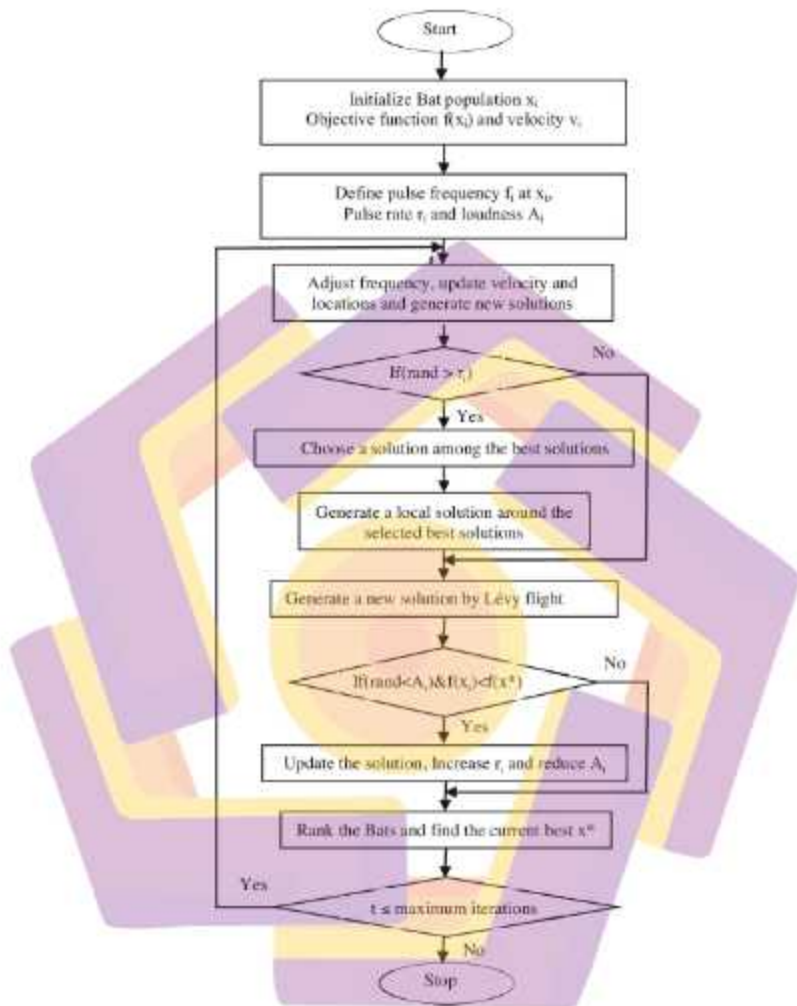
If $r \text{ and } < A_i$ dan $f(x_i) < (x^*)$, maka

Terima solusi baru.

Tambah r_i dan kurangi A_i .

Beri peringkat kelelawar dan temukan yang terbaik saat ini x^* .

Berdasarkan pseudocode diatas, berikut ini Flowchart Bat Algoritma seperti yang digambarkan pada gambar 3.12 dibawah ini :



Gambar 3.12. Flowchart Algoritma Bat

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Crawling Data Tweet (Twitter search API)

Pada tahap ini dilakukan pengujian pengambilan data dengan kata kunci pilkada dengan menggunakan tools RapidMiner yang telah terhubung dengan akun twitter. Berikut ini contoh hasil *crawling* pada RapidMiner ditunjukkan pada tabel 4.1 dibawah ini :

Tabel 4.1. Hasil Crawling data Twitter dari RapidMiner

ID	Text	Created At	Retweet Count	Reply Count	Like Count	Source
1234567890123456789	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:30:00	12	5	30	Twitter
1234567890123456790	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:31:00	10	4	28	Twitter
1234567890123456791	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:32:00	15	6	35	Twitter
1234567890123456792	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:33:00	11	5	29	Twitter
1234567890123456793	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:34:00	13	6	32	Twitter
1234567890123456794	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:35:00	14	7	33	Twitter
1234567890123456795	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:36:00	16	8	36	Twitter
1234567890123456796	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:37:00	17	9	37	Twitter
1234567890123456797	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:38:00	18	10	38	Twitter
1234567890123456798	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:39:00	19	11	39	Twitter
1234567890123456799	Siapa yang bakal menang di Pilkada? #Pilkada	2023-05-15 10:40:00	20	12	40	Twitter

Untuk diketahui bahwa crawling twitter ini hanya dapat menarik data selama 1 minggu kebelakang sehingga apabila ingin menarik data lebih banyak secara berkala maka proses crawling dapat dilakukan secara harian. Dalam pengujian ini data yang telah dicrawling disimpan kedalam file excel.

4.2 Analisis Sumber Data

Pada tahap ini akan dilakukan penghapusan data kembar dan pemilahan kolom data yang relevan dengan kebutuhan penelitian ini. Tahap ini berguna untuk memastikan bahwa data yang akan kita olah merupakan data yang sudah

relevan dengan studi kasus yang diambil dalam penelitian ini yaitu sentimen analisis terhadap pemilihan kepala daerah.

1. Penghapusan data kembar dan pemilahan data yang relevan

Tahap pengambilan data yang relevan disini dimaksudkan adalah membuang data hasil crawling yang memiliki kesamaan kalimat dan kemudian memilah kolom apa saja yang dibutuhkan pada penelitian ini. Pada tahap ini digunakan komponen operator *Remove Duplicates* dan *Select Attributes*.

2. Cleansing

Proses ini merupakan tahap penghapusan karakter-karakter yang mengiringi kata-kata tersebut yang hilangnya karakter-karakter tersebut tidak mengurangi makna dari penelitian ini. Contoh dari karakter-karakter yang ikut terbuang dari penelitian ini adalah tanda kurung “(”, petik dua “”, url, #, titik dua “:”, tanda titik “.”, tanda “@”, dan lain-lain. Untuk itu ditambahkan operator sub proses setelah komponen select attribut.

2. Labeling

Tahap labeling ini adalah tahap pemberian klasifikasi kalimat positif, negative dan netral. Pada tahap ini dilakukan pemberian label positif, negatif dan netral ini baik secara otomatis maupun secara manual. Pemberian label secara otomatis dilakukan oleh operator dari pihak ketiga yaitu *Rosette Text Analytics*. Sedangkan secara manual juga masih dilakukan untuk mencegah kesalahan otomatisasi pelabelan terkait atas makna bahasa yang digunakan serta memastikan jumlah bentuk sentimen sesuai dengan skenario penelitian.

Penentuan label ini ditunjukkan dengan inisial pos untuk positif, neu untuk netral dan neg untuk negatif. Selanjutnya hasil rangkaian proses crawling hingga tersimpan di file excel ini ditunjukkan pada tabel 4.2 dibawah ini :

Tabel 4.2. Hasil proses Crawling, cleaning dan labeling

ID	A	B	C	D
ID	Input	Text	Sentiment	
1	129976256818511105	di lapangan setelah dulu pernah setjembang endre bawanya salah itu juga wakil wali kota	neg	
2	130012688413479049	perlu itu hal yang baik tapi menurut persahabatan karena membuat masyarakat	neg	
3	130018136089037572	betonkran puluhan pembukuk bakal calon bupati tuchmanu jakur penangan rusak	neg	
4	130018136089037572	pengawat politik calon bupati prihadi kakri kemenduran demokrasi portal berita jawa tv neg	neg	
5	130018136089037572	pribadi di riau mendukung calon terimakasih kepada postandiburak demokrasi demokrasi neg	neg	
6	130018136089037572	formensi atau politik uang dalam wajan baru prihadi 2020	neg	
7	130018136089037572	pengembangan jumlah penerima bantuan sosial berbasis laporan hhd terhadap belanja neg	neg	
8	130018136089037572	sumbana dipanggil siapa sudah masuk prihadi wali walinya anakku gblah mau maju	neg	
9	130018136089037572	lembah in baru lalu kempung	neg	
10	130018136089037572	partai jomb tak hanya setting jadi jumlah senilai tapi partai pnsing makna nuamahnya	neg	
11	130018136089037572	namun kaitan para pencungng kalen prihadi melampirkan ke anek	neg	
12	130018136089037572	bayanganmua mudal dan lawan kotak kosong saat prihadi berharap yang menang kotak ke neg	neg	
13	130018136089037572	jakribat hari 2024 bisa pla yg menang pemilu yang rusak bisa saja maku rian dan pak neg	neg	
14	130018136089037572	1300 itu dia yg bergerak ekonomi duti mengaji ke masyarakat kandidat beromba neg	neg	
15	130018136089037572	lingkar ruginanya tantangan prihadi prihadi yg di tolak ukurnya bisa ugk mau neg	neg	
16	130018136089037572	atijaham anakkolong bumbersato hata umat masa dan dia calon bupati bakasi aja kala neg	neg	
17	130018136089037572	abuarata smpt deonytong? data saya rdn saya diukasi sdi dukung kapusan indogendering	neg	
18	130018136089037572	afatih ddi apatgi kalau mendekati pemilu ada surat cinta pnsig nyukah jga seleai pemilu neg	neg	
19	130018136089037572	agakamua fatmawati juga pernah bakal calon walikota pangsip 2013 gagal jadi calon kare neg	neg	
20	130018136089037572	apenasngatan harnafakhan yg ngarang sumernya selain rdn kermas yg nyalin prihadi di neg	neg	
21	130018136089037572	apenasngatan penerminat bagaimana tanggapan bapak apenasngatan untuk prihadi di neg	neg	
22	130018136089037572	akurati bagbag kue kemenangan pemilu 2023 tapi sayang kue tersebut pada bersemb tak neg	neg	
23	130018136089037572	afnyandatan lansana hayo mngga sila ni calon walikota anda jangan lupa pilih har lita s neg	neg	
24	130018136089037572	afnyandatan lansana lita atitudema@mana itu calon walikota paka meker model gtu neg	neg	
25	130018136089037572	afnyandatan masih banyak yg pekat tua calon walikota ngapain juga milih yg gk bermutu neg	neg	
26	130018136089037572	amthika dawatibapak2 maly10102 prihadi di ama prihadi walikota levelnya beda drun neg	neg	

4.3. PreProcessing Data

Dalam tahap ini dilakukan proses case folding, tokenizing, stop word removal, stemming. Proses ini berguna untuk membantu menghasilkan data yang berkualitas untuk dapat diklasifikasikan ke dalam 3 algoritma klasifikasi nantinya. Tahap ini juga menjadi bagian dari kelompok tahap pengoptimasian data algoritma klasifikasi dengan menggunakan algoritma bat. Adapun langkah-langkah pada tahap ini adalah :

1. Case Folding

Proses yang dilakukan pada tahap ini adalah merubah huruf kapital menjadi huruf kecil. Proses case folding ini memastikan bahwa format huruf yang akan diklasifikasikan tidak memiliki perbedaan. Hal ini membantu kualitas

pengklasifikasian kata yang digunakan. Implementasi case folding ini dilakukan setelah data berbentuk excel. Selanjutnya diberikan rumus '=lower()' pada kolom text untuk merubahnya menjadi huruf kecil.

3. Tokenizing

Proses ini memisahkan kata-kata yang telah dibersihkan dari karakter dan kata-kata yang tidak berhubungan dengan penelitian ini menjadi kata-kata yang tersendiri. Dalam tahap ini proses diawali dari proses import data dari excel ke dalam aplikasi RapidMiner. Hasil dari proses ini kemudian dipindahkan kedalam papan proses RapidMiner dengan nama *Retrieve Pilkada*.

4. Stop Words Removal

Pada proses ini dilakukan penghapusan atas kata-kata yang ketiadaan fungsinya tidak membuat berkurangnya makna & arti dari penelitian yang dilakukan. Contoh kata-kata yang dibuang dalam penelitian ini adalah yang, maka, itu, daripada, begitu, ini, dan lain-lain. Pada penelitian ini filter stop words dilakukan dengan menggunakan file kamus *Stopwords* yang berisi kata-kata apa saja yang harus dihapus dan *path* file kamus ini dipasang pada setting opsi operator *Filter Stopwords(Dictionary)*.

5. Stemming

Proses ini adalah sebuah proses untuk merubah kata yang awalnya memiliki imbuhan awalan, akhiran bahkan sisipan menjadi hanya kata dasarnya. Pada penelitian ini proses stemming juga digunakan untuk membersihkan kata-kata gaul/*slang words*.

pakai	kepala	pilkada	jabat	demo	potensi
-------	--------	---------	-------	------	---------

6. Term Weighting TF-IDF / Pembobotan Kata

Dari hasil dari tabel hasil tokenizing, stopwords removal dan stemming. Pada penelitian ini proses Pembobotan Kata disusun dalam tahap sebagai berikut :

1. Menyiapkan data hasil proses tokenizing, stopwords removal dan stemming.
 - a. Dokumen awal

Data ini merupakan salah satu dari hasil crawling pada penelitian ini. Data ini telah dilakukan proses case folding, stop words removal dan stemming seperti yang ditunjukkan pada tabel 4.4 dibawah ini :

Tabel 4.4. Contoh Dokumen awal

no	id	Dokumen
1	1283363798586322944	realita tujuan pilkada menang ajang potensi kader cemerlang
2	1283363686430543873	mari kawal pilkada sama hasil proses demokrasi sehat kualitas integritas
3	1283363679795380225	dinamika politik jelang pilih kepala daerah tensi politik mulai hangat
4	1284066020726984705	kasihan pecundang kalah pilkada lampias anies

b. Data Tokenizing

Langkah selanjutnya adalah melakukan tokenize pada dokumen sehingga menjadi kata yang berdiri sendiri. Berikut ini tabel contoh data tokenize yang ditunjukkan pada tabel 4.5 dibawah ini :

Tabel 4.5. Contoh Data Tokenizing

text									
1	2	3	4	5	6	7	8	9	10
realita	tujuan	pilkada	menang	ajang	potensi	kader	cemerlang		
marit	kawaf	pilkada	sama	hasil	proses	demokrasi	sehat	kuaitas	integritas
dinamika	politik	jelang	pilih	kepala	daerah	tensi	politik	mulai	hangat
kasihan	pecundang	kalah	pilkada	lampias	antes				

2. Menghitung Term Weighting TF-IDF / Pembobotan Kata

Setelah melakukan proses tokenizing pada dokumen kemudian lakukan proses perhitungan pembobotan kata dengan langkah-langkah sebagai berikut:

- Lakukan perhitungan jumlah frekuensi kata (misalkan kata pilkada) pada dokumen 1, 2 dan 3 sehingga didapatkan nilai d yaitu 1,1,0
- Lakukan penjumlahan pada frekuensi kata selain 0 untuk mendapatkan df
- Lakukan pembagian D dibagi dengan df
- Hitung nilai IDF dengan menghitung \log_{10} pada hasil pembagian D/df
- Tambahkan angka 1 pada hasil IDF
- Lakukan proses nilai W dengan cara mengkalikan nilai tf atau D dengan $IDF + 1$
- Jumlahkan W pada masing-masing dokumen
- Hasil dari proses Pembobotan kata ini dapat dilihat pada tabel 4.6

Tabel 4.6. Simulasi perhitungan Pembobotan Kata

Q	D				df	D/df	IDF	IDF+1	W=tf * (IDF+1)			
	1	2	3	4					d1	d2	d3	d4
pilkada	1	1	0	1	3	1,333	0,125	1,125	1,125	1,125	-	1,125
jabat	0	0	0	0	0	-	-	1,000	-	-	-	-
kawal	0	1	0	0	1	4,000	0,602	1,602	-	1,125	-	-
kepala	0	0	1	0	1	4,000	0,602	1,602	-	-	1,125	-
daerah	0	0	1	0	1	4,000	0,602	1,602	-	-	1,125	-
kader	1	0	0	1	4,000	0,602	1,602	1,125	-	-	-	-
demokras	0	1	0	0	1	4,000	0,602	1,602	-	1,125	-	-
proses	0	1	0	0	1	4,000	0,602	1,602	-	1,125	-	-
politik	0	0	2	0	1	4,000	0,602	1,602	-	-	2,250	-
Total Nilai Bobot Kata setiap Dokumen =									2,250	4,500	4,500	1,125

4.4 Klasifikasi dan Akurasi

Dalam melakukan proses Klasifikasi dan Akurasi ini, kita akan membagi menjadi 2 skenario percobaan. Skenario pertama adalah membuat suatu klasifikasi dan akurasi atas hasil preprocessing tanpa optimasi. Skenario kedua adalah membuat suatu klasifikasi dan akurasi setelah dilakukan suatu optimasi dengan algoritma Bat. Berikut ini hasil dari 2 skenario tersebut.

4.4.1. Skenario Pertama

Pada skenario pertama ini, kita akan melakukan pengklasifikasian data menggunakan Algoritma Naive Bayes, Decision Tree dan K-NN tanpa menggunakan proses optimasi.

1. Klasifikasi pada Naive Bayes

Pada proses pengklasifikasian text mining menggunakan Algoritma Naive Bayes ini dibutuhkan langkah-langkah sebagai berikut :

- a. Persiapan dokumen yang telah berlabel (Positif, Negatif, Neutral)

Pada langkah pertama ini disiapkan data dokumen yang telah berlabel sentiment yang sudah dilakukan pada proses awal penelitian ini sebagaimana yang ditunjukkan pada tabel 4.7 dibawah ini :

Tabel 4.7. Contoh dokumen yang akan diklasifikasi

no	Id	Dokumen	Sentimen
1	1283363798586322944	realita tujuan pilkada menang ajang potensi kader cemerlang	Positif
2	1283363686430543873	mari kawal pilkada sama hasil proses demokrasi sehat kualitas integritas	Neutral
3	1283363679795380225	dinamika politik jejang pilih kepala daerah tensi politik mulai hangat	Neutral
4	1284066020726984705	kasihan pecundang kalah pilkada lampias keras	Negatif

b. Rekap tabel berlabel dengan nilai pembobotan kata

Langkah selanjutnya adalah melakukan rekap data training. Pada rekap ini dikumpulkan data Kata, Dokumen, Bobot serta Sentimen itu sendiri. Berikut ini tampilan tabel training data yang ditunjukkan pada tabel 4.8 dibawah ini

Tabel 4.8. Contoh tabel Training Data

Kata	Dokumen	Bobot	Sentimen
pilkada	D1	1	POSITIF
pilkada	D2	1	NEUTRAL
pilkada	D4	1	NEGATIF
kawal	D2	1	NEUTRAL
kepala	D3	1	NEUTRAL
daerah	D3	1	NEUTRAL
kader	D1	1	POSITIF
demokrasi	D2	1	NEUTRAL
proses	D2	1	NEUTRAL
politik	D3	2	NEUTRAL

c. Hitung persentase tiap kolom klasifikasi (Naïve Bayes Model)

Langkah selanjutnya adalah menghitung prediksi dari masing-masing kolom. Langkah ini menghitung persentase peluang tiap klasifikasi terhadap keseluruhan klasifikasi pada tiap peluang sentimen. Misalkan pada sentimen positif yang hanya ada 2 peluang dari 10 peluang maka persentasenya adalah 20%. Jika dalam excel, rumus yang digunakan menggunakan COUNTIF. Untuk itu hasil dari seluruh persentase ditunjukkan pada tabel 4.9 dibawah ini :

Tabel 4.9. Naïve Bayes Model

Sentimen	POSITIF	NEGATIF	NEUTRAL
P(POSITIF/NEGATIF/NEUTRAL)	20%	10%	70%
TOTAL	100%		
Kata	POSITIF	NEGATIF	NEUTRAL
pilkada	50%	100%	14%
jabat	0%	0%	0%
kawal	0%	0%	14%
kepala	0%	0%	14%
daerah	0%	0%	14%
kader	50%	0%	0%
demokrasi	0%	0%	14%
proses	0%	0%	14%
politik	0%	0%	14%
TOTAL	100%	100%	100%
Dokumen	POSITIF	NEGATIF	NEUTRAL
D1	100%	0%	0%
D2	0%	0%	57%
D3	0%	0%	43%
D4	0%	100%	0%
TOTAL	100%	100%	100%
Bobot	POSITIF	NEGATIF	NEUTRAL
1	100%	100%	86%
2	0%	0%	14%
TOTAL	100%	100%	100%

Dengan total tiap klasifikasi 100% maka ini menunjukkan tidak ada kesalahan pada perhitungan rumusnya.

d. Testing Data Model

Langkah selanjutnya adalah menyusun tabel data uji/Testing Data. Tabel uji ini berisi data yang hampir sama dengan data latih/Training Data.

Berikut ini contoh dari tabel data uji/Testing Data yang ditunjukkan pada tabel 4.10 dibawah ini :

Tabel 4.10. Training Data Model

Testing data	Kata	Dokumen	Bobot	Sentimen
	kader	D1	1	POSITIF
	kawal	D2	1	NEUTRAL
	proses	D2	1	NEGATIF
	pilkada	D2	1	NEUTRAL
	pilkada	D4	1	NEGATIF

e. Prediction Model

Setelah menyusun Testing Data maka langkah selanjutnya adalah menghitung Class Prediction Model. Langkah ini menghitung peluang tiap sentiment dari data uji/Testing Data terhadap data latih/Training Data. Dalam rumus excel ini digunakan rumus VLOOKUP. Untuk Class Prediction didapatkan dari perbandingan nilai antar ke-3 Class Prediction. Misalkan pada baris-1 karena nilai positif ternyata lebih mendominasi dari nilai negatif dan neutral maka hasil Class Predictionnya menyatakan hasil positif. Dalam rumus excel tabel Class Prediction ini menggunakan rumus IF. Hasil dari Prediction Model ini ditunjukkan pada tabel 4.11 dibawah ini :

Tabel 4.11. Prediction Model

Prediction	CLASS PREDICTION	POSITIF	NEGATIF	NEUTRAL
	POSITIF	10,0%	0,0%	0,0%
	NEUTRAL	0,0%	0,0%	4,9%
	NEUTRAL	0,0%	0,0%	4,9%
	NEUTRAL	0,0%	0,0%	4,9%
NEUTRAL	0,0%	10,0%	0,0%	

f. Confusion Model

Untuk menghitung nilai akurasi dari algoritma Klasifikasi Naïve Bayes dalam RapidMiner menggunakan Cross Validation yang berisi perhitungan dengan teknik Confusion Matrik. Confusion Matrik ini menghitung jumlah sentiment pada data uji dibandingkan dengan jumlah sentiment pada data prediksi. Dalam rumus excel ini menggunakan COUNTIF. Hasil dari Confusion matrik ini ditunjukkan pada tabel 4.12 dibawah ini :

Tabel 4.12. Confusion Matrix Tabel

		CLASS			
		POSITIF	NEGATIF	NEUTRAL	
Confusion Table	Predicted ↓				
	POSITIF	1	0	0	
	NEGATIF	0	1	0	
		NEUTRAL	0	1	2
Accuracy =		80%			

g. Hitung Akurasi

Tahap akhir dari Algoritma klasifikasi ini adalah menghitung akurasi dari hasil Class Prediction dimana jumlah tiap prediksi sentiment dibagi dengan jumlah keseluruhan hasil Class Prediction. Dalam simulasi ini didapatkan hasil akurasinya yaitu 80 % seperti yang ditunjukkan pada tabel 4.12 diatas. Hasil akurasi 80 % ini maksudnya adalah ada hasil

sentimen dari tabel data uji/Testing Data yang tidak sesuai/tidak tercantum pada tabel data latih/Training Data.

Hal ini dapat dilihat pada baris ke-3 pada tabel Uji dimana hasilnya adalah Negatif sedangkan hasil Negatif tersebut belum pernah ada dalam Tabel Data Latih/Training Data. Hal ini berakibat hasil akurasi yang didapatkan menjadi hanya 80 % karena data prediksinya menjadi meleset karena adanya ketidaksamaan data antara data Uji dan data Latih.

h. Hasil RapidMiner

Untuk akurasi pada penelitian ini, diimplementasikan menggunakan aplikasi RapidMiner dimana dalam penelitian ini menggunakan operator *Cross Validation* untuk menampilkan perhitungan hasil akurasi pada Algoritma Klasifikasi Naïve Bayes. Pada operator *Cross Validation* ini didalamnya dibagi menjadi 2 layar proses yaitu proses Algoritma klasifikasi dan proses Akurasi seperti yang ditunjukkan pada gambar 4.2 dibawah ini :



Gambar 4.2 klasifikasi Naive Bayes dan akurasinya

Pada proses layar training digunakan operator Naive Bayes, K-NN dan Decision Tree yang dihubungkan dengan operator *Apply Model* dan *Performance*. Hasil akurasi dari Naive Bayes senilai 33,58 % ditunjukkan pada gambar 4.3 dibawah ini :

accuracy: 33.56% +/- 1.19% (micro average: 33.56%)

	true pos	true neu	true neg	class precision
pred. pos	1575	1600	1594	33.03%
pred. neu	1462	1458	1445	33.40%
pred. neg	1419	1374	1403	34.38%
data recall	38.35%	32.80%	32.50%	

Gambar 4.3 Hasil akurasi dari Naive Bayes

2. Klasifikasi pada K-NN

Pada proses pengklasifikasian text mining menggunakan Algoritma K-NN ini dibutuhkan langkah-langkah sebagai berikut :

- a. Persiapan dokumen yang telah disimbolkan dengan angka

Agar data klasifikasi pada Algoritma KNN ini dapat dihitung menggunakan rumus Euclidian Distance maka seluruh data pada data latih/Training Data dan data uji/Testing Data akan disimbolkan dalam bentuk angka. *Euclidian Distance adalah formula untuk mencari jarak antara 2 titik dalam ruang dua dimensi.* Berikut ini hasil konversi data string menjadi data numerik untuk dokumen, kata dan sentiment seperti yang ditunjukkan pada tabel 4.13 dibawah ini :

Tabel 4.13. Konversi Data String ke Numerik

Kata	Bobot
pilkada	1
jabat	2
kawal	3
kepala	4
daerah	3
kader	6
demokrasi	7
proses	8
politik	9

Sentimen	Bobot
NEGATIF	1
NEUTRAL	2
POSITIF	3

Dokumen	Bobot
D1	1
D2	2
D3	3
D4	4

b. Persiapkan data uji/Testing Data

Langkah selanjutnya adalah menyiapkan data Uji/Testing Data. Dalam simulasi ini kami menyiapkan 1 contoh data uji/Testing Data dimana nantinya data tersebut akan dihitung jarak antara data Uji dengan data Latih/Training Data. Berikut ini contoh data uji yang ditunjukkan pada tabel 4.14 dibawah ini :

Tabel 4.14. Contoh Data Testing/Uji Data

	Kata	Dokumen	TF	Sentimen
Testing Data	pilkada	D1	1	POSITIF
Bobot	1	1	1	3

c. Persiapkan Data Latih/Training Data

Untuk memastikan bahwa Algoritma K-NN ini dapat berjalan dengan benar maka perlu disiapkan tabel Data Latih/Training Data. Data Latih/Training Data ini telah dikonversi ke dalam bentuk numerik

sebagaimana data latih/Training Data. Berikut ini tabel data uji seperti yang ditunjukkan pada tabel 4.15 dibawah ini :

Tabel 4.15. Contoh Data Latih/Training Data

Training Data	Kata	Dokumen	TF	Sentimen
	8	1	1	3
	3	2	1	2
	8	2	1	1
	1	2	1	2
	9	4	1	1
	5	2	1	3
	2	4	1	1
	3	2	1	2
	4	3	1	2
	5	3	1	2
	6	1	1	3
	7	2	1	2
	8	2	1	1
	9	3	2	2
6	3	2	2	

d. Hitung Nilai Ecludian Distance / K-NN Model

Pada tahap selanjutnya, dilakukan perhitungan nilai Jarak Euclidean antara data latih dan data uji. Dengan menggunakan rumus $d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$ pada tahap ini dihitung jarak terdekat dari data uji/Testing Data ke data latih/Traming Data. Setelah ditemukan nilai jaraknya, kemudian dilakukan perhitungan nilai terkecil dari data jarak yang sudah dibuat apakah sesuai dengan nilai sentiment Negatif(1), Netral(2), Positif(3). Pengujian ini dilakukan hingga K=15. Hasil dari tahap K-NN Model ini ditunjukkan pada tabel 4.16 dibawah ini :

Tabel 4.16. Contoh K-NN Model

K-NN Model	Distance	K = 1	K = 5	K = 9	K = 13	K = 15
	7,000	0	0	0	3	3
	2,449	0	2	2	2	2
	7,348	0	0	0	1	1
	1,414	2	2	2	2	2
	8,775	0	0	0	0	1
	4,123	0	0	3	3	3
	3,742	0	1	1	1	1
	2,449	0	2	2	2	2
	3,742	0	2	2	2	2
	4,583	0	0	2	2	2
	5,000	0	0	3	3	3
	6,164	0	0	0	2	2
7,348	0	0	0	1	1	
8,367	0	0	0	0	2	
5,568	0	0	2	2	2	

e. Hitung Nilai Persentase Sentimen Data Uji/Testing Data

Proses ini dilakukan sebelum menghitung K-NN Model Prediction.

Hal ini dilakukan sebagai langkah awal untuk menghitung akurasi menggunakan tabel Confusion Matrix. Adapun tabel prediksi

Sentimen data uji ditunjukkan pada tabel 4.17 dibawah ini :

Tabel 4.17. Persentase Sentimen Data Uji/Testing Data

	NEGATIF	NEUTRAL	POSITIF
	1	2	3
P(NEGATIF/NEUTRAL/POSITIF)	27%	53%	20%
TOTAL	100%		

f. Hitung Nilai K-NN Model Prediction

Langkah selanjutnya adalah menghitung persentase perbandingan jarak distance dari total seluruh distance. Kemudian hasil persentase tersebut dibandingkan kepada masing-masing persentase masing-

masing sentimen. Nilai yang paling mendekati masing-masing sentiment merupakan hasil dari Class Prediction. Hasil dari K-NN Model Prediction ditunjukkan pada tabel 4.18 dibawah ini :

Tabel 4.18. K-NN Model Prediction

Distance	PERSENTASE			CLASS PREDICTION	
	JARAK	NEGATIF 1	NEUTRAL 2		POSITIF 3
7,000	47%	20%	0%	27%	NEUTRAL
2,449	16%	0%	0%	0%	POSITIF
7,348	49%	22%	0%	29%	NEUTRAL
1,414	9%	0%	0%	0%	POSITIF
8,775	58%	32%	5%	38%	NEUTRAL
4,123	27%	1%	0%	7%	NEUTRAL
3,742	25%	0%	0%	5%	POSITIF
2,449	16%	0%	0%	0%	POSITIF
3,742	25%	0%	0%	5%	POSITIF
4,583	31%	4%	0%	11%	NEUTRAL
5,000	33%	7%	0%	13%	NEUTRAL
6,164	41%	14%	0%	21%	NEUTRAL
7,348	49%	22%	0%	29%	NEUTRAL
8,367	56%	29%	2%	36%	NEUTRAL
5,568	37%	10%	0%	17%	NEUTRAL

g. Hitung Nilai Confusion Matrix dan Akurasi

Hasil akhir dari Algoritma K-NN ini adalah menghitung nilai akurasi menggunakan Confusion Matrix. Pada tahap ini dilakukan perhitungan perbandingan nilai sentiment pada data Uji dibandingkan dengan nilai sentiment pada data K-NN Model Prediction. Hasil dari nilai ini kemudian dihitung nilai akurasinya yang menghasilkan nilai 27%. Hasil perhitungan nilai Confusion Matrix dan Akurasi ini dapat dilihat pada tabel 4.19 dibawah ini :

Tabel 4.19. Confusion Matrix dan Akurasi

Confusion Table	Predicted	CLASS		
		NEGATIF	NEUTRAL	POSITIF
	1	2	3	
NEGATIF	0	0	0	
NEUTRAL	3	4	3	
POSITIF	1	4	0	
Accuracy =		27%		

h. Hasil RapidMiner

Untuk akurasi pada penelitian ini, diimplementasikan menggunakan aplikasi RapidMiner dimana dalam penelitian ini. Desain dari operator Algoritma KNN ditunjukkan pada gambar 4.4 dibawah ini :



Gambar 4.4. Layar proses Training dan Testing dengan klasifikasi K-NN

Pada penelitian ini hasil akurasi dari K-NN didapatkan nilai 32,82 % ditunjukkan pada gambar 4.5 dibawah ini :

accuracy: 32.82% +/- 1.73% (micro average: 32.82%)

	true pos	true neu	true neg	class precision
pred pos	1482	1438	1501	32.89%
pred neu	1452	1473	1511	33.36%
pred neg	1562	1521	1460	32.14%
class recall	32.81%	33.24%	32.47%	

Gambar 4.5. Hasil akurasi dari K-NN

Hasil dari performance vektor dari K-NN ditunjukkan pada gambar 4.6 dibawah ini :

PerformanceVector

```

PerformanceVector:
accuracy: 53.23% +/- 6.12% (micro average: 53.23%)
ConfusionMatrix:
True:  neg   neu   pos
neg:  106   0    50
neu:   0   99   66
pos:   44   50   34
kappa: 0.298 +/- 0.092 (micro average: 0.299)
ConfusionMatrix:
True:  neg   neu   pos
neg:  106   0    50
neu:   0   99   66
pos:   44   50   34

```

Gambar 4.6 Hasil performance vektor dari K-NN

3. Klasifikasi pada Decision Tree

Pada proses pengklasifikasian text mining menggunakan Algoritma Decision Tree ini dibutuhkan langkah-langkah sebagai berikut :

a. Persiapan Data Latih/Training Data

Pada tahap awal implementasi Decision Tree adalah mempersiapkan data latih/Training Data. Pada data latih ini disiapkan sebanyak 13 record data dan 3 atribut yaitu Kata, Dokumen, TF (Pembobotan Kata) serta 1 label yaitu Sentimen. Berikut ini tabel data latih/Training Data yang ditunjukkan pada tabel 4.20 dibawah ini :

Tabel 4.20. Data Latih Decision Tree

Training data	Kata	Dokumen	TF	Sentimen
	pilkada	D1	1	POSITIF
	kader	D2	2	NEUTRAL
	jabat	D4	1	NEGATIF
	kawal	D2	1	NEUTRAL
	kepala	D4	2	NEGATIF
	daerah	D3	1	NEUTRAL
	jabat	D1	2	POSITIF
	demokras	D2	1	NEUTRAL
	proses	D2	1	POSITIF
	politik	D3	2	NEUTRAL
	jabat	D1	1	NEUTRAL
	kawal	D2	1	POSITIF
	daerah	D3	2	NEUTRAL

b. Perhitungan Data Uji/Testing Data

Dalam data uji pada Decision Tree ditambahkan perhitungan Entropy dan Gain. Rumus hitung Entropy dalam pengujian ini adalah $Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i$ dan rumus Gain adalah $Gain(S,A) = Entropy(S) - \sum \frac{S_v}{S} Entropy(S_v)$.

Dalam uji data ini dihitung dulu jumlah record Data Training, sentiment positif, negatif dan neutral. Selanjutnya dihitung nilai masing-masing atribut terhadap masing-masing sentiment kemudian dilanjutkan pada perhitungan Entropy dan Gain. Berikut ini tabel data uji/Testing Data yang ditunjukkan pada tabel 4.21 dibawah ini :

Tabel 4.21. Data Uji/Testing Data Decision Tree

		Jumlah	POSITIF	NEGATIF	NEUTRAL	Entropy	Gain
Total		13	4	2	7	1,419556299	
Kata	pilkada	1	1	0	0	0	
	jabat	3	1	1	1	1,584962501	
	kawal	2	1	0	1	1	
	kepala	1	0	1	0	0	
	daerah	2	0	0	2	0	
	kader	1	0	0	1	0	
	demokrasi	1	0	0	1	0	
	proses politik	1	1	0	0	0	
Total						0,8999	
Dokumen	D1	3	2	0	1	0,918295834	
	D2	5	2	0	3	0,970950594	
	D3	3	0	0	3	0	
	D4	2	0	2	0	0	
Total						0,8342	
TF	1	8	3	1	4	1,405639062	
	2	5	1	1	3	1,370950594	
Total						0,0273	

Pada hitung data uji ini, dicari nilai Gain yang terbesar, kemudian dicari nilai Entropy yang selain nol. Apabila ada nilai Entropy > 0 maka akan dibuatkan tabel data latih baru sesuai dengan jumlah atribut yang memiliki nilai entropy > 0 .

Dalam simulasi ini didapatkan atribut Kata memiliki Gain yang paling besar, yaitu 0,8999 dengan Kata *jabat* dan *kawal* yang memiliki nilai entropy > 0 yaitu 1,5849625007 dan 1. Untuk itu dibuatkan data training baru yang berisi tentang kata *jabat*. Berikut ini hasil data training kata *jabat* yang ditunjukkan pada tabel 4.22 dibawah ini:

Tabel 4.22. Data Latih/Training Data Kata Jabat

	Kata	Dokumen	TF	Sentimen
jabat	jabat	D4	1	NEGATIF
	jabat	D1	2	POSITIF
	jabat	D1	1	NEUTRAL

Atas hal ini maka dibuat kembali tabel data uji/Testing Data untuk menguji data latih/Training Data kata jabat. Pada data uji kata jabat, atribut yang digunakan adalah Dokumen dan TF (pembobotan kata). Selanjutnya dihitung kembali entropy dan gain nya kembali. Berikut ini hasil simulasi tabel data uji dari kata jabat yang ditunjukkan pada tabel 4.23 dibawah ini :

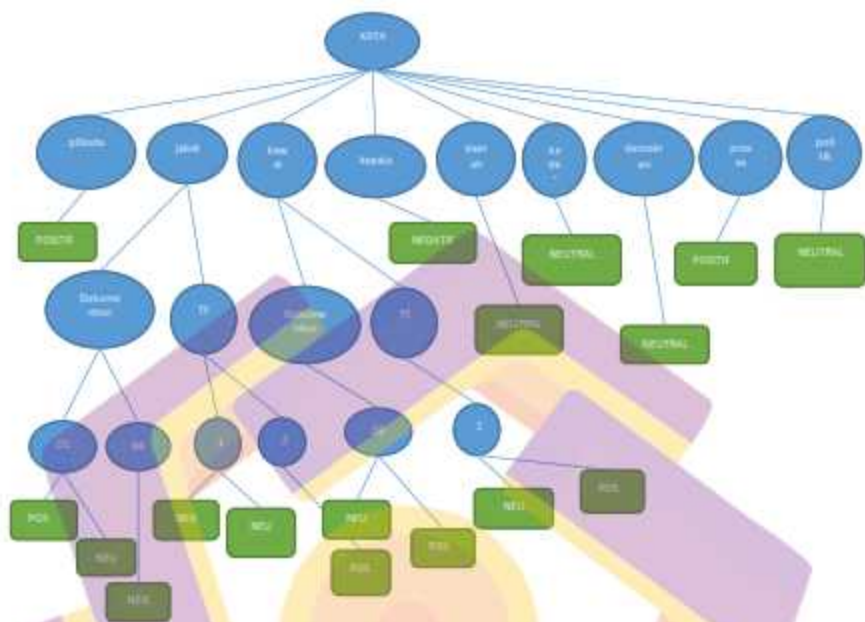
Tabel 4.23. Data Uji/Testing Data Kata Jabat

	Jumlah	POSITIF	NEGATIF	NEUTRAL	Entropy	Gain
Total	13	4	2	7	1,419556	
Dokumen	D1	2	1	0	1	
	D2	0	0	0	0	
	D3	0	0	0	0	
	D4	1	0	1	0	
Total						1,26571
TF	1	2	0	1	1	
	2	1	1	0	0	
Total						1,26571

Proses pembuatan data uji dan data latih ini akan berakhir apabila tidak ditemukan gain yang lebih besar dari yang lain. Jikalau ternyata ditemukan gain yang lebih besar maka harus diperiksa nilai dari entrophynya, apabila nilai semua entropy = 0 maka proses berakhir.

c. Desain Pohon Keputusan/Decision Tree

Pada simulasi ini dihasilkan gambar desain Decision Tree yang ditunjukkan pada gambar 4.7 dibawah ini :



Gambar 4.7. Desain Decision Tree

Susunan desain ini diawali dari atribut yang memiliki gain terbesar menuju atribut dengan gain terkecil. Hal ini ditandai dengan simbol bulat biru, sedangkan untuk hasil sentimennya ditandai oleh persegi panjang warna hijau.

d. Hasil RapidMiner

Tahap selanjutnya adalah implementasi pada aplikasi RapidMiner. Adapun desain dari operator Decision Tree ditunjukkan pada gambar 4.8 dibawah ini :



Gambar 4.8. Layar proses Training dan Testing dengan Decision Tree

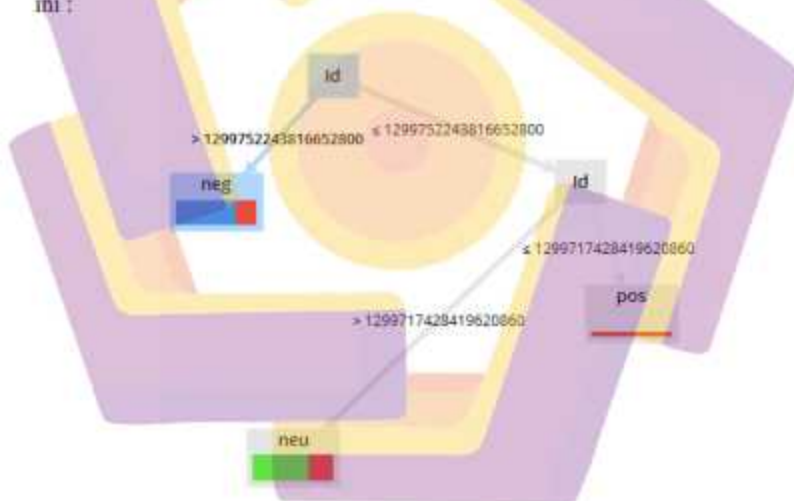
Hasil akurasi dari Decision Tree senilai 33,61 % ditunjukkan pada gambar 4.9 dibawah ini :

accuracy: 33.61% ± 0.04% (micro average: 33.61%)

	true pos	true neu	true neg	class predston
pred pos	0	0	0	0.00%
pred neu	1	0	1	0.00%
pred neg	4455	4432	4001	33.02%
class recall	0.00%	0.00%	99.98%	

Gambar 4.9 Hasil akurasi dari Decision Tree

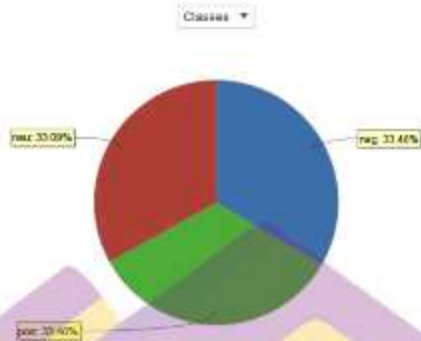
Bentuk Tree pada Decision Tree ditunjukkan pada gambar dibawah 4.10 dibawah ini :



Gambar 4.10. Hasil Tree pada Decision Tree

Berikut ini tabel statistik dari pengolahan algoritma Decision Tree seperti pada tabel 4.24. dibawah ini :

Tabel 4.24. Tabel Hasil Statistik pada Decision Tree



Gambar 4.13. Grafik class penelitian

4.4.2. Skenario Kedua

Mengacu pada gambar 3.3. tentang alur penelitian pada tesis ini maka optimasi pada Algoritma Bat ini dilakukan pada *hasil klasifikasi pembobotan kata pada RapidMiner atau jika dikerjakan secara manual maka optimasi diawali pada hasil Term Weighting TF-IDF / Pembobotan Kata dimana hasil dari pembobotan kata tersebut menjadi bahan olahan data pada Algoritma Bat.* Pada skenario kedua ini, kita akan melakukan pengklasifikasian data menggunakan Algoritma Naive Bayes, Decision Tree dan K-NN dengan menggunakan *data pembobotan kata* yang telah dioptimasi menggunakan Algoritma Bat. Tahapan proses pada skenario kedua ini dijelaskan pada tahapan dibawah ini :

1. Pengidentifikasian kekelawar

Pada proses ini *hasil pembobotan kata dari hasil klasifikasi yang telah dilakukan dalam scenario pertama kita jadikan sebagai bahan utama pemrosesan data yaitu dengan melakukan proses pengidentifikasian kolom kata*

Pada Tesis ini dilakukan percobaan menggunakan fungsi obyektif/fitness $f(x) = x^2$.

2.2. Penentuan nilai β , α , γ , ε , f_{\min} , f_{\max} , A_0

Sesuai dengan keadaan di alam yang belum diketahui oleh kelelawar (jenis microbat) dalam mencari mangsa di suatu lokasi ($f_{\min}=0$) maka kelelawar ini menggunakan sonar suara yang dipancarkan ke segala arah sesuai arah kepala sang kelelawar dimana jika sonar suara tersebut mengenai suatu obyek maka kelelawar tersebut dapat mendengar pantulan suara yang ditimbulkan dari suatu obyek.

Untuk nilai β merupakan nilai random dari uniform distribution $[0,1]$, nilai $\alpha = \gamma = A_0 = 0,9$, $f_{\min} = 0$, $f_{\max} = 1$, ε nilai random dari $[-1,1]$.

2.3. Hitung Frekuensi

Frekuensi pada setiap kelelawar di algoritma ini bersifat tetap pada setiap iterasinya. Dengan menggunakan rumus $f_i = f_{\min} + (f_{\max} - f_{\min})\beta$ maka dihasilkan data frekuensi seperti pada tabel 4.26 dibawah ini :

Tabel 4.26 Hasil perhitungan frekuensi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
4	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
6	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
7	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
8	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
9	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
10	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Nilai β didapat dari nilai random pada rumus excel =rand() sedangkan nilai $f_{\min} = 0$ dan $f_{\max} = 1$. Contoh rumus Frekuensi pada excel adalah = $\$D\$10+(\$E\$10-\$D\$10)*JUP18$.

2.4. Hitung velocity/Kecepatan (V)

Pada hitung kecepatan, seekor kelelawar terbang secara acak dengan kecepatan (V_i) pada posisi x , dengan frekuensi tetap f_i untuk setiap iterasinya. Pada hitung kecepatan dilakukan penjumlahan kecepatan sebelumnya ditambahkan dengan pengurangan posisi x dengan posisi x terbaik (paling minimum) dari seluruh kelelawar. Dengan menggunakan rumus $v_i^t = v_i^{t-1} + (x_i^t - x_*) f_i$ maka dihasilkan tabel kecepatan kelelawar yang ditunjukkan pada tabel 4.27 dibawah ini :

Tabel 4.27 Hasil perhitungan kecepatan kelelawar

The image shows a screenshot of an Excel spreadsheet with a grid of numerical data. The data is organized into columns and rows, with some cells highlighted in green and yellow. The spreadsheet appears to be a calculation table for bat velocity, as indicated by the caption. The data is dense and spans across many columns and rows.

Penerapan rumus Kecepatan/Velocity pada excel dengan memanfaatkan perintah if dengan kondisi apabila nilai random lebih kecil dari Pulse Emission Rate maka kecepatan kelelawar akan dijumlahkan dengan nilai posisi kelelawar dikurangi dengan nilai terkecil dari seluruh nilai posisi kelelawar kemudian dikalikan dengan frekuensi. Sebaliknya apabila nilai random lebih besar maka kecepatan akan tetap. Contoh rumus Velocity pada excel adalah $=IF(JUP18>EJV18;DAQ18+(BRL20-(MIN(\$BRL18:SDAP18))*ILK20);DAQ18)$

2.5. Hasil Perhitungan Loudness/Kebisingan (A)

Kebisingan merupakan hasil dari sonar suara yang dikeluarkan oleh seekor kelelawar. Prinsip kebisingan ini adalah *jika kelelawar makin mendekati obyek*

maka kebisingan makin menurun begitu pula sebaliknya. Karena kelelawar membuka sonar suara diawal dengan cukup keras maka nilai kebisingan ini dapat diset dengan nilai 0,9. Dengan rumus $A_i^{t+1} = \alpha A_i^t, 0 \leq \alpha \leq 1$ maka dihasilkan nilai kebisingan seperti yang ditunjukkan pada tabel 4.28 dibawah ini :

Tabel 4.28 Hasil loudness/kebisingan kelelawar

Pada rumus excel penerapan rumus Loudness ini membandingkan nilai random dengan nilai Loudness sebelumnya dimana jika lebih kecil random maka terjadi update nilai Loudness dengan mengkalikan nilai Loudness sebelumnya dengan nilai dari $\alpha = 0,9$ sedangkan jika nilai random tidak lebih kecil maka nilai Loudness akan tetap. Hal ini dapat menggunakan rumus if. Contoh rumus Loudness pada excel adalah =IF(JUP18<FTA18,\$BS10*FTA18,FTA18)

2.6. Hitung Pulse emission rate

Seekor kelelawar jika telah menemukan obyek maka nilai *Pulse emission rate* cenderung meningkat. *pulse emission rate* bisa diistilahkan sebagai detak sonar. Untuk itu nilai *pulse emission rate* diset dengan nilai yang cukup kecil diawalnya yaitu 0,01. Dengan rumus $r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)]$ maka dihasilkan data seperti pada tabel 4.29 dibawah ini :

Tabel 4.29 Hasil *pulse emission rate* kelelawar

MEXCEL		PULSE EMISSION RATE	
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			

Pada hitung Pulse Emission Rate di excel menggunakan rumus if dimana dilakukan perbandingan apakah nilai random lebih kecil dibandingkan loudness dan nilai fitness lebih kecil dibandingkan nilai ratio. Apabila benar maka dilakukan perkalian antara nilai Pulse Emission Rate sebelumnya dengan nilai 1 dikurangi hasil exponential dari minus γ dikalikan dengan nilai iterasinya. Namun apabila nilai tersebut salah maka nilai Pulse Emission Rate nya tetap.

Contoh rumus Pulse Emission Rate pada excel adalah
 $=IF(AND(JUP18<FTA18;B18<AIG18);EJV18*(1-EXP(-SCS10*SA20));EJV18)$

2.7. Hitung update posisi kelelawar

Update posisi kelelawar dilakukan apabila nilai random kelelawar lebih kecil dari nilai *pulse emission rate* dan nilai fitness nilai kecil dari sebelumnya. Dengan rumus $X_{new} = X_{old} + \epsilon A^t$ maka ditentukan posisi X yang terbaru dimana nilai Epsilon (ϵ) bernilai $-1 \leq \epsilon \leq 1$. Dengan ini hasil dari update posisi kelelawar ditampilkan pada tabel 4.30 dibawah ini :

Tabel 4.30 Hasil update posisi kelelawar

MEXCEL		PULSE EMISSION RATE	
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			

Pada hitung Update Posisi Kelelawar di excel menggunakan rumus if dimana dilakukan perbandingan apakah nilai random kelelawar lebih kecil dari nilai *pulse emission rate* dan nilai fitness nilai kecil dari sebelumnya. Dimana jika bernilai benar maka nilai posisi kelelawar akan ditambahkan dengan perkalian nilai epsilon dengan rata-rata nilai dari kecepatan.

Sebaliknya jika nilai salah maka nilai posisi kelelawar akan ditambahkan dengan nilai dari kecepatan. Contoh rumus excel *pulse emission rate* adalah =IF(AND(JUP20<EJV20;B20<B18);BRL20+(HCF20*AVERAGE(\$FTA20:\$HCE20));BRL20+DAQ20)

2.8. Hitung nilai fungsi obyektif / *fitness* kelelawar

Pada tahap akhir dari optimasi algoritma kelelawar ini adalah melakukan proses perhitungan fungsi obyektif / *fitness* untuk mendapatkan nilai optimumnya dimana perhitungan *fitness* ini menggunakan rumus $f(x) = x^2$. Dari perhitungan rumus *fitness* tersebut didapatkan hasil seperti yang ditunjukkan pada tabel 4.31 dibawah ini :

Tabel 4.31 Hasil perhitungan fungsi obyektif / *fitness* posisi kelelawar

	1	2	3	4	5	6	7	8	9	10
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Pada hitung *fitness* atau fungsi obyektif di excel menggunakan rumus if dimana dilakukan perbandingan apakah nilai *fitness* lebih kecil dari sebelumnya. Apabila lebih kecil maka nilai *fitness* yang baru akan menggantikan nilai *fitness* yang

lama. Apabila tidak lebih kecil maka nilai fitness akan menggunakan nilai fitness yang sebelumnya. Contoh rumus excel untuk hitung fungsi obyektif/fitness adalah =IF(B18<AIG18;B18;AIG20).

2.9. Rekap Data Fitness

Setelah fungsi obyektif/fitness diselesaikan pada masing-masing sentiment kemudian dilakukan rekap seluruh nilai iterasi sentiment per masing-masing id sesuai dengan jumlah id atau dokumen saat pertama kali dilakukan pengujian tanpa menggunakan Algoritma Bat. Hasil rekap iterasi ini nantinya akan dimasukkan kedalam rapidminer untuk dilakukan pengujian kembali hasil akurasi datanya. Hasil rekap iterasi ini ditunjukkan pada tabel 4.32 dibawah ini :

Tabel 4.32 Rekap fungsi obyektif/fitness posisi keelawar pada seluruh sentimen

The image shows a screenshot of an Excel spreadsheet with a large table of data. The table has columns labeled 'ID', 'Fitness', and other numerical values. The data is organized in a grid format with multiple rows and columns. The values appear to be fitness scores for different sentiment IDs. The table is partially obscured by a watermark.

3. Hitung nilai akurasi

Setelah seluruh tahapan optimasi dilakukan maka pada penelitian ini dilakukan beberapa simulasi untuk membandingkan kualitas akurasi dari fungsi yang telah dibuat serta membandingkannya dengan hasil sebelum dilakukan optimasi.

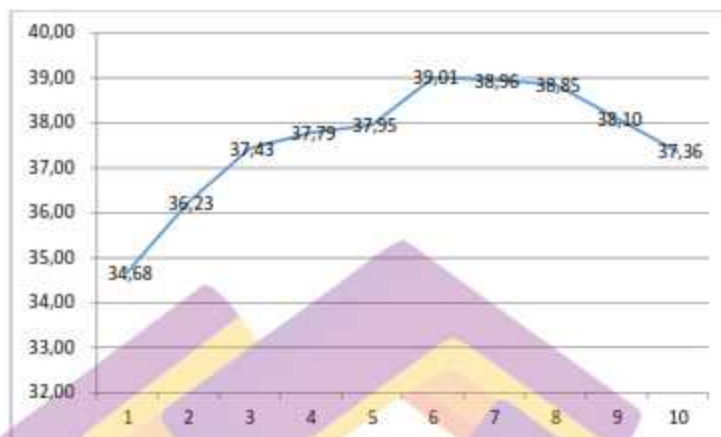
3.1 Optimasi pada Naïve Bayes Classifier

Optimasi pada Naïve Bayes Classifier dilakukan untuk menguji apakah hasil akurasi yang didapat setelah dioptimasi akan memberikan peningkatan atau sebaliknya. Pada Uji optimasi Algoritma Bat pada Naïve Bayes ini dilakukan pengujian dengan mengambil data hasil fitness sesuai dengan jumlah dokumen yang diujikan dimana proses pengujian ini dilakukan iterasi sebanyak 10 kali iterasi dengan atribut Bat dari 10 sampai 100 pada tiap ujinya. Hasil pengujian optimasi Naïve Bayes ini ditunjukkan pada tabel 4.33 dibawah ini :

Tabel 4.33 Akurasi Naïve Bayes dengan 100 Bat

Jumlah Bat	Percobaan akurasi pada fitness ke-i									
	1	2	3	4	5	6	7	8	9	10
100	34,08	36,23	37,43	37,79	37,95	39,01	38,96	38,85	38,10	37,36

Hasil penelitian menunjukkan bahwa nilai pada fitness ke-6 merupakan nilai fitness yang paling optimal dengan tingkat akurasi sebesar 39,01 %. Hal ini dapat dilihat pada gambar grafik 4.14 dibawah ini :



Gambar 4.14. Nilai Pengujian Akurasi Naïve Bayes Untuk 100 Bat Hasil persentase ini didapatkan dari proses percobaan sebanyak 10 kali dengan data yang diacak (10 fold Cross Validation). Hal ini lebih tinggi dari data sebelum dioptimasi yaitu senilai 33,58 %. Pada optimasi Algoritma Bat pada Naïve Bayes ini, dampak yang dirasakan pada proses perhitungan Algoritma Naïve Bayes adalah hasil persentase sentiment pada beberapa class prediction dapat mengalami peningkatan karena kenaikan persentase kesesuaian pada masing-masing atribut. Hal ini berdampak pada peningkatan perhitungan pada nilai prediction confusion table dan hasil akhir akurasi.

3.2. Optimasi pada Decision Tree

Optimasi pada Decision Tree dilakukan untuk menguji apakah hasil akurasi yang didapat setelah dioptimasi akan memberikan peningkatan atau sebaliknya. Dalam klasifikasi Decision Tree nilai pembobotan kata yang telah dioptimasi menggunakan Algoritma Bat kemudian dihitung menggunakan Algoritma Decision Tree dengan 10 kali percobaan sehingga didapatkan hasil dari pengujian

optimasi Algoritma Bat pada Decision Tree seperti yang ditunjukkan pada tabel 4.34 dibawah ini.

Tabel 4.34 Akurasi Decision Tree dengan 100 Bat

Jumlah Bat	Percobaan akurasi pada fitness ke-i									
	1	2	3	4	5	6	7	8	9	10
100	75,38	75,07	74,76	76,03	75,81	75,19	74,72	75,31	75,73	77,50

Hasil penelitian menunjukkan bahwa nilai pada fitness ke-10 merupakan nilai fitness yang paling optimal dengan tingkat akurasi sebesar 77,50 %. Hal ini dapat dilihat pada gambar grafik 4.15 dibawah ini :



Gambar 4.15. Nilai Pengujian Akurasi Decision Tree Untuk 100 Bat

Hasil persentase ini didapatkan dari proses percobaan sebanyak 10 kali dengan data yang diacak (10 fold Cross Validation). Hal ini lebih tinggi dari data sebelum dioptimasi yaitu senilai 76,63 %. Pada optimasi Algoritma Bat pada Decision Tree ini, dampak yang dirasakan pada proses perhitungan Algoritma Naïve Bayes adalah nilai entrophy > 0 lebih sedikit sehingga hal ini membuat level percabangan Decision Tree menjadi lebih pendek. Hal ini membuat hasil

persentase sentiment pada class prediction mengalami peningkatan sehingga membuat nilai prediction confusion table dan hasil akhir akurasi menjadi meningkat.

3.3. Optimasi pada K-NN

Optimasi pada K-NN dilakukan untuk menguji apakah hasil akurasi yang didapat setelah dioptimasi dengan algoritma Bat akan memberikan peningkatan atau sebaliknya. Adapun optimasi pada K-NN adalah pada pembobotan kata yang telah menggunakan data yang telah dioptimasi oleh Algoritma Bat. Selanjutnya dihitung pada tabel data uji/Testing Data. Dari percobaan klasifikasi data yang telah dioptimasi kemudian dengan menggunakan RapidMiner, akurasi K-NN dengan nilai k yang digunakan dapat dilihat pada tabel 4.35 dibawah ini :

Tabel 4.35 Akurasi KNN dari berbagai nilai K

k	1	3	5	9	13	17	21	25	29	33	37
akurasi	56,59	57,09	57,09	59,10	60,82	62,43	63,80	63,01	64,11	65,21	64,50
k	41	45	49								
akurasi	66,15	65,54	64,72								

Hasil penelitian menunjukkan bahwa nilai K=41 merupakan nilai k yang paling optimal dengan tingkat akurasi sebesar 66,15 %. Hal ini juga dapat dilihat dalam bentuk grafik yang ditunjukkan pada gambar 4.16 dibawah ini :



Gambar 4.16. Nilai Pengujian Akurasi KNN

Hasil persentase ini didapatkan dari proses percobaan sebanyak 10 kali dengan data yang diacak (10 fold Cross Validation) kemudian hasil klasifikasi yang muncul dibandingkan dengan data sebenarnya. Perhitungan ini menggunakan confusion matrix untuk menentukan persentase data yang sesuai dengan kenyataan dibandingkan jumlah keseluruhan data yang ada. Nilai k yang digunakan dalam penelitian ini seluruhnya adalah nilai ganjil dikarenakan label atau hasil akhir dari klasifikasi hanya memiliki 3 kemungkinan yaitu positif, negatif dan neutral. Penentuan hasil klasifikasi untuk nilai k lebih dari 1 digunakan metode hasil terbanyak atau mayoritas hasil klasifikasi.

3.4. Rekap Hasil Optimasi

Hasil perhitungan nilai akurasi dari ketiga Algoritma klasifikasi pada penelitian ini ditunjukkan pada tabel 4.36 dibawah ini :

Tabel 4.36 Perbandingan hasil optimasi

Klasifikasi	Optimasi %	
	Sebelum	$f(x) = x^2$
Naïve Bayes	33,58	39,01
Decision Tree	32,82	76,63
K-NN	33,61	66,15

Berikut ini hasil perbandingan optimasi dalam bentuk grafik seperti yang ditunjukkan pada gambar 4.17 :



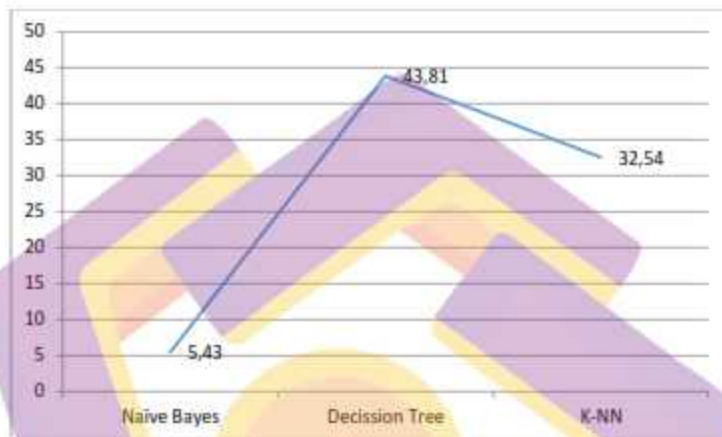
Gambar 4.17 Grafik perbandingan akurasi hasil optimasi dan sebelum optimasi
Penjelasan hasil dan analisa :

Dalam tabel dan gambar perbandingan hasil optimasi diatas menunjukkan bahwa Dari perhitungan didapatkan data seperti ditunjukkan pada tabel 4.38 dibawah ini:

Tabel 4.37. Perbandingan selisih hasil optimasi dibandingkan sebelum optimasi

Klasifikasi	$(f(x) = x^2) - \text{Sebelum}$
Naïve Bayes	5,43
Decision Tree	43,81
K-NN	32,54

Berikut ini hasil tampilan grafik dari selisih peningkatan hasil optimasi dibandingkan dengan sebelum dilakukan optimasi. Gambar grafik ditunjukkan pada gambar 4.18 dibawah ini :



Gambar 4.18 Selisih akurasi hasil optimasi dibandingkan sebelum optimasi

Penjelasan hasil dan analisa :

Dari tabel dan grafik perbandingan selisih peningkatan hasil optimasi dibandingkan sebelum optimasi menunjukkan bahwa ada variasi selisih optimasi yang terjadi dari sebelumnya dimana ada hasil akurasi dari Algoritma Naive Bayes memiliki selisih 5,43% sedangkan pada Algoritma Decision Tree mencapai nilai 43,81% dan Algoritma K-NN mencapai 32,54%. Hal ini menunjukkan Algoritma Decision Tree merupakan Algoritma yang menghasilkan nilai akurasi tertinggi dibandingkan Naive Bayes dan K-NN.

BAB V

PENUTUP

5.1. Kesimpulan

Pada penelitian ini metode optimasi menggunakan algoritma kelelawar menunjukkan suatu peningkatan nilai akurasi yang cukup signifikan dibandingkan sebelum dioptimasi pada kasus algoritma Naïve Bayes, Decision Tree, K-NN. Dari hasil pengujian diatas menunjukkan kesimpulan sebagai berikut :

1. Pada hasil perbandingan peningkatan optimasi, didapatkan bahwa tiap algoritma klasifikasi mengalami peningkatan akurasi dengan menggunakan fitness $f(x) = x^2$ yaitu mulai dari 5,43% hingga 43,81%
2. Pada hasil simulasi akurasi didapatkan Algoritma Decision Tree menempati selisih yang paling tinggi yaitu 43,81% dari sebelum optimasi. Hal ini bisa disimpulkan bahwa dalam penelitian ini algoritma kelelawar cocok digunakan untuk mengoptimasi algoritma klasifikasi KNN, selanjutnya diikuti oleh Decision Tree kemudian Naïve Bayes.

5.2. Saran

Pada penelitian selanjutnya, saran yang dapat diberikan adalah :

1. Para peneliti juga bisa mencoba persamaan fungsi yang lain pada fungsi obyektif/fitness pada algoritma kelelawar.
2. Para peneliti juga dapat melakukan iterasi > 150 kali pada tiap sentimennya serta jumlah Bat > 100 sehingga bisa menghasilkan kesimpulan uji yang lebih kompleks.

DAFTAR PUSTAKA

PUSTAKA BUKU

- Morissan, dkk, 2013. Teori Komunikasi Massa: Media, Budaya, dan Masyarakat. Bogor: Ghalia Indonesia.
- Feldman, R & Sanger, J. (2007) *The Text Mining Handbook-Advanced Approaches in Analyzing Unstructured Data*, USA: New York.
- B. Liu. (2010) *Handbook of Natural Language Processing, chapter Sentimen Analysis, 2nd Edition*.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Hearst, M. A. (1997) *Text data mining: Issues, techniques, and the relationship to information access*. Presentation notes for UW/MS workshop on data mining, July 1997
- Feldman, R & Dagan, I. (1995) *Knowledge discovery in textual databases (KDT). Dalam Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, Canada, August 20-21, AAAI Press, 112-117
- Triawati, Candra, Bijaksana, M.Arif, Indrawati, Nur; Saputro, Widyanto Adi. (2009) *Pemodelan Berbasis Konsep Untuk Kategorisasi Artikel Berita Berbahasa Indonesia. Dalam Seminar Nasional Aplikasi Teknologi Informasi 2009*.
- M. Durairaj & R. Deepika (2015), *Comparative Analysis of Classification Algorithms for the Prediction Of Leukimia Cancer*, Journal Of Advance Research In Computer Science and Software Engineering, Vol. 5, Issue-8, 2015.
- Go Alec & Bhayani, Richa & Huang, Lei (2009), *Twitter Sentimen Classification Using Distant Supervision*.
- Buntoro, Asrofi, Ghulam, 2019, *Analisis Sentimen Calon Gubernur Jawa Timur 2018 Metode Naive Bayes Classifier*, Journal Of Informatics Pelita Nusantara, e-ISSN 2541-3724, Volume 4 No 1, 2019.
- Cahyono, Yono, 2017, *Analisis Sentiment Pada Sosial Media Twitter Menggunakan Naive Bayes Classifier Dengan Feature Selection Particle*

Swarm Optimization Dan Term Frequency, Jurnal Informatika Universitas Pamulang, ISSN 2541-1004, Volume 2 No 1, 1 Maret 2017.

Nurhuda, Faisal & Sihwi, Widya Sari & Doewes, Afrizal (2013), Analisis Sentimen Masyarakat Terhadap Calon Presiden Indonesia 2014 Berdasarkan Opini Dari Twitter Menggunakan Metode Naive Bayes Classifier, Jurnal ITSMART, ISSN 2301-7201, Volume 2 No 2, Desember 2013.

Lorosae, T.A., Prakoso, B. D., Saifudin, S., & Kusrin, K. (2018), Analisis Sentimen Berdasarkan Opini Masyarakat Pada Twitter Menggunakan Naive Bayes, Seminar Nasional Teknologi Informasi & Multimedia Online, 6(1), pp 1-10.

Chandra, K.A, Widiartha, M.I, Muliantara, Agus, Analisis & Implementasi Algoritma Bat Sebagai Fitur Selektor Dalam Klasifikasi Dermatology, Jurnal Ilmiah Ilmu Komputer, Vol. IX, No. 2, ISSN 1979 – 5661, September 2016.

Wardhani, N.K, Rezkiani, Kurniawan, Sigit, Setiawan, Hendra, Gata Grace, Tohari, Siswanto, Gata, Windu, Wahyudi, Mochamad (2018), Sentiment Analysis Article News Coordinator Minister Of Maritime Affairs Using Algorithm Naive Bayes And Support Vector Machine With Particle Swarm Optimization, Journal of Theoretical and Applied Information Technology, Vol.96. No 24, ISSN: 1992-8645, E-ISSN: 1817-3195, December 2018.

Rusdianan, D, Rosiya, D, (2019), Analisa Sentimen Terhadap Tokoh Publik Menggunakan Metode Naive Bayes Classifier dan Support Vector Machine, Journal of Computer Engineering System and Science (CESS); Vol. 4 No. 2; p-ISSN : 2502-7131, e-ISSN: 2502-714x, 2 Juli 2019

Chaudhary, Kumar, (2016), Sentiment Analysis using Nature-Inspired Algorithm, Computer Science & Engineering Department Delhi Technological University, Roll No. 2K14/SWE/17.

Himja Khurana, Sanjib Kumar Sahu, Bat Inspired Sentiment Analysis of Twitter Data, Progress in Advanced Computing and Intelligent Engineering, Springer Nature Singapore Pte Ltd, 2018

Badan Pengembangan dan Pembinaan Bahasa, Kemdikbud. *KBBI Daring*. 2016. <https://kbbi.kemdikbud.go.id/entri/optimas> (diakses Juni 12, 2020).

Bansal, Prachi, dan Ramanjot Kaur. "Twitter Sentiment Analysis using Machine Learning and." *International Journal of Computer Applications* 179 (Februari 2018): 0975-8887.

Basari, Abd. Samad Hasan, Hussin Burairah, I Gede Pramudya Ananta, dan Junta Zeniarja. "Opinion Mining of Movie Review using Hybrid Method of

Support." *sciverse sciencedirect*. Durian Tunggal, Malacca, Malaysia: Elsevier, 2012. 453-462.

Chandra, Ketut Ardha, I Made Widiartha, dan Agus Muliantara. "ANALISIS & IMPLEMENTASI ALGORITMA KELELAWAR SEBAGAI FITUR SELEKTOR DALAM KLASIFIKASI DERMATOLOGY." *Jurnal Ilmiah ILMU KOMPUTER Universitas Udayana* (Jurusan Ilmu Komputer, Universitas Udayana Bali) IX, no. 2 (September 2016): 15-24.

Cormen, TH, CE Leiserson, RL Rivest, dan C Stein. *Introduction to Algorithms, Second Edition*. Massachusetts Institute of Technology: MIT Press, 2001.

Dey, Lopamudra, Sanjay Chakraborty, Anuraag Biswas, Beepa Bose, dan Sweta Tiwari. "Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier." *International Journal Information Engineering and Electronic Business (MECS (http://www.mecs-press.org/))* 4 (July 2016): 54-62.

Fong, Simon, Yan Zhuang, Jinyan Li, dan Richard Khoury. "Sentiment Anlysis of Online News using MALLET." *International Symposium on Computational and Business Intelligence*. Macau, China; Thunder Bay, Canada: CPS - Conference Publishing Services, 2013. 301-304.

Hasad, Andi. *Algoritma Optimasi Dan Aplikasinya*. Magister Ilmu Komputer, Bogor: Institut Pertanian Bogor, 2016.

Husin, Muhammad, Cahyo Adi Prasajo, Nur Afifah Sugianto, Listiya Surtiningsih, dan Imam Chollssodin. "OPTIMASI NAÏVE BAYES CLASSIFIER DENGAN MENGGUNAKAN PARTICLE SWARM OPTIMIZATION PADA DATA IRIS." *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)* (Fakultas Ilmu Komputer Universitas Brawijaya) 4, no. 3 (September 2017): 180-184.

Jain, Anurag P, dan Mr. Vijay D Katkar. "Sentiments Analysis Of Twitter Data Using Data Mining." *International Conference on Information Processing (ICIP) Vishwakarma Institute of Technology*. Pune, India: IEEE, 2015. 807-810.

Kumar, Akshi, Arunima Jaiswal, Member, dan IAENG. "Empirical Study of Twitter and Tumblr for Sentiment Analysis using Soft Computing Techniques." *The World Congress on Engineering and Computer Science*. San Francisco, USA: WCECS 2017, 2017. 1-5.

Pradhan, Vidisha M, Jay Vala, dan Prem Balani. "A Survey on Sentiment Analysis Algorithms for Opinion Mining." *International Journal of Computer Applications (ijcaonline)* 133 (January 2016): 7-11.

Rasenda, Hendarman Lubis, dan Ridwan. "Implementasi K-NN Dalam Analisa Sentimen Riba Pada Bunga Bank Berdasarkan Data Twitter." *Jurnal Media Informatika Budidarma (STMIK Nusa Mandiri)* 4, no. 2 (April 2020): 369-376.

Rizk-Allah, Rizk M., dan Aboul Ella Hassanien. "New binary bat algorithm for solving 0-1 knapsack problem." *Complex Intell. Syst.* (2018), 5 August 2017: 31-53.

Rozlini, Mohamed, Mohd Yusuf Munirah, dan Wahid Noorhaniza. "The Effectiveness of Bat Algorithm for Data." *IEEE International Conference on Control System, Computing and Engineering*. Penang, Malaysia: IEEE, 2016. 151-156.

Saputra, Pramana Yoga. "Implementasi Teknik Crawling Untuk Pengumpulan Data." *Jurnal Dinamika Dotcom* 8, no. 2 (Juli 2017): 160 - 168.

Sharma, Anuj, dan Shubhamoy Dey. "A Comparative Study of Feature Selection and Machine Learning Techniques for Sentiment Analysis." *RACS'12*. San Antonio, TX, USA: ACM, 2012. 1-7.

Sumarmo, Heny. "Komparasi Algoritma Klasifikasi Machine Learning." *BINA INSANI ICT JOURNAL* 4, no. ISSN : 2527-9777 (Desember 2017): 189-196.

PUSTAKA LAPORAN PENELITIAN

Pang, Bo & Lilian, Lee., 2008, *Opinion Mining and Sentimen Analysis*. Foundations and Trends in Information Retrieval 2(1-2), pp. 1-135

Dave, Kushal; Lawrence, Steve; Pennock, David M. (2003) *Mining the peanut gallery: Opinion Extraction and semantic classification of product reviews*.

Horn, C., 2010, *Analysis and Classification of Twitter Messages*, Master's Thesis, Graz University of Technology, Austria

Manning, C., Raghavan, P., dan Schütze, H., 2009, *Introduction to Information Retrieval*, Cambridge University Press

Himja Khurana, Sanjib Kumar Sahu, *Bat Inspired Sentiment Analysis of Twitter Data*, Progress in Advanced Computing and Intelligent Engineering, Springer Nature Singapore Pte Ltd, 2018

PUSTAKA ELEKTRONIK

- Yudha Pratomo, 16 Mei 2019, <https://tekno.kompas.com/read/2019/05/16/03260037/apjii-jumlah-pengguna-internet-di-indonesia-tembus-171-juta-jiwa>, diakses pada 18 Oktober 2019
- Charlie M.Sianipar, 22 April 2019, <https://www.tagar.id/jumlah-pengguna-twitter-indonesia-naik-pesat>, diakses pada 18 Oktober 2019
- Wahyunanda Kusuma Pertiwi, 01 Maret 2018, Riset ungkap pola Pemakaian Medsos Orang Indonesia, <https://tekno.kompas.com/read/2018/03/01/10340027/riset-ungkap-pola-pemakaian-medsos-orang-indonesia>, diakses pada 23 Oktober 2019
- Iskandar, Pilkada Serentak di Indonesia Hasilkan 370.000 Tweets, 13 Desember 2015, <https://www.liputan6.com/tekno/read/2388665/pilkada-serentak-di-indonesia-hasilkan-370000-tweets>, diakses pada 23 Oktober 2019
- Larose, D. T. 2006. *Naïve Bayes Estimation and Bayesian Networks, in Data Mining Methods and Models*, John Wiley & Sons, Inc., Hoboken, NJ, USA. doi: 10.1002/0471756482.ch5
- Kao, A & Poteet, S. (2007) *Natural Language Processing and Text Mining*. Springer Verlag, London, England
- Ricci, F.; Rokach, L.; et al, 2011. *Recommender Systems Handbook*. Berlin : Springer.
- Arguello, J., 2013. *Naïve Bayes Text Classification* (https://ils.unc.edu/courses/2013_fall/ins613_001/lectures/04NaiveBayesClassification.pdf). Diakses tanggal 12 April 2020). The University of North Carolina.