

**ANALISIS SENTIMEN TERHADAP NEW NORMAL DI ERA
COVID-19 MENGGUNAKAN ALGORITMA
*K-NEAREST NEIGHBOR (K-NN)***

SKRIPSI

**SUSAN MAYANG SARI
0701162003**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

**ANALISIS SENTIMEN TERHADAP NEW NORMAL DI ERA
COVID-19 MENGGUNAKAN ALGORITMA
*K-NEAREST NEIGHBOR (K-NN)***

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

**SUSAN MAYANG SARI
0701162003**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth,

Dekan Fakultas Sains dan Teknologi

Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengatakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama : Susan Mayang Sari

Nomor Induk Mahasiswa : 0701162003

Program Studi : Ilmu Komputer

Judul : Analisis Sentimen Terhadap New Normal Di
Era Covid-19 Menggunakan Algoritma
K-Nearest Neighbor (KNN)

Dapat disetujui untuk segera *dimunqasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Medan, 25 Maret 2021 M
11 Sya'ban 1442 H

Komisi Pembimbing,

Pembimbing I,



Dr. Mhd Furqan, S.Si., M. Comp.Sc.
NIP. 198008062006041003

Pembimbing II,



Sriani, S.Kom., M.Kom
NIB. 1100000108

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Susan Mayang Sari
Nomor Induk Mahasiswa : 0701162003
Program Studi : Ilmu Komputer
Judul : Analisis Sentimen Terhadap New Normal di Era
Covid-19 Menggunakan Algoritma K-Nearest
Neighbor (K-NN)

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 25 Maret 2021



Susan Mayang Sari
NIM. 701162003



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. IAIN No. 1 Medan 20235
Telp. (061) 661583-6622925, Fax. (061) 6615683
Url: <https://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor : B.106/ST/ST.V.2/PP.01.1/06/2021

Judul : Analisis Sentimen Terhadap New Normal di Era
Covid-19 Menggunakan Algoritma K-Nearest
Neighbor (K-NN)
Nama : Susan Mayang Sari
Nomor Induk Mahasiswa : 0701162003
Program Studi : Ilmu Komputer

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu
Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan
dinyatakan **LULUS**.

Pada Hari/Tanggal : Kamis, 25 Maret 2021
Tempat : Via Zoom Meeting (Home)

Tim Ujian Munaqasyah,
Ketua,

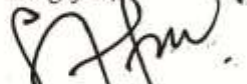

Ilka Zulfa, S.Kom., M.Kom
NIP. 198506042015031006

Dewan Penguji,

Penguji I,


Dr. Mhd Furqan, S.Si., M. Comp.Sc.
NIP. 198008062006041003

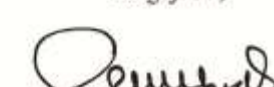
Penguji II,


Sriani, S.Kom, M.Kom
NIB. 1100000108

Penguji III,


Rakhmat Kurniawan R, S.T., M.Kom
NIP. 198503162015031003

Penguji IV,


Armansyah, M.Kom
NIB. 1100000074

Mengesahkan,

Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,


Dr. Mhd. Syahnan, M.A
NIP. 196609051991031002

ABSTRAK

Masa pandemi *covid-19* sudah mewabah virus nyaris ke segala kepenjuru dunia yang mengancam kesehatan, nyawa dan sudah mempengaruhi segala aspek kehidupan manusia. Di Indonesia pemerintah telah menerapkan *new normal* untuk mengembalikan masyarakat beraktivitas normal ditengah pandemi *covid-19* dan menjalankan protokol kesehatan. Penerapan *new normal* menuai beragam komentar dari masyarakat dan masuk kedalam daftar terpopuler di media sosial twitter. Analisis sentimen dilakukan agar dapat memprediksi komentar ataupun opini masyarakat yang kecenderungan beropini positif maupun negatif. Metode *preprocessing* yang akan digunakan yaitu *cleaning*, *case folding*, *normalisasi*, *stemming*, *filtering*, dan *tokenizing*. Pada normalisasi kata bertujuan untuk memperbaiki kesalahan kata pada penulisan (*typo*) berdasarkan KBBI. Metode pembobotan kata yang menggunakan *TF-IDF*. Data yang digunakan terdiri dari 1000 *tweet*. Pada klasifikasi menggunakan metode *K-Nearest Neighbor (K-NN)* mempunyai prinsip yang simpel yaitu bekerja menurut jarak terdekat dari data yang diuji serta data dilatih. Melakukan perbandingan pengujian agar mendapatkan hasil akurasi yang paling terbaik serta mengevaluasi hasil menggunakan *confusion matrix*. Hasil dari pelabelan bahwa kelas sentimen positif lebih unggul berjumlah 811 dibandingkan kelas sentimen negatif berjumlah 189. Hasil dari masing-masing pengujian yang dilakukan memperoleh hasil dengan nilai $k=1$, untuk pengujian *use training set* mendapatkan hasil *accuracy* sebesar 100%, *precision* sebesar 100%, *recall* sebesar 100% dan *f-measure* sebesar 100%. Pada pengujian *10-fold cross validation* yang didapat mendapatkan hasil *accuracy* sebesar 92,60%, *precision* sebesar 99,76%, *recall* sebesar 91,83% dan *f-measure* sebesar 95,63%. Dan yang terakhir *80% percentage split* mendapatkan hasil *accuracy* sebesar 94,50%, *precision* sebesar 100%, *recall* sebesar 93,89% dan *f-measure* sebesar 96,89%.

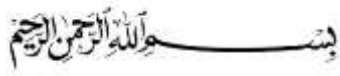
Kata Kunci: Sentimen analisis, *K-nearest neighbor (K-NN)*, *New normal*.

ABSTRACT

During the covid-19 pandemic, the virus has spread to almost all corners of the world, which threatens health, life and has affected all aspects of human life. In Indonesia, the government has implemented a new normal to return people to normal activities amid the covid-19 pandemic with health protocols. The application of the new normal reaped various comments from the public and entered the most popular list on Twitter. Sentiment analysis is to predict comments or public opinions that tend to have positive or negative opinions. The preprocessing methods that will be used are cleaning, case folding, normalization, stemming, filtering, and tokenizing. The word normalization aims to correct word errors in writing (typo) based on KBBI. Word weighting method using TF-IDF. The data used consists of 1000 tweets. The classification using the K-Nearest Neighbor (K-NN) method is has a simple principle, which is to work according to the shortest distance from the data tested and the data trained. Comparison of tests to get the best accuracy results and evaluate the results using a confusion matrix. The results of the labeling that the positive sentiment class is superior to 811 compared to the negative sentiment of 189. The results of each test carried out obtained results with a value of $k = 1$, for testing using the training set the results of accuracy were 100%, precision was 100%, recall of 100%, and f-measure of 100%. In the 10-fold cross-validation test, the results obtained are 92.60% accuracy, 99.76% precision, 91.83% recall, and 95.63% f-measure. And lastly, 80% percentage split got 94.50% accuracy, 100% precision, 93.89% recall and 96.89% f-measure.

Keywords: Sentiment analysis, K-nearest neighbor (K-NN), New normal.

KATA PENGANTAR



Puji dan syukur penulis ucapkan kepada Allah swt. yang telah memberikan rahmat, hidayah serta karunia-Nya dansalawat serta salam tetap tercurah kepada Nabi Muhammad saw. beserta keluarganya dan para sahabat, sehingga penulis mampu menyelesaikan skripsi ini dengan judul “*Analisis Sentimen Terhadap New Normal Di Era Covid-19 Menggunakan Algoritma K-Nearest Neighbor (K-NN)*”.Penyusunan skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Strata-1 di Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara.

Dalam proses pembuatan skripsi ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Syahrin Harahap, M.A selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak Dr. Mhd. Syahnan, M.A selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
3. Bapak Ilka Zufria, M.Kom selaku Ketua Program Studi Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara Medan.
4. Bapak Rakhmat Kurniawan, ST., M.Kom selaku Sekretaris Program Studi Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara Medan.
5. Bapak Dr. Mhd Furqan, S.Si., M.Comp.Sc selaku Dosen pembimbing I skripsi Program Studi Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara Medan yang telah berkontribusi membantu penulis dalam memberikan bimbingan, saran, ide dan motivasi kepada penulis selama penulis mengerjakan skripsi ini.
6. Ibu Sriani, M.Kom. selaku Dosen pembimbing II skripsi Program Studi Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara Medan yang telah berkontribusi membantu penulis dalam memberikan

bimbingan, saran, ide dan motivasi kepada penulis selama penulis mengerjakan skripsi ini.

7. Teristimewa ucapan terima kasih kepada orang tua penulis, Ayahanda Zuned Efendi dan Ibunda Sawiyah, yang selama ini senantiasa selalu mendoakan dan memberikan dorongan baik moril, materil, maupun memotivasi penulis dalam menyelesaikan penelitian ini.
8. Kepada kakak dan adik kandung penulis, Shinta Anggraini dan Sabirin terima kasih untuk dukungan, doa dan semangat, serta bantuan baik moril maupun materil yang selalu diberikan kepada penulis.
9. Seluruh dosen, staf dan karyawan Program Studi Ilmu Komputer dan Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara Medan.
10. Teman-teman seangkatan stanbuk 2016 Program Studi Ilmu Komputer yang tidak dapat disebut satu persatu, teman seperjuangan yang menguatkan dan menyenangkan.
11. Semua pihak yang tidak dapat disebutkan satu persatu, yang telah dengan tulus ikhlas memberikan doa dan motivasi kepada penulis sehingga dapat terselesaikan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak, demi pengembangan kemampuan penulis ke depannya. Akhir kata, hanya kepada Allah swt. Penulis memohon ridho-Nya, semoga keikhlasan dan bantuan yang telah diberikan kepada penulis dapat bernilai pahala disisi-Nya. Semoga skripsi ini dapat bermanfaat bagi para pembaca dan semua pihak yang membutuhkan khususnya dalam bidang Ilmu Komputer. Semoga Allah swt. Melimpahkan rahmat-Nya kepada kita semua. Aamin.

Medan, 16 Oktober 2020

Penyusun,

Susan Mayang Sari

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Text Mining.....	7
2.1.1 Visualisasi Data	7
2.2 Analisis Sentimen.....	8
2.3 Twitter	8
2.3.1 API Twitter	11
2.4 Preprocessing Data	11
2.4.1 Cleaning	11
2.4.2 Case Folding	12
2.4.3 Normalisasi Kata.....	12
2.4.4 Stemming	13
2.4.5 Filtering.....	13
2.4.6 Tokenizing	14
2.5 Klasifikasi.....	14

2.5.1 Pelabelan Kelas Sentimen.....	15
2.6 Pembobotan Tf-Idf	16
2.7 Machine Learning	17
2.7.1 Supervised Learning	18
2.7.2 Unsupervised Learning	18
2.8 Algoritma K-Nearest Neighbor (K-NN)	18
2.9 K-Fold Cross Validation	20
2.10 Confusion Matriks	21
2.11 Bahasa Pemrograman R.....	22
2.12 Weka	23
2.13 Flowchart	25
2.14 Literatur Sejenis	26
BAB III METODE PENELITIAN	30
3.1 Metode Pengumpulan Data	30
3.1.1 Studi Pustaka.....	30
3.1.2 Variabel Penelitian.....	30
3.1.3 Observasi	30
3.2 Metode Analisis Data	31
3.3 Bahan dan Alat Penelitian	32
3.3.1 Perangkat Keras	32
3.3.2 Perangkat Lunak	32
3.4 Gambaran Umum Sistem	33
BAB IV HASIL DAN PEMBAHASAN	34
4.1 Pembahasan	34
4.1.1 Analisis Data.....	34
4.1.2 Representasi Data	38
4.1.2.1 Preprocessing Data.....	38
4.1.3 Hasil Analisis Data	48
4.1.3.1 Pelabelan Kelas Sentimen.....	48
4.1.3.2 Pembobotan <i>TF-IDF</i>	50

4.1.3.3 Klasifikasi K-Nearest Neighbor (K-NN)	57
4.1.3.4 Confusion Matriks.....	58
4.1.4 Perancangan Sistem	59
4.2 Hasil	61
4.2.1 Visualisasi	62
4.2.2 Pengujian	67
4.2.3 Evaluasi Hasil	72
4.2.4 Penerapan.....	74
BAB V KESIMPULAN DAN SARAN	75
5.1 Kesimpulan.....	75
5.2 Saran.....	76
DAFTAR PUSTAKA	77
LAMPIRAN- LAMPIRAN	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1.	Contoh Cleaning	12
2.2.	Contoh Case Folding	12
2.3.	Contoh Normalisasi	13
2.4.	Contoh Stemming	13
2.5.	Contoh Filtering	14
2.6.	Contoh Tokenizing	14
2.7.	Bagan Proses klasifikasi (Han dan Kamber, 2006)	15
2.8.	Alur Pelabelan Sentimen	16
2.9.	Contoh KNN Classification.....	19
2.10.	Ilustrasi k-fold cross validation untuk k = 10.....	20
2.11.	Tampilan Awal Rstudio.....	23
2.12.	Tampilan Utama Weka.....	24
2.13.	Tampilan Weka Explorer	24
3.1.	Contoh Hasil Data Crawling	31
3.2.	Block Diagram Gambaran Umum Sistem.....	33
4.1.	Alur Sistem Crawling Data Twitter.....	34
4.2.	Tampilan Akun API Twitter.....	35
4.3.	Tampilan Consumer Key dan Consumer Secret	35
4.4.	Tampilan Crawling Data pada Aplikasi Rstudio.....	36
4.5.	Proses Data Crawling	37
4.6.	Kumpulan Data Crawling.....	37
4.7.	Proses Menyimpan Data Crawling.....	37
4.8.	Kumpulan Data Crawling Berupa Text.....	38
4.9.	Tampilan Sampel pada Teks Data Crawling.....	38
4.10.	Block Diagram Preprocessing Data	39
4.11.	Alur Proses Cleaning.....	39
4.12.	Alur Proses Case Folding	41
4.13.	Alur Proses Normalisasi	42

4.14.	Alur Proses Stemming.....	44
4.15.	Alur Proses Filtering	45
4.16.	Alur Proses Tokenizing	47
4.17.	Tampilan Hasil Term Frequency.....	52
4.18.	Flowchart Sistem.....	59
4.19.	Flowchart Algoritma K-Nearest Neighbor (K-NN)	60
4.20.	Hasil Kelas Sentimen Terhadap New Normal	61
4.21.	Frekuensi TDM	62
4.22.	Kata Yang Sering Muncul Pada Keseluruhan Data	63
4.23.	Kata Yang Sering Muncul Pada Data Kelas Negatif	64
4.24.	Wordcloud Kelas Sentimen Negatif.....	65
4.25.	Kata Yang Sering Muncul Pada Data Kelas Positif	66
4.26.	Wordcloud Kelas Sentimen Positif	67
4.27.	Data Vector Dari Data Tweet.....	68
4.28.	Hasil Klasifikasi K-NN dengan Use Training Set.....	68
4.29.	Hasil Klasifikasi K-NN dengan 10-Fold Cross Validation	69
4.30.	Hasil Klasifikasi K-NN dengan 80% Percentage Split	69
4.31.	Hasil Perbandingan Pengujian Test K-NN.....	74

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1.	Confusion Matrix.....	21
2.2.	Simbol-Simbol Bagan Alir Flowchart.....	25
4.1.	Proses Cleaning	40
4.2.	Proses Case Folding.....	41
4.3.	Proses Normalisasi.....	43
4.4.	Proses Stemming.....	45
4.5.	Proses Filtering	46
4.6.	Proses Tokenizing.....	47
4.7.	Perhitungan Pelabelan Skor Sentimen.....	49
4.8.	Contoh Proses Perhitungan Term Frequency	50
4.9.	Contoh Proses Perhitungan Document Frequency	52
4.10.	Contoh Proses Perhitungan Inverse Document Frequency	54
4.11.	Contoh Perhitungan Bobot (<i>Weight</i>)	55
4.12.	Contoh Perhitungan Confusion Matrix.....	58
4.13.	Perbandingan Jumlah Data Kelas Sentimen	61
4.14.	Perbandingan Keseluruhan Evaluasi Dari Beberapa Pengujian Test Menggunakan Algoritma K-Nearest Neighbor (K-NN).....	70
4.15.	Waktu Yang Dibutuhkan Untuk Membangun Model	71
4.16.	Confusion Matrix.....	72

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Kamus Stopwords Berbahasa Indonesia
2.	Tampilan Script Program
3.	Daftar Riwayat Hidup
4.	Kartu Bimbingan Skripsi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pandemi virus *covid-19* sudah mewabah virus nyaris ke segala kepenjuru dunia. Tidak hanya mengancam kesehatan serta nyawa, *covid-19* sudah mempengaruhi terhadap segala aspek kehidupan, aspek ekonomi, sosial, pendidikan serta budaya. Saat ini korban yang terinfeksi *virus corona* terus meningkat setiap harinya. *World Health Organization (WHO)* melansir bahwa lebih dari 17,6 juta orang terjangkit dan lebih dari 680 ribu orang meninggal dunia di 216 negara. Sementara kasus di Indonesia telah mencapai 130.718 orang terjangkit, 85.798 orang yang sembuh dan 5.908 orang yang meninggal dunia (Gugus tugas *covid-19*, 12-08-2020).

Permasalahan positif *covid-19* terus bertambah secara signifikan sangat dikhawatirkan bisa menimbulkan potensi konflik sosial ditengah masyarakat. Semenjak pemerintah menerapkan kebijakan PSBB, dimana masyarakat Indonesia diharuskan berdiam diri dan bekerja dirumah masing-masing, hal ini menuai reaksi kekecewaan dari masyarakat dan ada juga masyarakat yang mengabaikan aturan yang ada seperti membuka restoran atau warung makanan yang dipenuhi dengan masyarakat dan kebijakan ini juga menimbulkan dampak kondisi ekonomi yang tidak stabil dan terpaksa melakukan Pemutusan Hubungan Kerja (*PHK*) sedangkan masih produktif dalam bekerja. Pada akhirnya, jika tidak dikelola dengan persiapan yang memadai, kebijakan ini berpotensi menimbulkan konflik sosial.

Penerapan *new normal* yaitu usaha pemerintah untuk mengembalikan masyarakat untuk beraktivitas normal ditengah pandemi *covid-19* dan menjalankan protokol kesehatan untuk pencegahan wabah tersebut serta upaya pemerintah dalam perihal pemulihan ekonomi. Penerapan *new normal* diberlakukan hingga vaksin *corona* ditemuiserta diproduksi massal. Pada penerapan *new normal* ini diharapkan dapat memperbaiki kondisi jarak sosial dan fisik yang melebar akan menjadi sesuatu yang normal. Tetapi tiap kebijakan yang

dicoba oleh pemerintah pastinya terdapat *proserta kontra* dikalangan masyarakat, walaupun sebagian ada yang mentaati peraturan new normal namun ada juga sebagian yang tidak mendukung. Sebagian masyarakat belum teredukasi dengan baik perihal yang boleh serta tidak boleh dilakukan sebab masih terdapat sebagian masyarakat yang tidak taat dalam menjalankan protokol kesehatan.

New normal yang diterapkan pemerintah untuk sejumlah wilayah Indonesia menuai beragam komentar dari para warganet atau sering disebut *netizen*. Komentar-komentar *netizen* yang membahas tentang *new normal* masuk kedalam daftar topik terpopuler di media sosial twitter, ada yang berkomentar negatif dan ada juga berkomentar positif. Dari sekian banyaknya komentar, tidak jarang dari *netizen* yang menentang, berselisih maupun berbeda pendapat dikarenakan masih banyak masyarakat yang tidak mematuhi penerapan *new normal* dalam protokol kesehatan sehingga virus *covid-19* terus meningkat. Hal ini dapat memicu sebuah konflik sosial. Berkaitan dengan hal ini Allah S.W.T. berfirman dalam QS. An-Nisa 4 : 59 sebagai berikut ini :

يَا أَيُّهَا الَّذِينَ آمَنُوا أَطِيعُوا اللَّهَ وَأَطِيعُوا الرَّسُولَ وَأُولِيَ الْأَمْرِ
مِنْكُمْ فَإِنْ تَنَازَعْتُمْ فِي شَيْءٍ فَرُدُّوهُ إِلَى اللَّهِ وَالرَّسُولِ إِنْ كُنْتُمْ
تُؤْمِنُونَ بِاللَّهِ وَالْيَوْمِ الْآخِرِ ذَلِكَ خَيْرٌ وَأَحْسَنُ تَأْوِيلًا ٥٩

Terjemahannya :

“Wahai orang-orang yang beriman! Taatilah Allah dan taatilah Rasul (Muhammad), dan Ulil Amri (pemegang kekuasaan) di antara kamu. Kemudian, jika kamu berbeda pendapat tentang sesuatu, maka kembalikanlah ia kepada Allah (Al-Qur'an) dan Rasul (Sunnahnya), jika kamu beriman kepada Allah dan hari kemudian. Yang demikian itu, lebih utama (bagimu) dan lebih baik akibatnya (QS. An-Nisa 4 : 59).

Kandungan isi dari QS. An-Nisa 4:59 menerangkan bahwa setiap manusia yang beriman harus taat kepada Allah dan Rasulnya serta kepada pemimpin juga

harus taat jika pemimpin itu benar, berdasarkan *Al-Qur'an* dan *Hadist*. Dan apabila terjadi perselisihan dan perbedaan pendapat dalam suatu urusan, maka harus kembali kepada Allah dan Rasul-Nya.

Dimasa sulit dalam menghadapi pandemi *covid-19* sekarang ini sangatlah penting untuk masyarakat mematuhi penerapan pemerintah dengan menerapkan protokol kesehatan dalam pencegahan virus *covid-19*. Jika terjadi perbedaan pendapat, maka seharusnya melakukan *tabayyun* terlebih dahulu, karena jika tidak melakukan *tabayyun* dapat mengakibatkan dampak buruk bagi diri sendiri maupun orang lain. Oleh karena itu, penulis membangun sebuah sistem yang salah satunya berfungsi untuk *tabayyun* dengan mempermudah dan memberi solusi dalam melakukan analisis sentimen dengan mengumpulkan berbagai opini dari masyarakat secara otomatis sehingga mendapatkan hasil yang berguna bagi masyarakat.

Menganalisis opini masyarakat pada *tweet* disebut analisis sentimen atau opini (*sentiment analysis*). Analisis sentimen ialah untuk memprediksi pendapat yang cenderung beropini negatif, positif dan netral (Rizal, 2017). Analisis sentimen merupakan proses analisis dari berbagai bentuk data yang berupa opini atau pandangan sehingga dapat menghasilkan kesimpulan dari berbagai opini yang ada. Data tersebut membagikan informasi untuk bisa membantu orang ataupun organisasi dalam pengambilan kesimpulan. Jumlah data yang besar tidak mungkin bagi manusia dapat membaca serta menganalisis data secara manual.

Untuk melakukan analisis sentimen dibutuhkan metode yang menunjang klasifikasi. Metode yang digunakan adalah metode algoritma *K-Nearest Neighbor* (*K-NN*), berdasarkan hasil penelitian analisis sentimen sebelumnya yang dilakukan oleh Nurjanah et al., (2017). Pada penelitian tersebut dilakukan analisis sentimen terhadap tayangan televisi berdasarkan opini masyarakat pada media sosial twitter menggunakan metode *k-nearest neighbor* dan pembobotan jumlah *retweet*. Dalam penelitian ini menggunakan pengujian pembobotan tekstual dan pembobotan non-tekstual. Hasil yang diperoleh dari pengujian dengan nilai *k* 3 menggunakan pembobotan tekstual yaitu 82,50%, menggunakan pembobotan non-

tekstual 60% dan menggunakan penggabungan keduanya nilai akurasi sebesar 83,33% (Nurjanah et al., 2017).

Berdasarkan kajian yang dilakukan penulis dengan melihat permasalahan yang dilakukan peneliti lainnya pada penelitian klasifikasi, maka penulis tertarik untuk melakukan penelitian pengklasifikasian sentimen menggunakan metode algoritma *K-Nearest Neighbor (K-NN)*. Metode *K-NN* ini memiliki keunggulan dapat mengklasifikasikan data analisis sentimen terhadap opini masyarakat pada media sosial twitter dengan adanya data latih dan data uji. Metode algoritma *K-Nearest Neighbor (K-NN)* merupakan algoritma klasifikasi yang sederhana dan mudah diimplementasikan, data yang digunakan memiliki label sehingga memudahkan dalam proses pengelompokkan ke dalam kelas yang paling sesuai. Metode *K-NN* memiliki kelebihan dalam aspek kemudahan menerjemahkan hasil dan akurasi dari prediksi, serta dapat mengklasifikasikan data secara akurat dengan memilih terlebih dahulu nilai *K* tetangga terdekat dengan tepat (Jabal Tursina, 2019).

Berdasarkan latar belakang yang dikemukakan di atas, penulis akan melakukan penelitian yang berjudul "Analisis Sentimen Terhadap *New Normal* Di Era *Covid-19* Menggunakan Algoritma *K-Nearest Neighbor (K-NN)*".

Tujuan penelitian ini yaitu diharapkan dapat memberi tambahan pengetahuan sebagai media pembelajaran tentang penerapan algoritma *k-nearest neighbor (K-NN)* dan dapat memberikan gambaran mengenai bagaimana analisis sentimen terhadap *new normal* di era *covid-19* pada media sosial twitter untuk mengklasifikasikan opini positif dan negatif untuk berbagai kepentingan serta pengoptimalan informasi media sosial untuk kepentingan publik.

Pada penelitian ini, penulis akan mencoba untuk melakukan klasifikasi teks opini masyarakat Indonesia terhadap *new normal covid-19* guna mengelompokkan opini pada kalimat untuk mengetahui sentimen kelas positif atau negatif. Prosedur pada klasifikasi dimulai dengan menggunakan teks preprocessing yang terdiri dari beberapa tahapan, yakni *cleaning*, *case folding*, *normalisasi*, *stemming*, *filtering*, dan *tokenizing*. Metode pembobotan kata yang digunakan *TF-IDF (Term Frequency-Inverse Document Frequency)* dan menggunakan algoritma *K-Nearest*

Neighbor (K-NN) sebagai metode klasifikasinya. Melakukan pengujian test yaitu *Use Training Set*, *10-Fold Cross-Validation* dan *80% Percentage Split*. Serta mengetahui hasil akurasi terbaik pengujian yang diperoleh menggunakan confusion matrix yaitu *accuracy*, *precision*, *recall* dan *f-measure*.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang diatas, maka rumusan masalah untuk penelitian ini yaitu :

1. Bagaimana tahap pengumpulan data opini masyarakat Indonesia terhadap *new normal* di era *covid-19* pada media sosial twitter?
2. Bagaimana mendapatkan sentimen positif dan negatif dari opini masyarakat pada media sosial twitter?
3. Bagaimana penerapan klasifikasi dengan menggunakan algoritma *K-Nearest Neighbor (K-NN)*?
4. Bagaimana hasil klasifikasi dari penerapan metode algoritma *K-Nearest Neighbor (K-NN)* pada opini masyarakat Indonesia terhadap *new normal* di era *covid-19* pada media sosial twitter?

1.3 Batasan Penelitian

1. Data yang digunakan terdiri dari *tweet* twitter berbahasa Indonesia dengan jumlah data yang digunakan seribu *tweet*.
2. Data yang digunakan hanya mengenai opini masyarakat tentang *new normal*.
3. Data akan melewati tahapan *preprocessing* yaitu *cleaning*, *case folding*, *normalisasi*, *stemming*, *filtering* dan *tokenizing*.
4. Teknik pembobotan kata menggunakan *TF-IDF*.
5. Pengambilan, mengolah dan memproses data opini menggunakan bahasa R dengan Aplikasi *Rstudio* Version 1.3.1093 dan *Weka (Waikato Environment Analysis)* Version 3.8.4 .
6. Algoritma yang digunakan dalam penganalisisan sentiment ini menggunakan algoritma *K-Nearest Neighbor (K-NN)*.

7. Melakukan pengujian test yaitu *Use Training Set, 80% Percentage split* dan *10-Fold Cross-Validation*. Untuk mengevaluasi hasil menggunakan *confusion matrix*.

1.4 Tujuan Penelitian

Tujuan penelitian ini yaitu diharapkan dapat memberi tambahan pengetahuan sebagai media pembelajaran tentang penerapan algoritma *K-Nearest Neighbor (K-NN)* dan dapat memberikan gambaran mengenai bagaimana analisis sentimen terhadap *new normal* di era *covid-19* pada media sosial twitter untuk mengklasifikasikan opini positif dan negatif untuk berbagai kepentingan serta pengoptimalan informasi media sosial untuk kepentingan publik.

1.5 Manfaat Penelitian

Adapun manfaat yang dapat diambil dari penelitian ini adalah :

1. Mengklasifikasikan analisis sentimen berupa opini masyarakat yang terdapat pada twitter dalam jumlah yang besar secara otomatis.
2. Mencari sebuah informasi tentang suatu kebijakan pemerintah terhadap *new normal* di era *covid-19* dan menentukan apakah kebijakan tersebut mendapatkan respon positif atau negatif dari masyarakat Indonesia pada media sosial twitter.
3. Mengetahui berapa keberhasilan algoritma *K-Nearest Neighbor (K-NN)* pada pengklasifikasian analisis sentimen berupa opini masyarakat terhadap kebijakan pemerintah tentang penerapan *new normal* di Indonesia melalui media sosial twitter.

BAB II

TINJAUAN PUSTAKA

2.1 *Text Mining*

Text mining bisa didefinisikan sebagai temuan data baru yang tidak dikenal oleh komputer sebelumnya, yaitu secara otomatis mengekstrak data dari sumber yang berbeda. Proses ini merupakan mencampurkan data yang sukses diekstraksi dari bermacam sumber (Hearst, 2003).

Text mining merupakan penambangan data yang berbentuk teks dari sesuatu dokumen untuk memeriksa kata-perkata dari isi dokumen sehingga penganalisisan bisa dilakukan. Pada *text mining* menggunakan kumpulan teks yang mempunyai format bacaan yang tidak terstruktur. Proses penganalisisan teks menerapkan metode data mining untuk memeriksa pola dalam teks sehingga memperoleh informasi yang bernilai dari sekumpulan dokumen.

2.1.1 Visualisasi Data

Pada metode visualisasi data merupakan konversidata dalam wujud format visual ataupun tabel sehingga ciri dari data dan hubungan di antara data ataupun atribut bisa dianalisis serta dilaporkan. Metode visualisasi data ini bisa mempermudah manusia dalam mempelajari konsep dari data yang hendak ditampilkan sebab pada dasarnya manusia lebih gampang mengartikan suatu foto daripada bacaan sehingga bisa menerangkan tentang arti dari foto tersebut (Nimas Safitri, 2020).

Wordcloud ialah salah satu metodedari *text mining* digunakan untuk menunjukkan kata-perkata yang terkenal dengan kata kunci dari internet dan data teks. *Wordcloud* kerap digunakan untuk menyoroti sebutan populer ataupun tren bersumber pada frekuensi pengguna. *Wordcloud* ialah teknik pendekatan yang bisa menerangkan persoalan riset dengan gampang serta kilat, juga bisa menjelajahi *wordcloud* dengan secara singkat serta bisa menerapkan analisis yang komprehensif. Pada *wordcloud* akan ditampilkan juga kata yang selalu muncul

didalam data teks yang mempunyai wujud yang sangat besar, begitu pula kebalikannya.

2.2 Analisis Sentimen

Analisis sentimen disebut sebagai *emotion artificial intelligence* merupakan pemrosesan bahasa natural, komputasi *linguistik*, analisis teks serta autentikasi *biometrik* untuk menganalisis, mengekstrak, menghitung serta mengkaji informasi khusus secara terstruktur (Rizal, 2017).

Analisis sentimen ialah bidang studi dari pengolahan bahasa natural yang berperan menganalisis pendapat dan juga dapat diartikan sebagai suatu pandangan, keadaan serta perilaku seseorang dalam tulisan ataupun lisan mereka (Hartonoet *al.*, 2016). Analisis sentimen juga sebagai penilaian, evaluasi, perilaku serta emosi seorang terhadap suatu produk, organisasi, orang, kejadian ataupun topik (Liu, 2015). Analisis sentimen adalah sentimen atau komentar yang diekspresikan kedalam bentuk teks dan apakah ekspresi tersebut menunjukkan positif (menguntungkan) atau negatif (tidak menguntungkan) terhadap subjek (Nasukawa & Yi, 2003).

Dari uraian diatas, dapat diambil kesimpulan kalau analisis sentimen ialah cabang studi pada *text mining* ataupun penggalian data berbentuk teks, yang antara lain ada proses pengolahan serta mengekstrak data teks secara otomatis untuk memperoleh informasi. Analisis sentimen salah satu metode dalam mengekstrak data berbentuk pemikiran (*sentiment*) seorang terhadap sesuatu isu ataupun peristiwa. Analisis sentimen bisa digunakan untuk menguak opini publik dari sesuatu isu, kejadian, kepuasan pelayanan, memperkirakan harga saham, serta analisis pesaing bersumber pada data tekstual.

2.3 Twitter

Twitter merupakan jenis situs jejaring sosial dan *mikroblogging online*. *Mikroblog* merupakan alat komunikasi *online* dimana pengguna bisa memperbaharui statusnya tentang komentar pada sesuatu objek ataupun fenomena tertentu (Manalu, 2014).

Twitter memungkinkan para penggunanya untuk berkomunikasi dan menyebarkan informasi melalui mengirim pesan dan membaca pesan secara singkat, padat dan *real time* yang berbasis teks sampai 140 karakter, yang sering disebut dengan istilah kicauan (*tweet*).

Twitter dibuat oleh Jack Dorsey pada maret tahun 2006 serta diluncurkan pada juli di tahun yang sama. Twitter merupakan jejaring sosial yang menyediakan fasilitas untuk para penggunanya dengan dapat membagikan perbaruan (*update*) data tentang sesuatu objek, isu peristiwa, fenomena tertentu, bisnis serta sebagainya (Elcom, 2010).

Sejak dibuat pada oleh Jack Dorsey, twitter mendapatkan popularitas yang luas yang memiliki perkiraan 500 ribu sampai 200 juta pengguna di seluruh dunia. *Tweet* tersebut ditampilkan di halaman profil pengguna dan dikirimkan ke pengguna lain melalui perangkat seluler yang telah dikenal sebagai pengikut (*followers*) (D'Monte, 2009). Popularitas twitter yang tinggi membuat twitter dimanfaatkan untuk berbagai macam keperluan dalam setiap aspek, misalnya untuk kampanye politik, sarana keluhan, sarana pendidikan serta sebagai media komunikasi yang lain.

Twitter dikenal dengan sebutan kicauan (*tweet*) yang bisa dilihat secara publik dapat diakses oleh siapapun, dimanapun dan kapanpun. Twitter membantu pengguna untuk bisa terhubung ke orang terdekatnya dan dapat mempermudah untuk mengetahui informasi, cerita, mengikuti tren dan berita dari berbagai belahan dunia. Berikut adalah beberapa fitur yang dikenal dalam twitter :

1. Home

Home adalah halaman awal yang ditampilkan oleh twitter untuk melihat *tweet* orang lain yang diikuti oleh pengguna.

2. Profile

Profile adalah halaman mengenai data diri pengguna twitter yang dapat dilihat semua orang.

3. Followers

Followers adalah orang yang mengikuti akun seseorang di twitter dan menjadikannya sebagai teman. Pengikut dapat melihat *tweets* seseorang

yang diikuti masuk pada halaman utamanya.

4. Following

Following ialah kebalikan dari *followers*, akun twitter seseorang yang diikuti oleh pengguna lain agar *tweets* yang dikirim oleh orang yang diikuti masuk ke dalam halaman awalnya.

5. Mention

Mention adalah salah satu cara pengguna twitter untuk menandai suatu tweet kepada pengguna lain. *Mention* ditulis dengan tanda '@' dan nama akun pengguna twitter lain. Akun yang ditandai akan menerima notifikasi mention dari pengguna.

6. Hashtag

Hashtag untuk menandai suatu topik isu pembahasan di twitter agar pengguna dapat mencari serta melihat topik yang ditulis oleh pengguna lain dengan mudah. *Hashtag* sering disebut tanda pagar '#' yang penulisannya diikuti dengan topik yang sedang dibahas. Misalnya : #NewNormal, #Covid-19 dan sebagainya.

7. Retweet

Retweet dilakakun dengan menuliskan 'RT' di awal *tweet* untuk menandakan bahwa memposting kembali sebuah konten atau *tweet* orang lain. *Retweet* membantu pengguna twitter dengan cepat membagikan *tweet* kepada semua pengikut (*followers*).

8. Favorite

Favorite adalah tweet yang ditandai oleh pengguna agar tidak tenggelam oleh halaman tweet berikutnya.

9. Trendings Topics

Trendings Topics adalah suatu topik yang sedang banyak dibicarakan dan yang sangat populer di twitter.

10. Direct Message

Direct Message adalah halaman yang berfungsi sebagai tempat pengiriman pesan langsung antara pengguna lain.

2.3.1 Api Twitter

API biasa disebut *Application Programming Interface* yaitu program yang disediakan oleh pihak *developer* aplikasi tertentu agar pihak *developer* aplikasi lainnya dapat lebih mudah untuk mengakses suatu aplikasi. Jadi, *API* twitter dibuat agar mempermudah seseorang untuk mengakses data ataupun informasi web yang ada didalam twitter dengan bertujuan pengembangan aplikasi.

API berperan selaku penghubung sistem yang dibuat dengan twitter. Untuk dapat mengakses *API* dibutuhkan *key*, *secret key*, *access token* dan *secret access token* yang berfungsi sebagai kunci agar aplikasi yang dibuat dapat diketahui oleh twitter dan cara mendapatkannya dengan mendaftarkan aplikasi *API* twitter pada laman <https://developer.twitter.com/>. Kemudian untuk mencari data dari twitter dengan memakai bahasa pemrograman *R* dan dilakukan pencarian data berdasarkan *keyword* dan data yang sudah di *download* akan disimpan ke *database* yang diberi nama *tweet*.

2.4 Preprocessing Data

Preprocessing data ialah tahapan pertama dari *text mining*, proses ini untuk menyiapkan data agar dapat diproses selanjutnya (Sari, 2020). *Preprocessing* merupakan satu proses yang berguna dalam kegiatan teks mining. *Preprocessing* bertujuan untuk meningkatkan kualitas dari data. Jika data itu digunakan untuk proses pencarian pengetahuan kualitasnya rendah, maka pengetahuan yang dihasilkan akan rendah pula (Han & Kamber, 2006).

Preprocessing adalah tahapan seleksi pemeriksaan pada teks untuk melakukan pembersihan, memperbaiki kesalahan pada teks serta menyederhanakan teks sehingga teks dapat diproses lebih lanjut. *Preprocessing* meliputi *cleaning*, *case folding*, *normalisasi*, *filtering*, *stemming*, dan *tokenizing*.

2.4.1 Cleaning

Cleaning adalah tahapan untuk membersihkan atau menghilangkan karakter yang tidak diperlukan pada data dokumen. Karakter yang dihapus berupa seperti tanda baca, *username*, *url*, *mention*, *hashtag*, *retweet* serta simbol atau karakter *numeric* seperti : ('"~&?!><#%{ }([0-9]+:;') [1122].

Teks Data Tweet		Cleaning
Aku benci111	→	Aku benci
SISTEM2/.,		SISTEM
ZONA MERAH1!!!!!!!!!!		ZONA MERAH

Gambar 2.1 Contoh *Cleaning* (Sumber : Jabal Tursina, 2019)

2.4.2 Case Folding

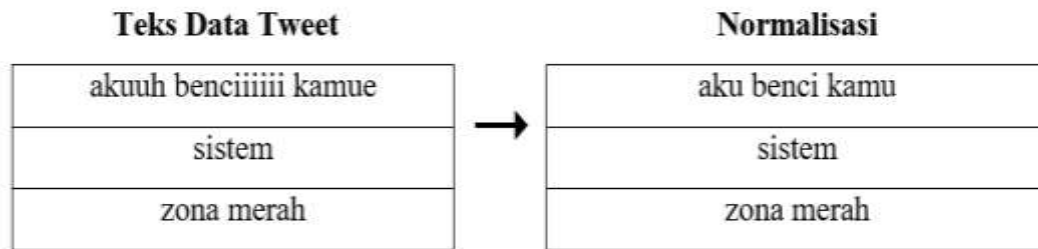
Penggunaan teks pada dokumen tidak semua penulisan hurufnya stabil dalam menggunakan kapital. Oleh sebab itu dibutuhkan proses *case folding* untuk mengubah semua teks huruf capital jadi huruf kecil (*lowercase*) pada semua ulasan yang terdapat didalam dokumen. *Case folding* cuma mendeteksi huruf ‘a’ hingga huruf ‘z’.

Teks Data Tweet		Casefolding
Aku benci	→	aku benci
SISTEM		sistem
ZONA MERAH		zona merah

Gambar 2.2 Contoh *Case Folding* (Sumber : Jabal Tursina, 2019)

2.4.3 Normalisasi Kata

Menormalisasikan kata yaitu sebuah proses untuk mengubah dan memperbaiki kata yang disingkat ke dalam kata yang memiliki arti sama yang berdasarkan KBBI agar menjadikannya informasi yang dapat diproses dan memudahkan dalam proses selanjutnya. Contoh kata yang disingkat seperti *tdk*, *yg*, *smga*, *iy*, *utk*, dan lain sebagainya.

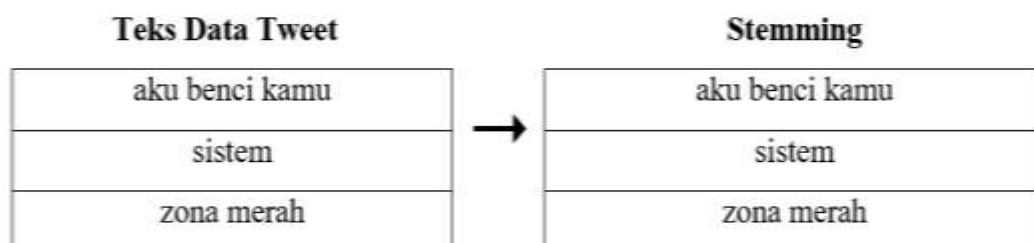


Gambar 2.3 Contoh Normalisasi (Sumber : Jabal Tursina, 2019)

2.4.4 Stemming

Stemming merupakan tahapan yang dilakukan untuk pencarian kata-kata dasar dari setiap kata hasil dari *filtering*. Dalam tahapan *stemming* dilakukan untuk menghilangkan semua kata imbuhan yang terdapat pada data dokumen. Penghapusan kata imbuhan terdiri dari awalan (*prefix*), akhiran (*suffix*) dan kombinasi awalan-akhiran (*konfix*) (Prihantini, 2015).

- Prefiks* (awalan) yaitu imbuhan dibagian pertamadari kata dasar bahasa indonesia yang meliputi : *per-*, *ke-*, *se-*, *me-*, *ter-*, *di-*.
- Surfiks* (akhiran) yaitu yang dibubuhkan pada akhir kata didalam bahasa indonesia yang meliputi : *-an*, *-i*, *-kan* dan *-nya*.
- Konfiks* (gabungan) dalam bentuk kombinasi *prefiks* dan *surfiks* didalam bahasa indonesia meliputi : *di-kan*, *di-i*, *diper-kan*, *diper-i*, *ter-kan*, *ter-i*, *ke-an*, *se-nya*, *ber-kan*, *ber-an*, *pe-an*, *per-an*, *per-kan*, *per-i*, *me-kan*, *me-i*, dll.

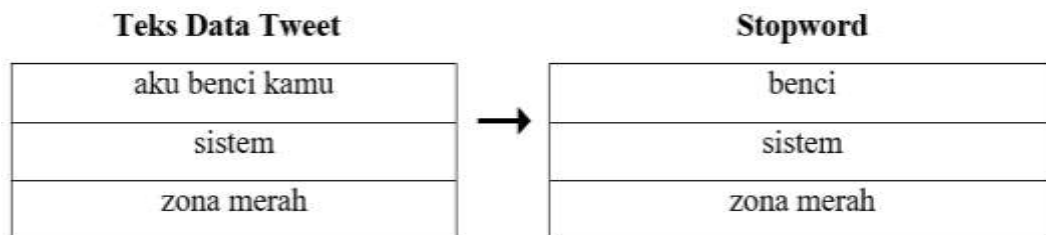


Gambar 2.4 Contoh *Stemming* (Sumber : Jabal Tursina, 2019)

2.4.5 Filtering

Filtering digunakan untuk pengambilan kata penting dan menghapuskan kata yang tidak penting. Dalam proses ini, menggunakan dengan algoritma *stoplist* atau *stopword*. *Stopword* adalah tahap dimana menghapuskan kata yang

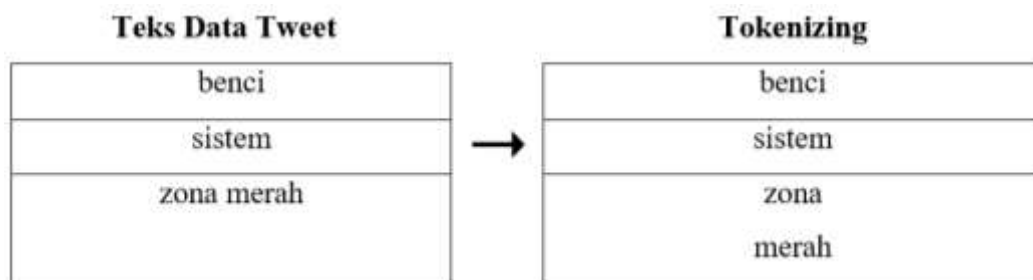
tidak memiliki makna atau kurang penting pada data dokumen. Pada *stopword* yaitu kata-kata yang biasa dipakai tetapi tidak deskriptif dan dapat dibuang, seperti kata “yang”, “itu”, “atau”, “dan”, “dengan”, “di”, “dari” dan sebagainya.



Gambar 2.5 Contoh *Filtering* (Sumber : Jabal Tursina, 2019)

2.4.6 Tokenizing

Tahapan *tokenizing* merupakan tahapan yang dibuat untuk memotong *string input* berdasarkan kata kompilernya (Nugraha *et al.*, 2020). Dalam tahapan *tokenizing* ini, dilakukan pemotongan kalimat sehingga menjadi kata-kata tunggal. Proses ini dilakukan agar memperoleh kata yang mempunyai nilai. *Tokenizing* juga bisa memudahkan untuk mencari frekuensi data di *corpus*.



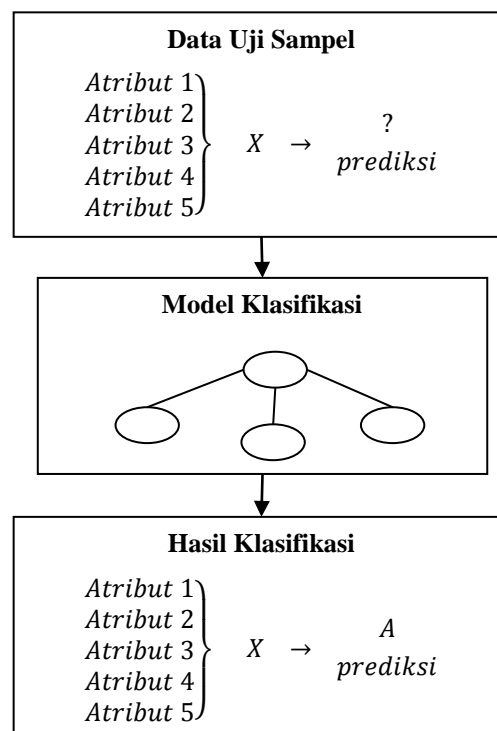
Gambar 2.6 Contoh *Tokenizing* (Sumber : Jabal Tursina, 2019)

2.5 Klasifikasi

Klasifikasi ialah proses menemukan fungsi atau model yang membedakan kelas data dengan bertujuan bisa memakai model tersebut untuk memprediksi kelas objek yang labelnya tidak diketahui (Han & Kember, 2006). Klasifikasi dapat diartikan sebagai teknik yang menerapkan evaluasi pada sesuatu objek data agar data termasuk ke kelas tertentu dari beberapa kelas yang ada. Klasifikasi menerapkan pemakaian model sebagai *prototipe* untuk penyimpanan memori serta

model melakukan prediksi pada sesuatu objek data supaya mengetahui kelas mana objek data tersebut didalam model yang telah tersimpan (Jabal Tursina, 2019).

Klasifikasi dapat diartikan juga dengan sesuatu pekerjaan yang menerapkan pembelajaran pada fungsi f yang memetakan tiap set atribut x ke satu dari beberapa label kelas y yang ada. Hal tersebut akan menciptakan sesuatu model yang setelah itu akan disimpan sebagai memori (Jabal Tursina, 2019).



Gambar 2.7 Bagan Proses klasifikasi

(Sumber : Han & Kember, 2006)

2.5.1 Pelabelan Kelas Sentimen

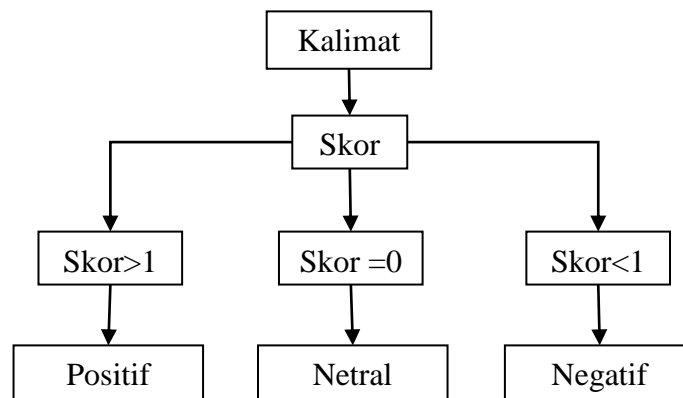
Data yang sudah melewati tahap *preprocessing*, maka setelah itu data akan ditentukan pelabelan sentimennya yaitu positif dan negatif. Pada pelabelan ini dilakukan secara otomatis dengan metode menghitung skor nilai sentimen memakai kamus *lexicon*. Kamus *lexicon* merupakan metode pengklasifikasian yang menggunakan kamus yang berisi kata-kata opini untuk menentukan suatu sentimen positif atau negatif dari suatu data teks.

Kamus yang digunakan pada penelitian ini memakai kamus opini positif dan negatif dari penelitian oleh Liu *et al.*, (2005), yang sudah jadi terjemahan

bahasa Indonesia. Menghitung nilai skor pelabelan sentimen dengan metode perhitungan jumlah skor sentimen positif dikurangi dengan skor sentimen negatif padatiap kalimat komentar (Mahendrajaya *et al.*, 2019). Adapun proses perhitungan skor menggunakan rumus sebagai berikut :

$$Skor = \left(\sum \text{kata positif} - \sum \text{kata negatif} \right)$$

Hasil dari perhitungan skor bilasesuatu kalimat mempunyai skor >0 maka dikategorikan kelas positif, bila kalimat mempunyai skor <0 maka dikategorikan kelas negatif dan bila kalimat mempunyai skor =0 maka dikategorikan kelas netral.



Gambar 2.8 Alur Pelabelan Sentimen

(Sumber : Santoso & Nugroho, 2019)

Contoh data opini atau kalimat komentar yang telah melalui tahap preprocessing "*wilayah ini tidak sehat karena terkena dampak zona merah virus dan aku benci hal tersebut*". Terdapat tiga kata positif dan satu kata negatif yang terdeteksi oleh pelabelan sentimen menggunakan kamus *lexicon*, yaitu "*sehat*" sebagai kata positif, sedangkan "*benci*" dan "*virus*", sebagai kata negatif.

2.6 Pembobotan *TF-IDF*

Pembobotan *TF-IDF* (*Term Frequency-Inverse Document Frequency*) merupakan metode yang melakukan perhitungan pada tiap bobot kata yang terdapat didalam data dokumen. Data dokumen akan diubah menjadi vektor sebanyak kata (*term*) yang akan digunakan untuk klasifikasi. *TF-IDF* digunakan

agar bisa melakukan analisis dengan memakai algoritma *K-Nearest Neighbor* (*K-NN*). Adapun rumus *TF-IDF* ialah sebagai berikut :

$$idf = \log \frac{N}{df} \quad (1)$$

$$w(k, d) = tf(k, d) * idf \quad (2)$$

$$w(k, d) = tf(k, d) * \log \frac{N}{df}$$

Keterangan :

$W(k, d)$ = Bobot kata (*term*) yang tersedia dalam dokumen

$tf(k, d)$ = jumlah frekuensi munculnya kata didalam dokumen

N = jumlah seluruh dokumen yang tersedia didalam database

df = jumlah dokumen yang mengandung *term*

2.7 Machine Learning

Machine learning merupakan kecerdasan buatan (*Artificial Intelligence*) yang bisa membuat sistem mempunyai keahlian belajar sendiri secara otomatis serta tingkatan kemampuannya berdasarkan pengalaman tanpa perlu diprogram oleh manusia. *Machine learning* juga diartikan sebagai ilmu yang memungkinkan komputer berperilaku seperti manusia, di mana komputer dapat meningkatkan pemahamannya melalui pengalaman atau dengan berjalannya waktu secara otomatis. *Machine learning* bertujuan untuk memahami atau mengenali struktur suatu data dan mengonversi data tersebut ke dalam suatu model. *Machine learning* sebagai pengembangan program komputer yang bisa mengakses data serta bisa belajar dari data tersebut. Kemampuan belajar tersebut diperoleh dengan mengakses data dan informasi secara terus-menerus (Kusuma, 2009).

Machine learning juga bertujuan untuk menemukan dan mengaplikasikan pola-pola yang terdapat pada data dengan menggunakan teknik-teknik statistik untuk menemukan pola-pola tersebut. Seringkali data yang dicari polanya berukuran besar. Pengolahan data tidak cuma berupa teks, namun bisa juga berbentuk foto, audio, video, atau aktivitas-aktivitas pengguna selama berselancar atau mengakses internet.

Berdasarkan paparan diatas, dapat disimpulkan bahwa *machine learning* dapat didefinisikan sebagai suatu program komputer yang dapat membuat sistem menjadi cerdas dengan mempelajari data-data yang tersedia di mana program tersebut tidak didefinisikan secara eksplisit. Terdapat beberapa teknik yang dipunyai oleh *machine learning* yaitu *supervised learning* dan *unsupervised learning*.

2.7.1 Supervised Learning

Supervised learning merupakan bagian dari *machine learning* yang bertujuan untuk membangun fungsi hubungan input dan *output* berdasarkan data yang tersedia. *Supervised learning* juga didefinisikan sebagai algoritma yang belajar berdasarkan data latih yang sudah dilabeli, data yang disediakan untuk membangun fungsi disebut sebagai data latih yang selanjutnya akan digunakan untuk memprediksi data *output* atas data input yang tersedia. Selain data latih, juga terdapat data uji yang digunakan sebagai data untuk menguji akurasi sistem atau fungsi (Kusuma, 2009). Algoritma yang digunakan pada *supervised learning* yaitu *decision tree*, *naive bayes classifier*, *k-nearest neighbor classifier*, *support vector machine* dll.

2.7.2 Unsupervised Learning

Unsupervised learning merupakan algoritma *machine learning* yang dalam proses belajarnya tidak memakai data latih ataupun data *training* untuk memprediksi ataupun klasifikasi. Bersumberkan pada model matematisnya, *unsupervised learning* ini tidak mempunyai target variabel namun cuma bertujuan untuk mengkategorikan objek yang hampir sama kedalam suatu area tertentu. Algoritma digunakan pada *unsupervised learning* yaitu *k-means*, *clustering*, *fuzzy c-means*, *DBSCAN* dan *self-organizing map*.

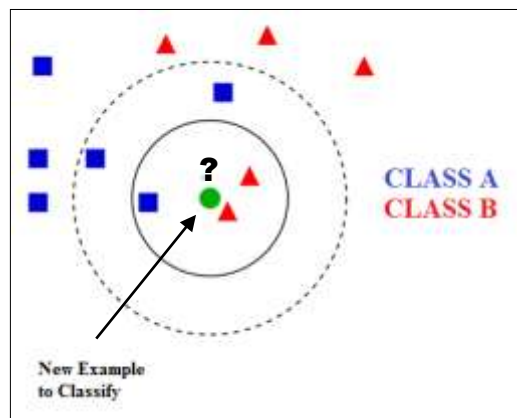
2.8 Algoritma K-Nearest Neighbor (K-NN)

Algoritma *K-Nearest Neighbor (K-NN)* termasuk kedalam algoritma klasifikasi. Tujuan algoritma klasifikasi adalah untuk memprediksi kelas baru dari dataset yang mempunyai kelas (Suntoro, 2019). Algoritma *K-Nearest Neighbor*

(*K-NN*) merupakan salah satu teknik yang sangat simpel untuk memecahkan permasalahan klasifikasi. Algoritma ini kerap dipakai untuk klasifikasi teks dan data (Satrio & Fauzi, 2019).

Algoritma *K-Nearest Neighbor (K-NN)* merupakan algoritma yang digunakan buat menerapkan klasifikasi pada sesuatu data objek kedalam kelas yang sudah ditetapkan sebelumnya berdasarkan nilai k -data latih terdekat dengan objek tersebut (Han & Kember, 2006). Algoritma *K-Nearest Neighbor (K-NN)* merupakan algoritma supervised learning yang bekerja menggunakan hasil dari model yang baru diklasifikasikan bersumber pada mayoritas dari jenis k -tetangga terdekat.

Tujuan algoritma merupakan buat mengklasifikasikan model baru berdasarkan atribut serta sampeldari *training* data. Metode *K-Nearest Neighbor (K-NN)* mempunyai kelebihan berbentuk kestabilan dari berapapun variasi nilai- k . Hasil yang sudah diperoleh lewat implementasi serta pengujian sistem merupakan jumlah data *training* (latih) , keseimbangan jumlah jenis data *training* dan nilai k mempengaruhi ketepatan hasil dari analisis sentimen.



Gambar 2.9 Contoh *KNN Classification*

(Sumber : Antti Ajanki AnAj via Wikimedia Commons, CC BY SA 3.0)
(<https://commons.wikimedia.org/wiki/File:KnnClassification.svg>)

Mencari jauhnya jarak antar titik pada kelas k akan dihitung memakai jarak *Euclidean*. Jarak *Euclidean* merupakan metode pencarian jarak antar dua titik x_1 dan x_2 yang didefinisikan sebagai berikut :

$$\begin{aligned}
 d(x_1, x_2) &= \sqrt{(x_{11} - x_{21})^2 + \dots + (x_{1p} - x_{2p})^2} \\
 &= \sqrt{\sum_{j=1}^p (x_{1j} - x_{2j})^2}
 \end{aligned}$$

Keterangan :

$d(x_1, x_2)$: jarak antara variabel x_1 dan x_2

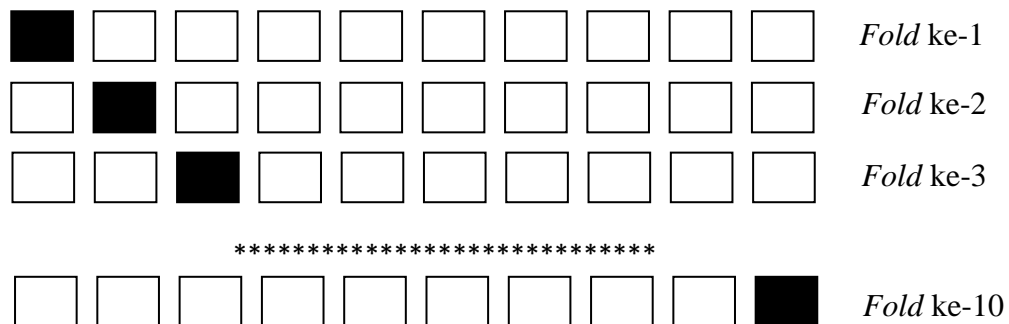
x : variabel

p : jumlah dimensi variabel

2.9 K-Fold Cross Validation

K-fold cross-validation merupakan metode yang terkenal dari *cross-validation*, dan juga merupakan metode dari *data mining* bertujuan untuk mendapatkan hasil akurasi terbaik yang optimal. *K-fold cross-validation* berguna untuk menghitung kinerja proses suatu algoritma yang membagi dataset jadi beberapa k -buah secara acak serta mengelompokkan data itu sebanyak k -fold. Setelah itu salah satu kategori k -fold itu akan dijadikan sebagai data *testing* (uji) dan sisa dari kategori yang lain akan dijadikan data *training* (latih) (Hutapea, 2018).

Pada penelitian inipenulis memakai metode *10-fold cross-validation*. Cara metode *10-fold cross-validation* bekerja yakni membagi data jadi *10-fold* model yang berukuran sama sehingga memiliki 10 subset data yang masing-masing digunakan 9 subset data latih dan 1 subset untuk data uji dengan 10 kali iterasi. Pengukuran *10-fold cross-validation* bersumber pada nilai rata-rata dari 10 kali pengujian.



Gambar 2.10 Ilustrasi k -fold cross-validation untuk $k = 10$

(Sumber : Lukito & Chrismanto, 2015)

2.10 Confusion Matriks

Teknik yang digunakan untuk mengevaluasi hasil yaitu *confusion matrix*. *Confusion matrix* banyak digunakan dalam penelitian untuk mengevaluasi hasil dan mengukur kinerja suatu metode klasifikasi dan juga digunakan untuk menghitung serta bisa menarik kesimpulan dari hasil proses penelitian yang sudah dilakukan. Hal ini dibutuhkan untuk menilai seberapa baik sistem dalam proses klasifikasi data (Mahendrajaya *et al.*, 2019). Dalam *confusion matrix* akan dihitung *accuracy*, *precision*, *recall* dan *f-measure* yang dirumuskan sebagai berikut (Han & Kember, 2006) :

Tabel 2.1 Confusion Matrix

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	FALSE	FN (False Negative) Missing result	TN (True Negative) Correct absence of result

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \times 100\%$$

$$Precision = \frac{TP}{TP + FP} \times 100\%$$

$$Recall = \frac{TP}{TP + FN} \times 100\%$$

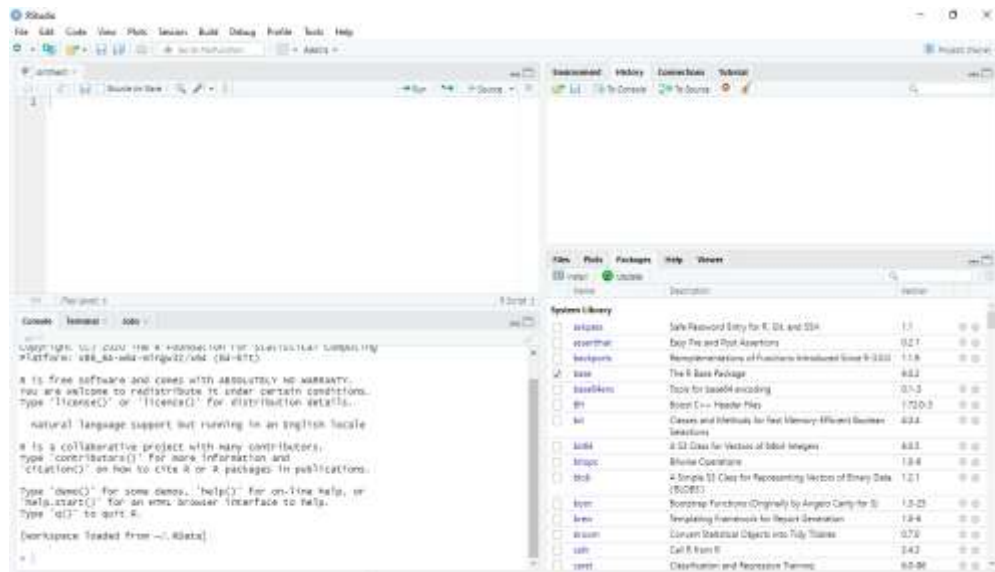
$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \times 100\%$$

2.11 Bahasa Pemrograman R

Bahasa Pemrograman *R* merupakan bahasa pemrograman dan perangkat lunak untuk mengerjakan segala hal yang terkait komputasi statistik dan grafik. Pada mulanya dikembangkan pada tahun 1993 oleh dua orang pakar statistik yaitu Ross Ihaka dan Robert Gentleman di University of Auckland, New Zealand. Bahasa pemrograman *R* semakin populer dan terus berkembang secara pesat hingga sekarang karena meningkatnya kebutuhan perusahaan akan data scientist untuk mengolah dan menganalisis data.

Saat sebelum *R* diketahui terdapat *S* yang dibuat oleh John Chambers dari Bell Laboratories. Dapat dikatakan *R* adalah bentuk lain dari *S*. *R* juga dapat dianggap sebagai versi *free* dari *S*. *R* tersedia sebagai *free software* dibawah lisensi *Free Software Foundation's GNU General Public License*. *R* bisa mengoperasikan berbagai sistem operasi komputer antara lain seperti *UNIX*, *Linux*, *Windows* dan *MacOS*. *R* merupakan sistem aplikasi statistik yang memiliki banyak *package* yang dibuat oleh pengembang dan komunitasnya berguna untuk keperluan analisa statistik, seperti *linear*, *nonlinear*, *classification*, *modeling*, *statistical tests* dan lain-lain. *R* bisa ditambahkan *package* lainnya yang bisa meningkatkan fiturnya (Jabal Tursina, 2019).

Rstudio merupakan perangkat lunak yang terintegrasi *development environment (IDE)* yang ditujukan untuk memudahkan dalam menggunakan *R*. *Rstudio* memiliki antarmuka lebih baik daripada *RGui*. *RStudio* memiliki dua versi lisensi, yaitu *Open Source Edition* dan *Commercial Edition*. *Rstudio* tersedia untuk platform *Windows*, *Mac OS X*, *Linux* versi *Ubuntu* dan *Fedora* (Faisal & Nugrahadi, 2019).



Gambar 2.11 Tampilan Awal Rstudio

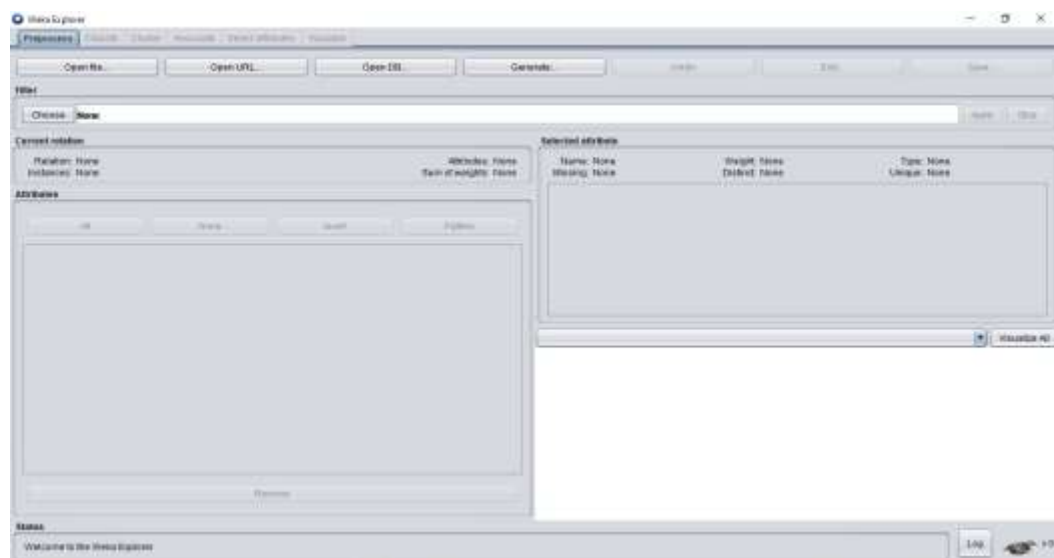
2.12 Weka

Weka singkatan dari *Waikato Environment for Knowledge Analysis* yang dikembangkan di Universitas Waikato, New Zealand (Purnamasari *et al.*, 2013). *Weka* ialah perangkat lunak yang dibentuk memakai bahasa pemrograman java. *Weka* berguna juga dalam penelitian, pendidikan serta bermacam aplikasi yang bisa membantu masalah keperluan *data mining*, terutama pada pengklasifikasian yang berdasarkan pendekatan *machine learning*. Pemakaian *weka* dijalankan dengan metode pembelajaran ke-dataset serta menganalisis buat mendapatkan informasi tentang data.

Weka juga mempunyai kemampuan untuk memvisualisasikan hasil analisis data dalam bentuk *scatterplot*, *bar*, *chart* dan *tree*. *Weka* dapat menerima input dari berbagai sumber, seperti berkas *ARFF*, *CSV*, *XLS*, bersumber dari *database JDBC*, ataupun dari *internet (URL)* (Kurnia, 2012). *Weka* menggunakan metode yaitu *predictive* dan *descriptive* karena sistem ini mendukung metode-metode seperti data *preprocessing*, *regression*, *clustering*, *visualization*, *classification*, dan *feature reduction* (Witten *et al.*, 2011).



Gambar 2.12 Tampilan Utama Weka



Gambar 2.13 Tampilan Weka Explorer

Weka memiliki sub menu yaitu *test options* yang berfungsi untuk pengujian model klasifikasi yaitu sebagai berikut :

a. *Use Training Set*

Use training set untuk pengujian yang memakai data *training* itu sendiri. Maka akurasi yang dihasilkan sangat tinggi, namun belum memberikan hasil akurasi yang sesungguhnya terhadap data lain.

b. *Supplied Test Set*

Supplied test set untuk pengujian yang memakai data lain berbentuk data latih dan data uji secara terpisah sehingga bisa melakukan prediksi pada data *test*.

c. *Cross Validation*

Cross validation melakukan pengetesan dengan menggunakan pilihan banyaknya *fold*. *Default* nilainya ialah 10.





d. *Percentage-Split*

Percentage split untuk pengujian memakai $k\%$ dari data itu, dimana k merupakan bagian dari dataset yang dipakai untuk data *training*. Persentasi dikolom merupakan bagian dari data yang dipakai sebagai *training set*. Pada *options* ini terdapat data *training* dan *testing* dalam satu *file*.

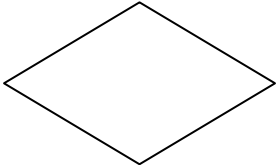



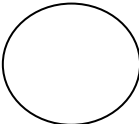
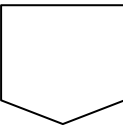
2.13 Flowchart

Bagan yang memiliki arus untuk menampilkan prosedur penyelesaian dalam sesuatu permasalahan. *Flowchart* dapat diartikan sebagai bagan alir dalam program ataupun prosedur sistem secara logika. Berikut bagan alir yang digunakan dalam menggambarkan suatu *flowchart* (Rizki, 2019).

Tabel 2.2 Simbol-Simbol Bagan Alir (Sumber : Rizki, 2019)

Nama	Simbol	Deskripsi
Terminator		Simbol Mulai Dan Selesai
Flowline/Arah Aliran		Arus Aliran Sistem
Proses		Melakukan Suatu Proses
Input/Output		Menggambarkan atau membaca hasil data input/output

Lanjutan Simbol-Simbol Flowchart

Keputusan		Simbol Keputusan Pilihan yaitu True atau False
Preparation		Memberi Nilai Awal Suatu Besaran
Subprogram		Sub Program Memproses Suatu Fungsi Program
Dokumen		Menampilkan Data Yang Bisa Dibaca Secara Fisik Oleh Manusia
On Connector		Penghubung Pada Halaman Yang Sama
Off Connector		Penghubung Pada Halaman Yang Berbeda

2.14 Literatur Sejenis

Pada penelitian oleh Winda Estu Nurjanah, Rizal Setya Perdana, Mochammad Fauzi (2017) Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode *K-Nearest Neighbor* dan Pembobotan Jumlah *Retweet*. Membahas tentang opini masyarakat terhadap tayangan televisi pada media sosial twitter. Data yang digunakan sejumlah 400. Menggunakan preproses yaitu *tokenisasi*, *cleansing*, *case folding*, *filterisasi* dan *stemming*. Pembobotan kata menggunakan *TF-IDF*. Metode yang digunakan yaitu *K-Nearest Neighbor* dengan menambahkan fitur pembobotan jumlah *retweet* (non-tekstual). Hasil yang diperoleh dari pengujian dengan nilai k 3 menggunakan pembobotan tekstual yaitu 82,50%, menggunakan pembobotan non-tekstual 60% dan menggunakan penggabungan keduanya 83,33% (Nurjanah *et al.*, 2017).

Penelitian yang dilakukan oleh Akhmad Deviyanto dan M. Didik R, Wahyudi (2019) Penerapan Analisis Sentimen pada Pengguna Twitter menggunakan metode *K-Nearest Neighbor*. Membahas tentang topik Pilkada DKI 2017. Data yang digunakan sejumlah 2000. Menggumpulkan data menggunakan *Twitter scraper* yang bernama *package python*. Menggunakan preproses yaitu *Case folding*, *tokenizing*, *stopword removal* dan *stemming*. Menggunakan algoritma *K-Nearest Neighbor* dengan pembobotan kata *TF-IDF* dan fungsi *Cosine Similarity*. Hasil dari pengujian diperoleh akurasi terbesar 67,2% dengan nilai $k = 5$, presisi tertinggi 56,94% dengan $k = 5$ dan recall 78,24% dengan $k = 15$ (Deviyanto & Wahyudi, 2018).

Penelitian oleh Rakhman Halim Satrio, Mochammad Ali Fauzi, Indriati (2019) Klasifikasi Tweets Pada Twitter Menggunakan Metode *K-Nearest Neighbor (K-NN)* Dengan Pembobotan *TF-IDF*. Membahas tentang klasifikasi *tweet* yang tersedia di twitter kemudian akan digolongkan jadi satu berdasarkan jenisnya, misalkan subansi sepakbola, voli, serta tenis digolongkan pada kategori olahraga. Klasifikasi data yang dikategorikan yaitu teknologi, kesehatan, ekonomi, olahraga dan otomotif. *Preprocessing* yang dilakukan terdiri dari beberapa tahapan yaitu *tokenizing*, *stopword removal* dan *stemming*. Metode pembobotan yang digunakan ialah *Term Frequency-Inverse Document Frequency (TF-IDF)* dan *K-Nearest Neighbor (K-NN)* untuk metode klasifikasinya. Data digunakan berjumlah 140 data dengan uraian 100 data latih dan 40 data uji serta angka nilai k yang dimasukan ialah 1, 3, 5, dan 7. Hasil akurasi dari masing-masing nilai k yaitu $k=1$ akurasi sebesar 75,0%, $k= 3$ ketepatan 72,5%, $k=5$ ketepatan 62,5% dan $k=7$ ketepatan 55,0% (Satrio & Fauzi, 2019).

Penelitian oleh Rachmad Mahendrajaya, Ghulam Asrofi Buntoro, Moh. Bhanu Setyawan (2019) Analisis Sentimen Pengguna Gopay menggunakan metode *Lexicon Based* dan *Support Vector Machine*. Membahas tentang opini tentang ulasan Gopay dari media sosial twitter. Data yang digunakan sejumlah 1210. Menggunakan *preprocessing* yaitu normalisasi, *cleaning*, *case folding*, *filtering* dan *tokenizing*. Untuk menentukan label kelas sentimen menggunakan kamus *Lexicon Based*. Pembobotan kata menggunakan *TF-IDF* dan klasifikasinya

menggunakan metode *Support Vector Machine (SVM)*. Hasil dari klasifikasi metode *SVM* menggunakan *kernel Linear* menghasilkan akurasi 89,14% dan untuk *kernel Polynomial* akurasinya 84,38% (Mahendrajaya *et al.*, 2019).

Pada penelitian oleh Vynska Amalia Permadi (2020) Analisis Sentimen Menggunakan Algoritma *Naive Bayes* Terhadap *Review* Restoran Di Singapura. Penelitian ini membahas tentang pengelompokan *review* kepuasan pengunjung pada restoran di Singapura. Data yang digunakan berjumlah 1000. Mengumpulkan data dengan metode *web scraping* menggunakan *package “rvest”* pada bahasa pemrograman *R*. *Preprocessing* yang digunakan hanya metode *cleaning*. Metode klasifikasinya menggunakan algoritma *Naive Bayes Classifier*. Hasil dari pengujian klasifikasi yaitu nilai *precision* sebesar 73,02%, *recall* sebesar 74% dan akurasi sebesar 73,33% (Permadi, 2020).

Penelitian oleh Retno Sari (2020) Analisis Sentimen Pada Review Objek Wisata Dunia Fantasi menggunakan Algoritma *K-Nearest Neighbor (K-NN)*. Membahas tentang *review* objek wisata dunia fantasi. Pada penelitian ini data yang diambil dari situs *review www.tripadvisor.com* yang berjumlah 100 data *review*, terdiri dari 50 *review* positif dan 50 *review* negatif. Melakukan *preprocessing* yaitu *tokenize*, *filtering*, *stopword* dan *bi-gram*. Metode algoritma *K-Nearest Neighbor* dengan menggunakan *10-fold cross-validation*. Hasil yang diperoleh pada penelitian ini yaitu akurasi terbesar 77,01 dengan nilai $k = 7$, *precision* 92,38%, *recall* 61,56% dan nilai kurva AUC 0.894 (Sari, 2020).

Pada penelitian penulis sekarang yaitu topik yang akan dianalisis mengenai analisis sentimen terhadap *new normal* di era *covid-19* menggunakan algoritma *K-Nearest Neighbor (K-NN)*. Data yang akan digunakan yaitu *tweet* tentang opini masyarakat terhadap kebijakan pemerintah tentang *new normal* yang akan diambil dari media sosial twitter. Data *tweet* akan melalui tahap *preprocessing* yang akan digunakan yaitu *cleaning*, *case folding*, *normalisasi*, *stemming*, *filtering*, dan *tokenizing*. Pada normalisasi kata bertujuan untuk memperbaiki kesalahan pada penulisan kata tidak sesuai (*typo*) berdasarkan KBBI. Metode yang digunakan ialah algoritma *K-Nearest Neighbor (K-NN)* sebagai klasifikasinya. Untuk mendapatkan nilai akurasi terbaik dengan

menggunakan metode *k-fold cross validation* dan melakukan pengujian tes yaitu *use training set*, *10-fold cross-validation* dan *80% percentage split*. Sebagai evaluasi hasil menggunakan *confusion matriks* yang terdiri dari *accuracy*, *recall*, *precision* dan *f-measure*.

BAB III

METODE PENELITIAN

3.1 Metode Pengumpulan Data

3.1.1 Studi Pustaka

Pada penelitian ini penulis mengumpulkan data-data melalui literatur sejenis, jurnal, skripsi, buku dan yang berhubungan dengan kajian teori.

3.1.2 Variabel Penelitian

Variabel penelitian ialah sesuatu atribut ataupun objek yang bisa berupa apa saja yang penulis tetapkan untuk dipelajari sehingga bisa mendapatkan informasi yang setelah itu bisa ditarik kesimpulannya (Sugiyono, 2011). Penelitian ini menggunakan variabel yaitu *review* yang merupakan opini atau isi komentar pengguna twitter mengenai *new normal*.

3.1.3 Observasi

Observasi dilakukan oleh penulis secara langsung dengan mengambil data opini masyarakat mengenai kebijakan pemerintah tentang *new normal covid-19* pada media sosial twitter melalui *API* twitter.

Analisis sentimen pada penelitian ini dibatasi hanya pada topik pendapat atau opini masyarakat tentang *new normal* saja. Data yang digunakan berbentuk *tweet* bahasa Indonesia yang diambil dari media sosial twitter. Proses pengambilan data menggunakan metode *crawling* pada twitter yaitu memanfaatkan fasilitas *Application Interface (API)* yang tersedia oleh twitter. Data tersebut diambil dengan *API* twitter dengan *keyword* “*new normal*”. Untuk mengakses *API* twitter menggunakan url yaitu <https://developer.twitter.com>. Pada sistem *API* twitter harus menggunakan *consumer key* dan *secret key* pengguna agar bisa mengakses *tweet* yang bersangkutan. Data yang diambil dari twitter ialah sebanyak 1000 *tweet*. Proses pengumpulan data *tweet* hanya mengambil data yang berbentuk *text* dan tidak memiliki gambar. Data yang telah dikumpulkan ialah sebagai input sistem agar dapat diproses lebih lanjut. Adapun contoh data hasil *crawling* yakni sebagai berikut :

@DilaKamilahA15 View 27 September 2020. Tapi juga bisa sebabkan penularan massal plus banyak kematian. Tapi, terima kasih, kita jadi makin yakin isi kepala pejabat Indonesia. Kesehatan dan keselamatan rakyat jelas bukan prioritas. #NewNormal

@CommuterLine View 26 September 2020. Di era kenormalan baru, mari utk tidak berbicara secara langsung/melalui telepon genggam saat di dlm KRL. Ayo... #NewNormal

@farididoev View 26 September 2020. Rakyat menjadi korban #NewNormal yg di Kampanyekan rezim ini sebelum waktunya. #NewNormal

@FaqihHendra3 View 26 September 2020. Negara New Normal, rakyat makin percaya Covid19 itu tidak ada. Pemerintah tuding rakyat tidak disiplin, lha ini wakil rakyat yang bikin. #NewNormal

@16putiandamdewi View 26 September 2020. Mau diluar atau pun didalam rumah ada enak dan gak enaknya. Bersyukur kuncinya. Tetap jaga kesehatan karena kita sekarang hidup di peradaban baru. #NewNormal

@Hilmahilmo1 26 September 2020. Untuk calon kepala daerah harus menjadi influencer dan mengajak pendukungnya untuk mengikuti protokol kesehatan... #NewNormal

@bappeda_jambi 26 September 2020. Bersama kita memutus mata rantai penyebaran Covid-19, dengan disiplin menerapkan protokol kesehatan! #newnormal

Gambar 3.1 Contoh Hasil Data *Crawling*

3.2 Metode Analisis Data

1. Analisis Deskriptif, berguna untuk memberi gambaran terhadap *new normal* berdasarkan opini masyarakat yang ada pada media sosial twitter.
2. Analisis Sentimen, untuk memberi pelabelan sentimen pada data yang dimasukkan pada kategori sentimen positif dan negatif berbasis kamus *lexicon*.
3. Visualisasi, digunakan untuk mengidentifikasi dan membentuk pola kata yang berguna untuk memperoleh data yang penting dan bermanfaat..
4. Metode Pengujian. Pada penelitian ini akan digunakan klasifikasi dengan algoritma *K-Nearest Neighbor (K-NN)* dan melakukan pengujian test yaitu *Use Training Set, 10-Fold Cross-Validation dan 80% Percentage Split*.
5. *Confusion Matriks* digunakan untuk mengevaluasi hasil dan mengukur kinerja dari suatu metode klasifikasi sehingga mendapatkan kesimpulan dari proses penelitian yang dilakukan.

3.3 Bahan Dan Alat Penelitian

Adapun bahan yang digunakan penulis pada penelitian ini ialah sebagai berikut :

3.3.1 Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam sistem dengan spesifikasi sebagai berikut :

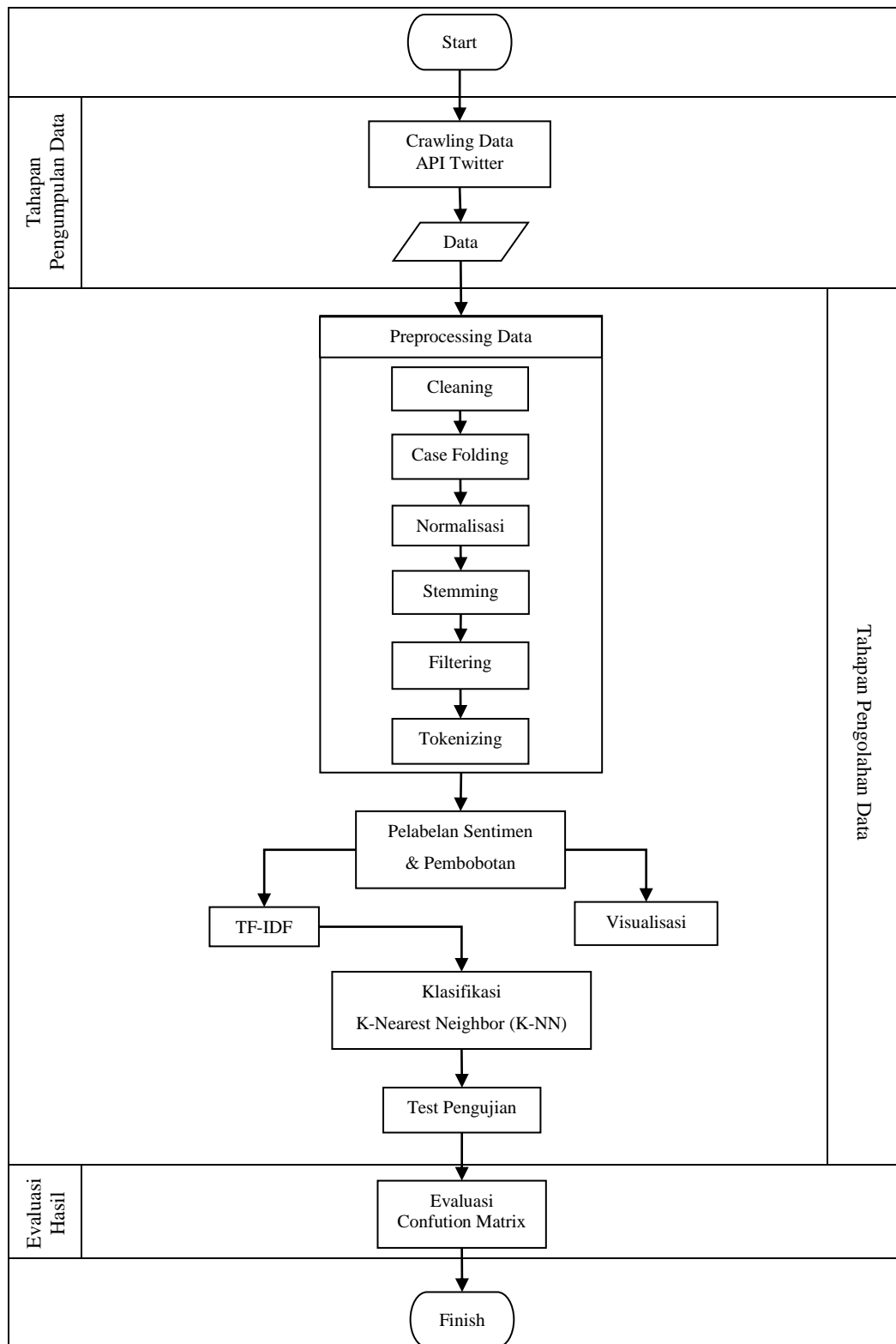
- a. Processor : Intel(R) Celeron® CPU N3060 @1.60Hz 1.60GHz
- b. Memori : 4 GB
- c. Hardisk : 465 GB

3.3.2 Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan untuk membangun sistem yaitu :

- a. Sistem Operasi : Windows 10
- b. Tipe Sistem : 64bit
- c. Perangkat Lunak : Rstudio Version 1.3.1093 & Weka (Waikato Environment Analysis Version 3.8.4)

3.4 Gambaran Umum Sistem



Gambar 3.2 Block Diagram Gambaran Umum Sistem

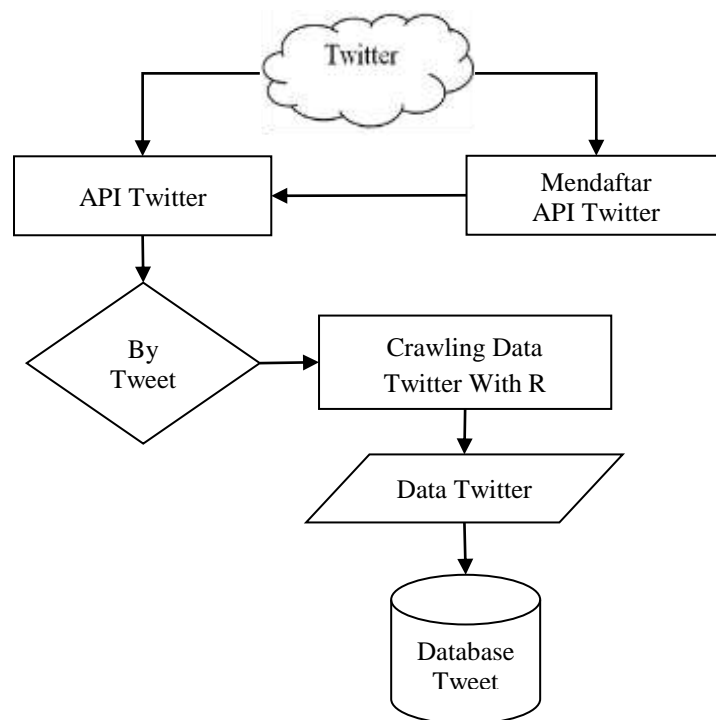
BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

4.1.1 Analisis Data

Dalam penelitian ini penulis akan menganalisis data sentimen terhadap *new normal* di era *covid-19* dengan melakukan pengumpulan data. Proses pengumpulan data opini masyarakat mengenai *new normal* di era *covid-19* di media sosial twitter menggunakan metode *crawling* di twitter yaitu memanfaatkan fasilitas *Application Interface (API)* yang tersedia oleh twitter dengan alamat <https://developer.twitter.com>.

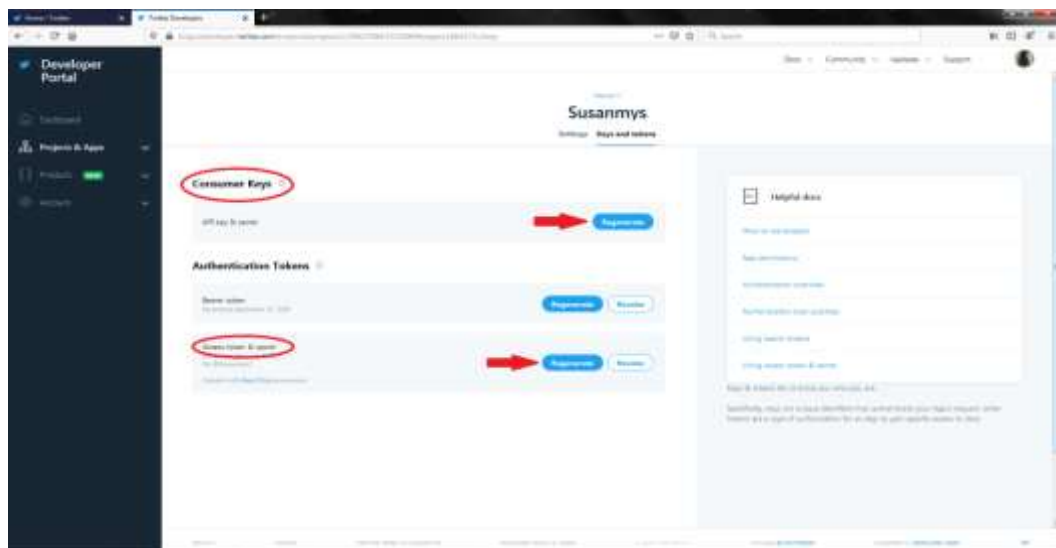


Gambar 4.1 Alur Sistem *Crawling* Data Twitter

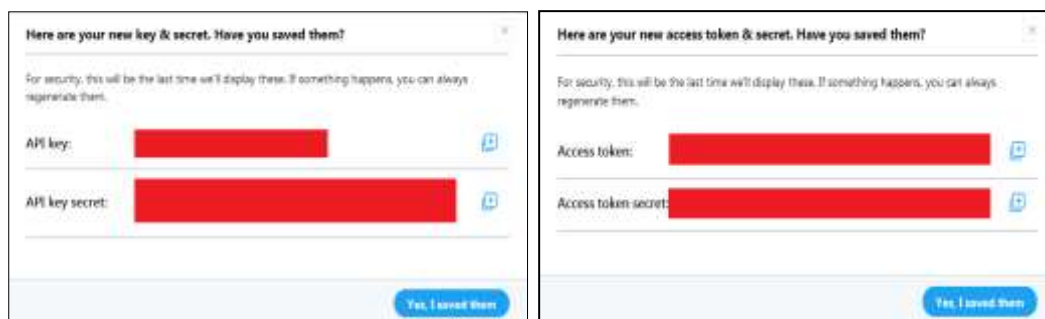
Untuk dapat menggunakan *API* twitter pertama harus mempunyai akun twitter pribadi terlebih dahulu, selanjutnya buka halaman *developer* twitter,

kemudian mendaftarkan akun *App twitter* dengan mengisi *form* data pada *API Key*. Sesudah akun terdaftar lalu bisa memperoleh *API Key* yang berfungsi sebagai penghubung antara aplikasi *Rstudio* dan twitter. Selanjutnya agar dapat mengakses *tweet* pada sistem *API* twitter harus menggunakan *consumer key* dan *consumer secret* seperti Gambar 4.1, dan klik pada bagian *regenerate* setelah itu akan tampil *key* yang dapat dilihat pada Gambar 4.2.

Consumer key dan *consumer secret* ini nantinya berfungsi untuk sebagai kata kunci atau kode untuk mengakses *API* agar nanti sistem aplikasi yang dibuat dapat diketahui oleh twitter. *Key* pada *API* twitter ini bersifat rahasia karena setiap pengguna bisa mempunyai kode akses yang berbeda-beda, jika tidak dijaga kerahasiaannya bisa dipakai oleh pengguna lain.



Gambar 4.2 Tampilan Akun *API* Twitter

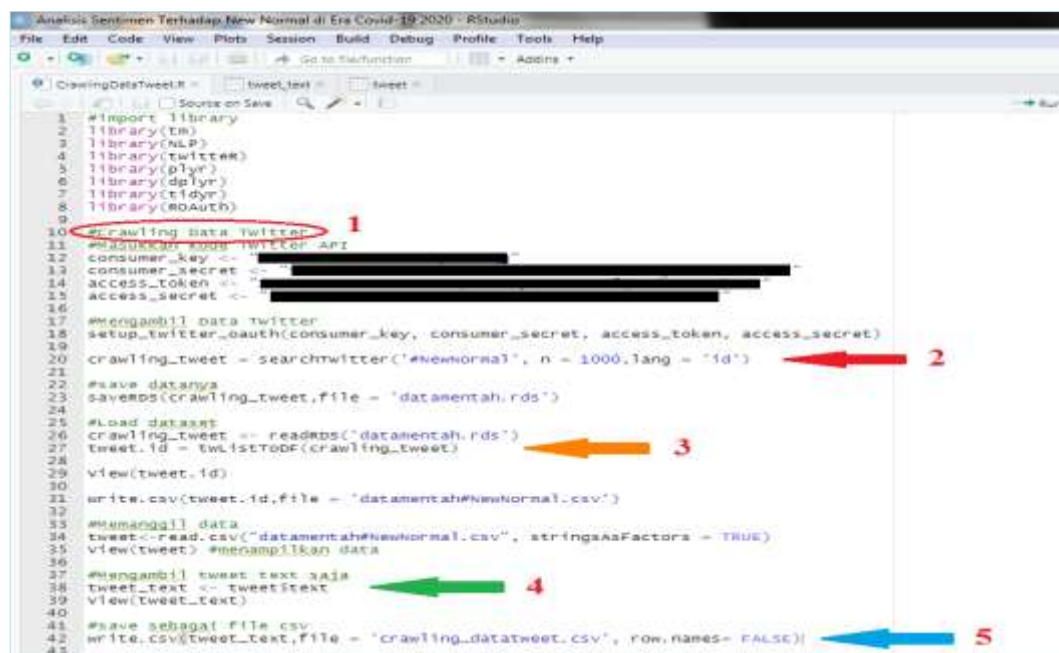


API Key & Secret Key ID

Access Token & Access Token Secret

Gambar 4.3 Tampilan *Consumer Key* dan *Consumer Secret*

Crawling data twitter akan menggunakan bahasa pemrograman *R* pada Aplikasi Rstudio. Sesudah memperoleh *APIkey* proses bisa dilakukan dan kemudian mengaktifkan *package* pada Rstudio yang telah terinstal sebelumnya yaitu "*twitteR*". Jika *package* telah diaktif, setelah itu memproses autentifikasi *APIkey* yang sudah didapat tadi. Kemudian *consumer key* dan *consumer secret key* di *copy*, selanjutnya membuka aplikasi Rstudio dan menekan tombol *ctrl+v* pada program (lingkaran berwarna merah pada Gambar 4.4). Kemudian data *tweet* ini diambil dari *API* twitter dengan *keyword* "*#NewNormal*". Data *tweet* yang diambil dari twitter ialah sebanyak 1000 *tweet*, proses pengumpulan data *tweet* hanya mengambil data yang berbentuk teks berbahasa Indonesia dan tidak memiliki gambar (tahap ke-2 panah berwarna merah pada Gambar 4.4).



Gambar 4.4 Tampilan *Crawling* Data pada Aplikasi Rstudio

Setelah melakukan *crawling* data dengan *keyword* pencarian *#NewNormal* pada *API* twitter dilanjutkan dengan menyimpan hasilnya dalam daftar yang diberi nama *crawling_tweet* dan disimpan dengan nama *datamentah.rds*, untuk dapat melihat dan membaca hasil *list* data *tweet* akan diubah menjadi *data.frame* menggunakan fungsi *twListToDF* dari *package* *twitteR* dengan nama *tweet.id* seperti pada Gambar 4.5 :



Gambar 4.8 Kumpulan Data *Crawling* Berupa Teks

@CommuterLine View 26 September 2020. Di era kenormalan baru, mari utk tidak berbicara secara langsung/melalui telepon genggam saat di dlm KRL. Ayo... #NewNormal <https://t.co/AbdtRIIVg2>

@FaqihHendra3 View 26 September 2020. Negara New Normal, rakyat makin percaya Covid19 itu tidak ada. Pemerintah tuding rakyat tidak disiplin, lha ini wakil rakyat yang bikin. #NewNormal #KhalifahAjaranIslam #Covid19 <https://t.co/BTXjsiA1oo>

@DilaKamilahA15. View 27-09-2020. Tapi juga bisa sebabkan penularan massal plus banyak kematian. Tapi, terima kasih, kita jadi makin yakin isi kepala... <https://t.co/EneBbTNFbr>

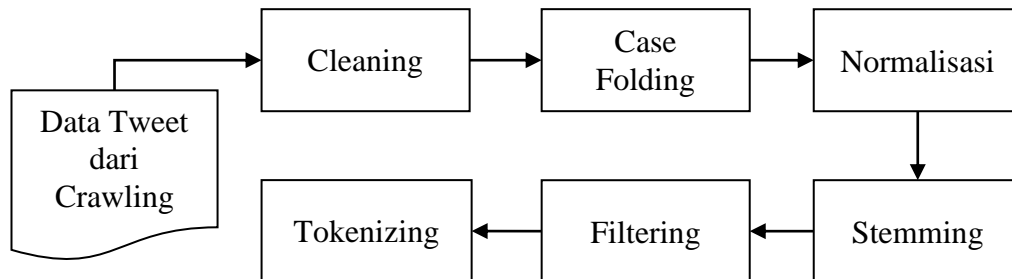
Gambar 4.9 Tampilan Sampel pada Teks Data *Crawling*

4.1.2 Representasi Data

4.1.2.1 Preprocessing Data

Pada tahapan ini data yang digunakan berbentuk data teks yang sudah diambil sebelumnya dari proses *crawling* pada *API* twitter dan akan melakukan proses *preprocessing* data karena data tersebut masih mempunyai bentuk teks yang tidak terstruktur yang mempunyai banyak *noise*, jadi data itu perlu dibersihkan terlebih dahulu.

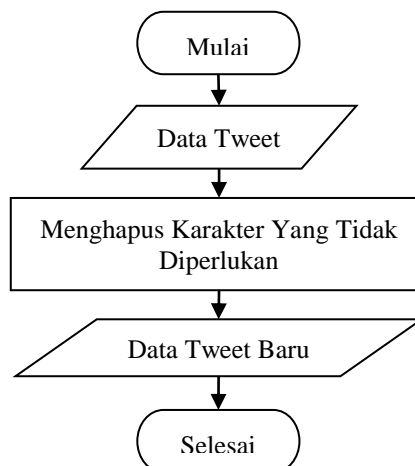
Berikut alur yang menjelaskan tahapan sistem yang sedang melakukan proses *preprocessing* meliputi *cleaning*, *case folding*, *normalisasi*, *filtering*, *stemming*, dan *tokenizing* :



Gambar 4.10 Block Diagram *Preprocessing* Data

a) *Cleaning*

Tahap *cleaning* untuk membersihkan atau menghilangkan karakter yang tidak diperlukan pada data *tweet* seperti tanda baca, *numeric*, *url*, *username*, *mention*, *hashtag* dan *retweet* seperti (“”~&?!><#%{}([0-9]+;,:) [1122] :



Gambar 4.11 Alur Proses *Cleaning*

Langkah-langkah pada proses *cleaning* yang dapat diimplementasikan pada kasus klasifikasi analisis sentimen ini yaitu sebagai berikut (Nugraha,*et al.* 2020) :

1. Menghapus tanda baca, angka, dan karakter selain huruf seperti `//,,:;''^*[]()><{}\\-+=&!?` dll.

2. Menghapus *username* yang ditandai dengan *mention* '@' diawal kata seperti @susanmys
3. Menghapus *url* dan *retweet* seperti *http*, *https* dan *www*. *Retweet* yang ditandai dengan 'RT'.
4. Menghapus *noise-noise* lainnya.

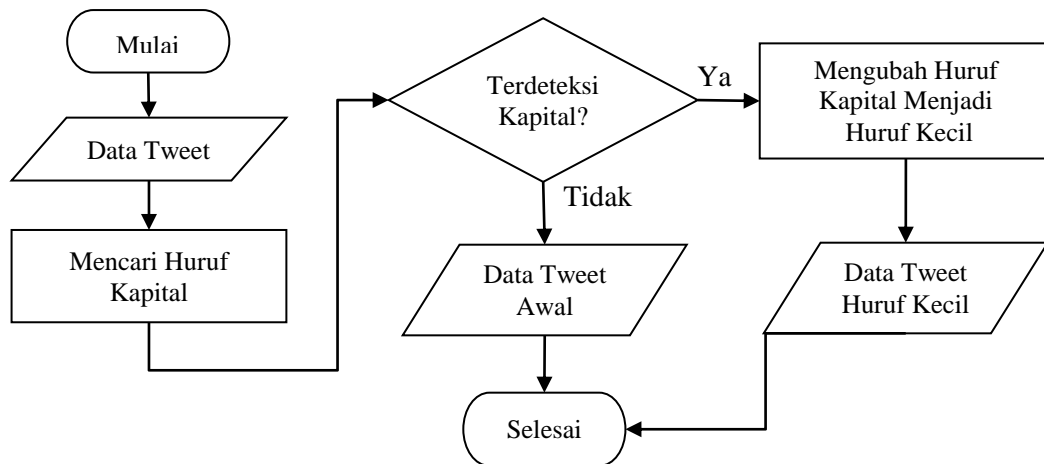
Adapun contoh hasil *cleaning* seperti terlihat pada Tabel 4.1 berikut ini :

Tabel 4.1 Proses *Cleaning*

Data Tweet		Hasil Cleaning
Di era kenormalan baru, mari utk tidak berbicara secara langsung/melalui telepon genggam saat di dlm KRL. Ayo... https://t.co/LCdvtbXsUPu	→	Di era kenormalan baru mari utk tidak berbicara secara langsung melalui telepon genggam saat di dlm KRL Ayo
Gegara New Normal, rakyat makin percaya Covid19 itu tidak ada.\n\nPemerintah tuding rakyat tidak disiplin lha ini wa.. https://t.co/BTXjsiA1oo	→	Gegara New Normal rakyat makin percaya Covid itu tidak ada Pemerintah tuding rakyat tidak disiplin lha ini wa
Tapi juga bisa sebabkan penularan massal plus banyak kematian.Tapi, terima kasih, kita jadi makin yakin isi kepala... https://t.co/EneBbTNFbr	→	Tapi juga bisa sebabkan penularan massal plus banyak kematian Tapi terima kasih kita jadi makin yakin isi kepala

b) *Case Folding*

Tahapan ini untuk mengkonversi atau mengubah huruf kapital kedalam huruf kecil (*lowercase*) pada semua data yang terdapat didalam dokumen.

Gambar 4.12 Alur Proses *Case Folding*

Berikut ini adalah langkah *Case folding* :

1. Mendeteksi dan membaca tiap baris data sebagai satu *tweet* apakah terdapat huruf kapital.
2. Ubah seluruh huruf kapital kedalam bentuk huruf kecil (*lowercase*)

Berikut contoh hasil *case folding* terhadap *tweet* seperti terlihat pada Tabel 4.2 berikut ini :

Tabel 4.2 Proses *Case Folding*

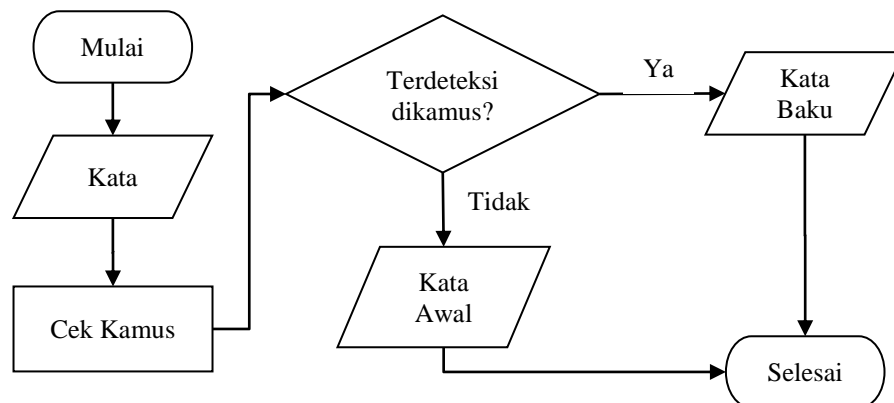
Data Tweet		Hasil Case Folding
Di era kenormalan baru mari utk tidak berbicara secara langsung melalui telepon genggam saat di dlm KRL Ayo	→	di era kenormalan baru mari utk tidak berbicara secara langsung melalui telepon genggam saat di dlm krl ayo
Gegara New Normal rakyat makin percaya Covid itu tidak ada Pemerintah tuding rakyat tidak disiplin lha ini wa	→	negara new normal rakyat makin percaya covid itu tidak ada pemerintah tuding rakyat tidak disiplin lha ini wa

Lanjutan Proses *Case Folding*

Data Tweet		Hasil Case Folding
Tapi juga bisa sebabkan penularan massal plus banyak kematian Tapi terima kasih kita jadi makin yakin isi kepala	→	tapi juga bisa sebabkan penularan massal plus banyak kematian tapi terima kasih kita jadi makin yakin isi kepala

c) *Normalisasi*

Tahap *normalisasi* untuk mengubah dan memperbaiki kata yang disingkat ke dalam kata yang memiliki arti sama berdasarkan KBBI agar menjadi informasi yang dapat diproses dengan mudah misalnya "*utk*" menjadi "*untuk*", "*yg*" menjadi "*yang*" dan sebagainya.



Gambar 4.13 Alur Proses Normalisasi

Langkah-langkah Normalisasi kata :

1. Mencari kata-kata yang disingkat pada data *tweet*
2. Bila ditemukan, maka diganti kata tersebut dengan kata normal kata yang memiliki arti sama yang berdasarkan KBBI misalnya "*tdk*" menjadi "*tidak*", "*utk*" menjadi "*untuk*" dan sebagainya.
3. Jika tidak ditemukan, maka sistem berhenti.

Berikut contoh hasil normalisasi terhadap tweet seperti terlihat pada Tabel 4.3 berikut :

Tabel 4.3 Proses Normalisasi

Data Tweet		Hasil Normalisasi
di era kenormalan baru mari utk tidak berbicara secara langsung melalui telepon genggam saat di dlm krl ayo	→	di era kenormalan baru mari untuk tidak berbicara secara langsung melalui telepon genggam saat di dalam kereta rel listrik ayo
negara new normal rakyat makin percaya covid itu tidak ada pemerintah tuding rakyat tidak disiplin lha ini wa	→	negara new normal rakyat makin percaya covid itu tidak ada pemerintah tuding rakyat tidak disiplin lah ini wa
tapi juga bisa sebabkan penularan massal plus banyak kematian tapi terima kasih kita jadi makin yakin isi kepala	→	tapi juga bisa sebabkan penularan massal plus banyak kematian tapi terima kasih kita jadi makin yakin isi kepala

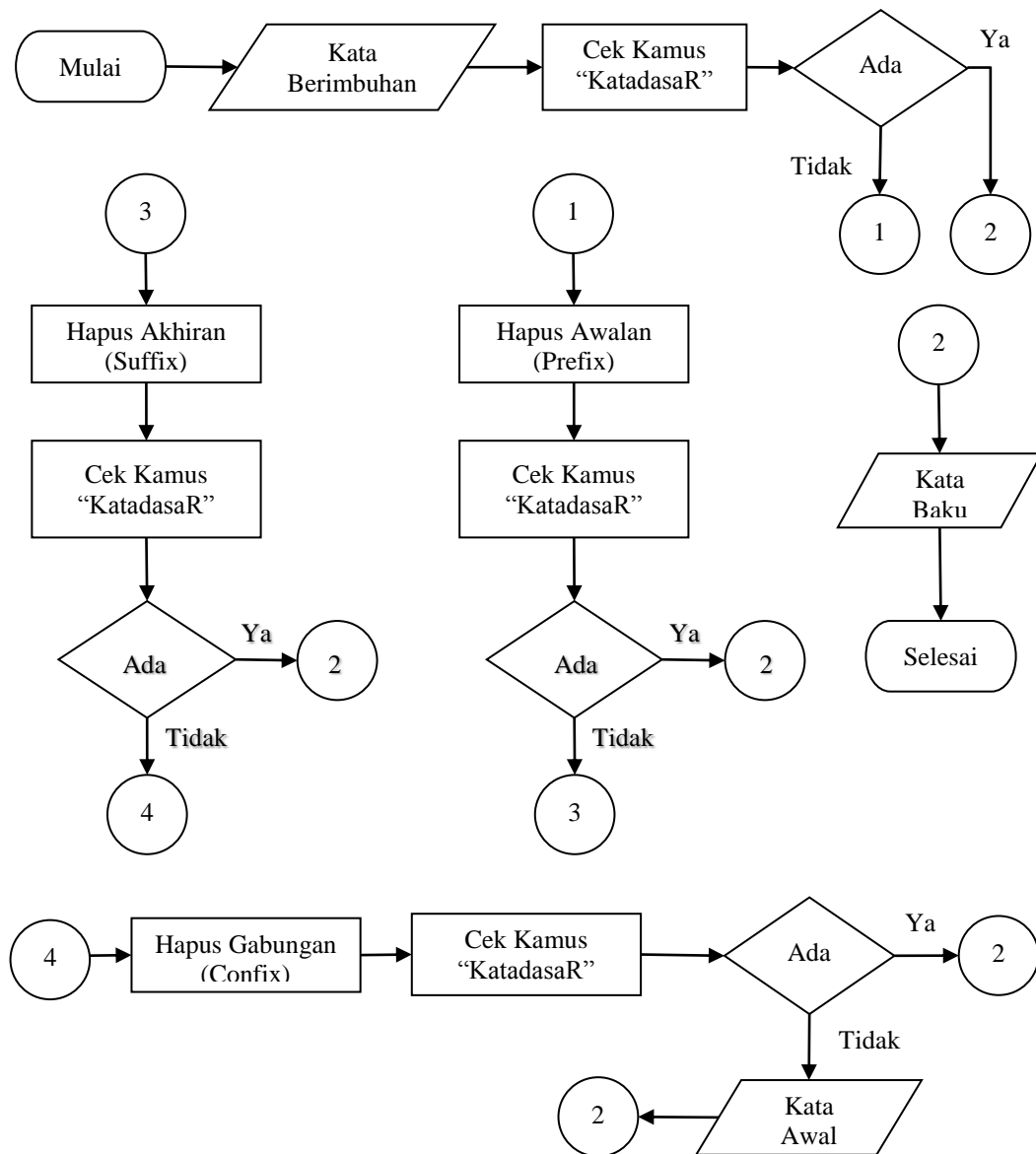
d) *Stemming*

Tahapan ini berfungsi untuk menghapuskan seluruh kata imbuhan yang terdapat pada data *tweet* seperti *prefix*, *suffix* dan *konfix*.

Langkah-langkah *Stemming* :

1. Mengecek atau mencari tiap kata dengan bantuan kamus pada daftar *package KatadasaR*
2. Bila kata ditemukan pada kamus katadasar maka diasumsikan sebagai *root word*.
3. Bila kata ditemukan pada kamus katadasar, maka akan dihapus imbuhan awal (*prefix*) pada kata sehingga menjadi kata baku dan proses diberhentikan, bila tidak ada maka dilanjutkan pada langkah ke-4.
4. Bila kata ditemukan dalam kamus katadasar maka akan dihapus imbuhan akhir (*suffix*) pada kata sehingga menjadi kata baku dan proses diberhentikan, bila tidak ada maka dilanjutkan pada langkah ke-5.

5. Bila kata ditemukan didalam kamus katadasar, akan dihapus gabungan awalan dan akhiran (*confix*) pada kata sehingga jadi kata baku dan prosesnya berhenti, bila tidak ada yang ditemukan kata tersebut akan dikembalikan sebagai kata asli sebelum dilakukan *stemming* dan proses berhenti.



Gambar 4.14 Alur Proses *Stemming*

(Sumber : Rizki, 2019)

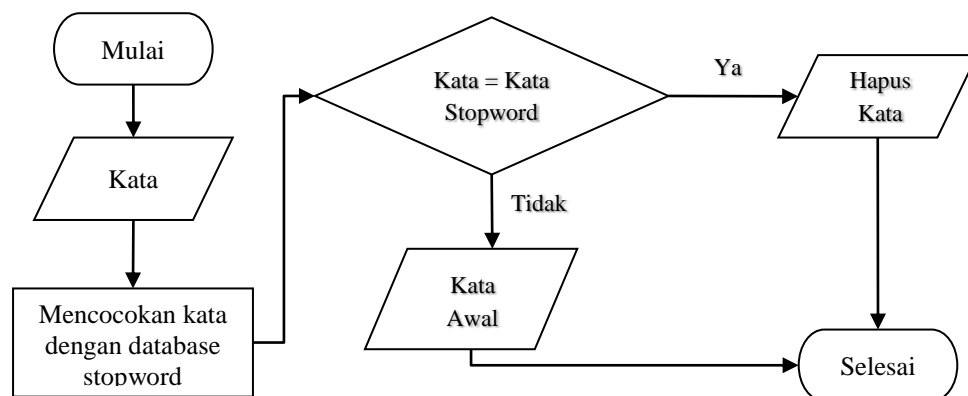
Berikut contoh hasil *stemming* terhadap seperti terlihat pada Tabel 4.4 berikut :

Tabel 4.4 Proses *Stemming*

Data Tweet		Hasil Stemming
di era kenormalan baru mari untuk tidak berbicara secara langsung melalui telepon genggam saat di dalam kereta rel listrik ayo	→	di era normal baru mari untuk tidak bicara cara langsung lalu telepon genggam saat di dalam kereta rel listrik ayo
negara new normal rakyat makin percaya covid itu tidak ada pemerintah tuding rakyat tidak disiplin lah ini wa	→	negara new normal rakyat makin percaya covid itu tidak ada pemerintah tuding rakyat tidak disiplin lah ini wa
tapi juga bisa sebabkan penularan massal plus banyak kematian tapi terima kasih kita jadi makin yakin isi kepala	→	tapi juga bisa sebab tular massal plus banyak mati tapi terima kasih kita jadi makin yakin isi kepala

e) *Filtering*

Tahapan ini untuk menghapuskan kata yang tidak mengandung makna ataupun kurang penting pada data. Pada tahapan ini menerapkan algoritma *stoplist* atau *stopword*. *Stopword* berisi kata-kata yang biasa dipakai tetapi tidak jelas dan dapat dibuang seperti "dari", "yang", "untuk", "dan", "di" dan sebagainya.



Gambar 4.15 Alur Proses *Filtering*

Langkah-langkah *Filtering* :

1. Membaca *token* kata-kata yang ada didaftar *stopwords* dilihat pada Lampiran I.
2. Mengecek tiap kata dan mencocokkan kata yang terdapat didalam daftar *stopwords*.
3. Menghapus kata jika yang ditemukan cocok dengan kata yang terdapat didalam daftar *stopwords*.

Berikut contoh hasil *filtering* terhadap tweet seperti terlihat pada Tabel 4.5 berikut :

Tabel 4.5 Proses *Filtering*

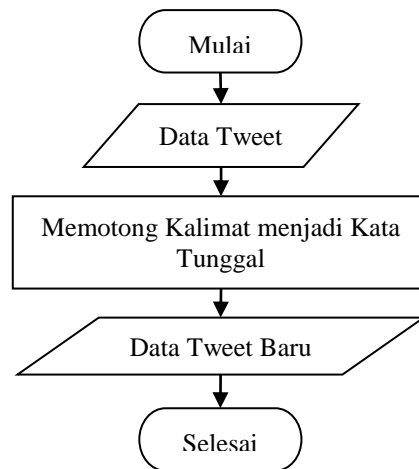
Data Tweet		Hasil Filtering
di era normal baru mari untuk tidak bicara cara langsung lalu telepon genggam saat di dalam kereta rel listrik ayo	→	era normal baru mari bicara cara langsung telepon genggam kereta rel listrik ayo
negara new normal rakyat makin percaya covid itu tidak ada pemerintah tuding rakyat tidak disiplin lah ini wa	→	negara new normal rakyat percaya covid pemerintah tuding rakyat disiplin
tapi juga bisa sebab tular massal plus banyak mati tapi terima kasih kita jadi makin yakin isi kepala	→	tular massal plus mati terima kasih yakin isi kepala

f) *Tokenizing*

Tahap *tokenizing* dilakukan untuk memisahkan *string* atau memecahkan kalimat jadi kata per kata agar mendapatkan kata yang memiliki nilai.

Berikut ini adalah langkah *Tokenizing* :

1. Membaca setiap baris teks data sebagai satuan *tweet*.
2. Memecah kalimat atau dokumen menjadi kata per kata yang biasa disebut *token*
3. Menyimpan setiap data *tweet* terdiri dari kata tunggal

Gambar 4.16 Alur Proses *Tokenizing*

Adapun contoh hasil *tokenizing* terhadap tweet seperti terlihat pada Tabel 4.6 berikut :

Tabel 4.6 Proses *Tokenizing*

Hasil Filtering		Hasil Tokenizing	
era normal baru mari untuk bicara cara langsung telepon genggam kereta rel listrik ayo	→	era normal baru mari untuk bicara cara	langsung telepon genggam kereta rel listrik ayo
negara new normal rakyatpercaya covidpemerintah tuding rakyat disiplin	→	negara new normal rakyat percaya	covid pemerintah tuding rakyat disiplin
tular massal plus mati terima kasih yakin isi kepala	→	tular massal plus mati terima	kasih yakin isi kepala

Setelah selesai melakukan proses *preprocessing*, data disimpan berbentuk *file.csv* dan setelah itu akan digunakan pada tahap proses selanjutnya.

4.1.3 Hasil Analisis Data

4.1.3.1 Pelabelan Kelas Sentimen

Sesudah data *tweet* melakukan tahapan *preprocessing*, kemudian data itu ditentukan kelas sentimennya yaitu positif, netral dan negatif. Pada pelabelan ini data akan diproses secara otomatis yaitu melakukan perhitungan nilai skor dengan kamus *lexicon*.

Penilaian data yang masuk pada kelompok sentimen positif atau sentimen negatif yang ditentukan menggunakan kamus *lexicon* yang berisi kumpulan-kumpulan kata positif dan negatif yang berbahasa Indonesia. Proses akan dilakukan secara otomatis menggunakan Aplikasi *Rstudio* dan hasil dari perhitungan skor bilasesuatu kalimat mempunyai skor >0 maka dikategorikan kelas positif, bila kalimat mempunyai skor <0 maka dikategorikan kelas negatif dan bila kalimat mempunyai skor $=0$ maka dikategorikan kelas netral.

Penulis mengambil contoh salah satu kalimat pada data tweet yaitu “*tular massal plus mati terima kasih jadi yakin isi kepala*” terdapat dua kata positif seperti “*terima kasih*” sedangkan pada kata negatif terdapat tiga kata seperti “*mati*”, “*tular*” dan “*isi kepala*”. Maka perhitungan skor sentimennya yaitu :

$$Skor = \left(\sum \text{kata positif} - \sum \text{kata negatif} \right)$$

$$Skor = (\text{jumlah kata positif}) - (\text{kata negatif}) = 2-3 = -1$$

Nilai tersebut menghasilkan skor nilai=-1 atau <0 , maka akan diidentifikasi sebagai kata sentimen negatif.

Tabel 4.7 Perhitungan Pelabelan Skor Sentimen

Opini	Kata Positif	Kata Negatif	Skor	Kelas Sentimen
tular massal plus mati terima kasih jadi yakin isi kepala	terima kasih	mati tular isi kepala	-2	negatif
	1	3		
era normal baru mari untuk bicara cara langsung telepon genggam kereta rel listrik ayo	normal	-	1	positif
	1	0		
negara new normal rakyat percaya covid pemerintah tuding rakyat disiplin	normal	tuding	0	netral
	1	1		

Pada penelitian ini terdapat tiga sentimen data yang terdiri dari sentimen positif, negatif dan netral. Namun data yang dipakai hanya sentimen positif dan negatif saja, disebabkan sentimen netral kurang masukan berguna dan bermanfaat. Jenis sentimen positif mengandung kata-kata seperti pujian, ucapan terima kasih ataupun hal yang membanggakan lainnya. Untuk jenis sentimen negatif mengandung kata-kata seperti kekecewaan, ungkapan ketidakpuasan ataupun penghinaan dan lain sebagainya, sedangkan jenis sentimen netral mengandung kata-kata seperti opini yang memiliki *statement* positif dan negatif yang seimbang dan ada juga opini yang tidak memiliki *statement* positif maupun negatif maka dalam jenis sentimen netral semacam ungkapan tanpa sentimen, iklan dan lain sebagainya.

4.1.3.2 Pembobotan *TF-IDF*

Tahap selanjutnya setelah melewati tahapan pelabelan kelas sentimen adalah tahap pembobotan *TF-IDF* (*Term Frequency-Inverse Document Frequency*), dimana pada tahapan ini menggunakan teknik perhitungan setiap pembobotan kata (*term*) yang ada didalam data dokumen dihitung dari setiap kata dan setiap kata kemudian akan dikalikan *idf*. Adapun langkah implementasi dari tahapan *tf-idf* sebagai berikut :

1. Menghitung nilai *tf* dari tiap-tiap kata yang ada didalam dokumen.
2. Menghitung nilai *tf* dari tiap-tiap kata yang ada didalam dokumen.
3. Hitung bobot pada data *tweet* dengan cara mengalikan nilai dari *tf* dan *idf*.
4. Mengulangi langkah-langkah dari 1 sampai 3 pada tiap dokumen.

Adapun contoh implementasi dari proses pembobotan kata sebagai berikut :

- a. Menghitung *TF* (*Term Frequency*)

Tabel 4.8 Contoh Proses Perhitungan *Term Frequency*

Kata (<i>Term</i>)	TF		
	D1	D2	D3
tular	1	0	0
massal	1	0	0
plus	1	0	0
mati	1	0	0
terima	1	0	0
kasih	1	0	0
jadi	1	0	0
yakin	1	0	0
kepala	1	0	0
era	0	1	0
normal	0	1	1
baru	0	1	0
mari	0	1	0

Lanjutan Contoh Proses Perhitungan *Term Frequency*

Kata (Term)	TF		
	D1	D2	D3
untuk	0	1	0
bicara	0	1	0
cara	0	1	0
langsung	0	1	0
telepon	0	1	0
genggam	0	1	0
kereta	0	1	0
rel	0	1	0
listrik	0	1	0
ayo	0	1	0
negara	0	0	1
new	0	0	1
rakyat	0	0	2
percaya	0	0	1
covid	0	0	1
pemerintah	0	0	1
tuding	0	0	1
disiplin	0	0	1

Kata :	tular	Term :	1
Kata :	massal	Term :	1
Kata :	plus	Term :	1
Kata :	mati	Term :	1
Kata :	terima	Term :	1
Kata :	kasih	Term :	1
Kata :	jadi	Term :	1
Kata :	yakin	Term :	1
Kata :	kepala	Term :	1
Kata :	era	Term :	1
Kata :	normal	Term :	2
Kata :	baru	Term :	1
Kata :	mari	Term :	1
Kata :	untuk	Term :	1
Kata :	bicara	Term :	1
Kata :	cara	Term :	1
Kata :	langsung	Term :	1
Kata :	telepon	Term :	1
Kata :	genggam	Term :	1
Kata :	kereta	Term :	1
Kata :	rel	Term :	1
Kata :	listrik	Term :	1
Kata :	ayo	Term :	1
Kata :	gegara	Term :	1
Kata :	new	Term :	1
Kata :	rakyat	Term :	2
Kata :	percaya	Term :	1
Kata :	covid	Term :	1
Kata :	pemerintah	Term :	1
Kata :	tuding	Term :	1
Kata :	disiplin	Term :	1

Gambar 4.17 Tampilan Hasil *Term Frequency*

Term frequency melakukan perhitungan dengan ditambahkan jumlah kata (*term*) yang muncul pada tiap dokumen. *TF* berfungsi buat mencermati apakah sesuatu kata terdapat ataupun tidak pada tiap dokumen, bila kata tersebut terdapat disuatu dokumen maka diberikan nilai satu dan bertambah bila disuatu dokumen mempunyai lebih dari satu kata yang muncul.

b) Menghitung *DF* (*document frequency*)

Tabel 4.9 Contoh Proses Perhitungan *Document Frequency*

Kata (<i>Term</i>)	TF			DF
	D1	D2	D3	
tular	1	0	0	1
massal	1	0	0	1
plus	1	0	0	1
mati	1	0	0	1
terima	1	0	0	1
kasih	1	0	0	1
jadi	1	0	0	1

Lanjutan Contoh Proses Perhitungan *Document Frequency*

Kata (<i>Term</i>)	TF			DF
	D1	D2	D3	
yakin	1	0	0	1
kepala	1	0	0	1
era	0	1	0	1
normal	0	1	1	2
baru	0	1	0	1
mari	0	1	0	1
untuk	0	1	0	1
bicara	0	1	0	1
cara	0	1	0	1
langsung	0	1	0	1
telepon	0	1	0	1
genggam	0	1	0	1
kereta	0	1	0	1
rel	0	1	0	1
listrik	0	1	0	1
ayo	0	1	0	1
negara	0	0	1	1
new	0	0	1	1
rakyat	0	0	2	1
percaya	0	0	1	1
covid	0	0	1	1
pemerintah	0	0	1	1
tuding	0	0	1	1
disiplin	0	0	1	1

Pada proses perhitungan *df* melakukan perhitungan jumlah dokumen yang memiliki kata (*term*). Dimana bila sesuatu kata terdapat disuatu dokumen maka *df* akan ditambahkan satu hingga dokumen terakhir buat mengenali jumlah semua *df*.

c. Menghitung *IDF* (*Inverse Document Frequency*)Tabel 4.10 Contoh Proses Perhitungan *Inverse Document Frequency*

Kata (<i>Term</i>)	TF			DF	N/DF	IDF(log N/DF)
	D1	D2	D3			
tular	1	0	0	1	3	0.477121255
massal	1	0	0	1	3	0.477121255
plus	1	0	0	1	3	0.477121255
mati	1	0	0	1	3	0.477121255
terima	1	0	0	1	3	0.477121255
kasih	1	0	0	1	3	0.477121255
jadi	1	0	0	1	3	0.477121255
yakin	1	0	0	1	3	0.477121255
kepala	1	0	0	1	3	0.477121255
era	0	1	0	1	3	0.477121255
normal	0	1	1	2	1.5	0.176091259
baru	0	1	0	1	3	0.477121255
mari	0	1	0	1	3	0.477121255
untuk	0	1	0	1	3	0.477121255
bicara	0	1	0	1	3	0.477121255
cara	0	1	0	1	3	0.477121255
langsung	0	1	0	1	3	0.477121255
telepon	0	1	0	1	3	0.477121255
genggam	0	1	0	1	3	0.477121255
kereta	0	1	0	1	3	0.477121255
rel	0	1	0	1	3	0.477121255
listrik	0	1	0	1	3	0.477121255
ayo	0	1	0	1	3	0.477121255

Lanjutan Contoh Perhitungan Bobot (*Weight*)

W		
D1	D2	D3
0.477121255	0	0
0.477121255	0	0
0.477121255	0	0
0	0.477121255	0
0	0.176091259	0.176091259
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0.477121255	0
0	0	0.477121255
0	0	0.477121255
0	0	0.954242509
0	0	0.477121255
0	0	0.477121255
0	0	0.477121255
0	0	0.477121255

Selanjutnya melakukan tahapan untuk menghitung bobot pada tiap kata dengan rumus sebagai berikut :

$$W_{td} = TF_{td} * IDF_t$$

Pada perhitungan bobot (*Weight*) dilakukan perhitungan dokumen ke-*d* terhadap kata (*term*). Sedangkan *tf* merupakan jumlah kemunculan *term* (*t*) dalam dokumen (*d*). *IDF* diperoleh dari hasil perhitungan proses sebelumnya. Hasil dari *term frequency* dikalikan dengan hasil *IDF* akan memperoleh setiap bobot dari masing-masing kata.

4.1.3.3 Klasifikasi *K-Nearest Neighbor* (*K-NN*)

Setelah melakukan pembobotan dari masing-masing *tweet* dan *vector* dari masing-masing *tweet*, maka akan dilakukan klasifikasi menggunakan metode *K-Nearest Neighbor* (*K-NN*). Pada metode ini mengambil *k*-tetangga terdekat dengan menggunakan jarak *Euclidean*. Kemudian akan masuk ketahap dengan menghitung akurasi menggunakan metode algoritma *K-Nearest Neighbor* (*K-NN*). Model ini merupakan pengetahuan yang akan digunakan untuk memprediksi kelas data yang belum pernah ada, sehingga semakin besar data latih yang digunakan, maka akan semakin baik *machine learning* dalam memahami pola suatu data.

Langkah-langkah dalam proses *K-Nearest Neighbor* (*K-NN*) :

1. Pada klasifikasi algoritma *K-NN* menggunakan *IBk* dalam weka untuk menentukan dataset opini masyarakat terhadap kebijakan pemerintah tentang new normal.
2. Menentukan nilai parameter *k* pada *K-NN*
3. Memilih opsi pengujian pada *K-NN* meliputi *use training set*, *k-fold cross-validation* dan *percentage split*.
4. Hitung jarak antara data *training* dan data *testing* memakai *Euclidean Distance*.
5. Menghasilkan *classifier* dan *confusion matrix*.
6. Menghitung nilai *accuracy*, *precision*, *recall* dan *f-measure*.

4.1.3.4 Confusion Matriks

Pada penelitian ini akan mengevaluasi hasil dan mengukur kinerja suatu metode klasifikasi dengan menggunakan *confusion matrix*. Setelah melakukan perhitungan *K-Nearest Neighbor* maka akan dilakukan pengujian akurasi menggunakan *confusion matrix* untuk mengetahui keakuratan hasil klasifikasi dan untuk mengetahui seberapa besar keberhasilan sistem dalam melakukan klasifikasi. Dalam *confusion matrix* akan dihitung *accuracy*, *precision*, *recall* dan *f-measure*.

Tabel 4.12 Contoh Perhitungan *Confusion Matrix*

Prediksi	Nilai	
	Positif	Negatif
Positif	2	0
Negatif	0	1

Maka perhitungannya :

$$Accuracy = \frac{2 + 1}{2 + 0 + 0 + 1} \times 100\% = 100\%$$

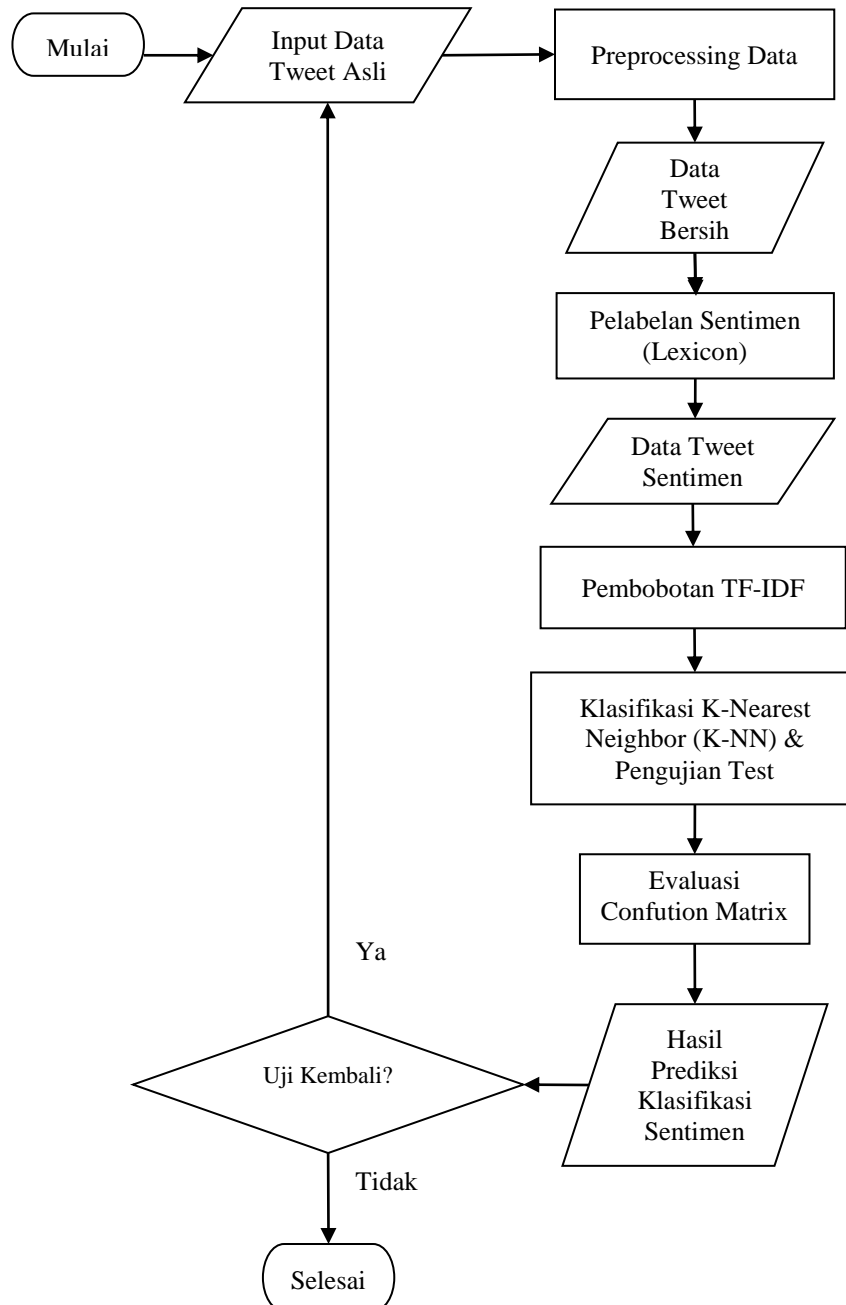
$$Precision = \frac{2}{2 + 0} \times 100\% = 100\%$$

$$Recall = \frac{2}{2 + 0} \times 100\% = 100\%$$

$$F - measure = 2 \times \frac{100 \times 100}{100 + 100} = 100\%$$

Hasil *accuracy*, *precision*, *recall* dan *f-measure* yang telah di dapatkan yaitu 100%.

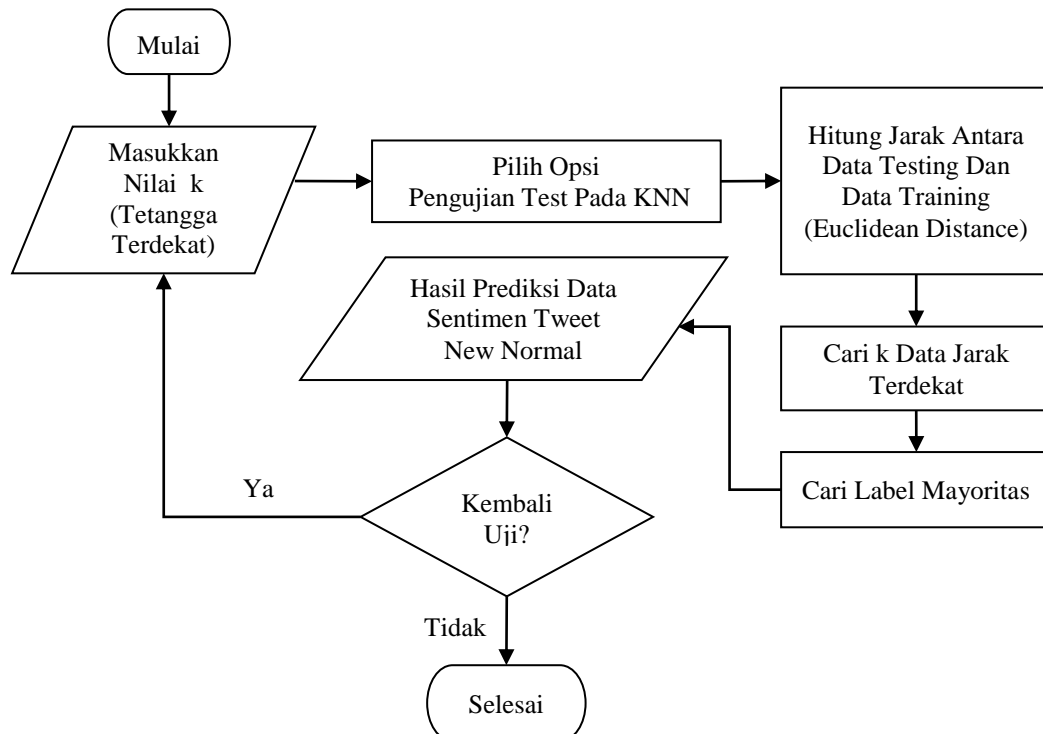
4.1.4 Perancangan Sistem



Gambar 4.18 *Flowchart* Sistem

Pada Gambar 4.18 merupakan alur sistem *flowchart* dalam melakukan analisis sentimen terhadap *new normal* di era *covid-19* menggunakan algoritma *K-Nearest Neighbor (K-NN)*. Dimulai dari menginput data *tweet* asli, kemudian proses *preprocessing* data. Setelah didapatkan hasil data *tweet* bersih, selanjutnya data *tweet* akan melewati tahap pelabelan sentimen yang dimana data akan diberikan kelas sentimennya berupa positif dan negatif. Kemudian data *tweet* sentimen akan diubah menjadi *vector* dengan dihitung bobotnya menggunakan *TF-IDF*.

Setelah didapat data *tweet vector* maka data akan diklasifikasi menggunakan algoritma *K-Nearest Neighbor (K-NN)* dengan pengujian yaitu *use training-set*, *10-fold cross-validation* dan *80% percentage-split* untuk membandingkan pengujian mana yang mendapatkan hasil akurasi yang terbaik. Selanjutnya data akan dievaluasi menggunakan *confution matrix* untuk mengetahui keakuratan hasil klasifikasi dan mendapatkan gambaran seberapa besar sistem model berhasil dalam melakukan klasifikasi.



Gambar 4.19 Flowchart Algoritma *K-Nearest Neighbor (K-NN)*

Pada Gambar 4.19 merupakan alur *flowchart* algoritma *K-Nearest Neighbor (K-NN)* dalam melakukan klasifikasi pada data *tweet* sentimen terhadap *new normal* di era *covid-19*. Dimulai dengan memasukkan nilai *k* tetangga terdekat, kemudian pilih opsi pengujian pada *KNN*. Selanjutnya dihitung jarak pada data *testing* dan data *training* memakai *euclidean distance*, setelahnya mencari data terdekat dan mencari label mayoritas. Kemudian didapatkan hasil prediksi data sentimen terhadap *new normal* di era *covid-19*.

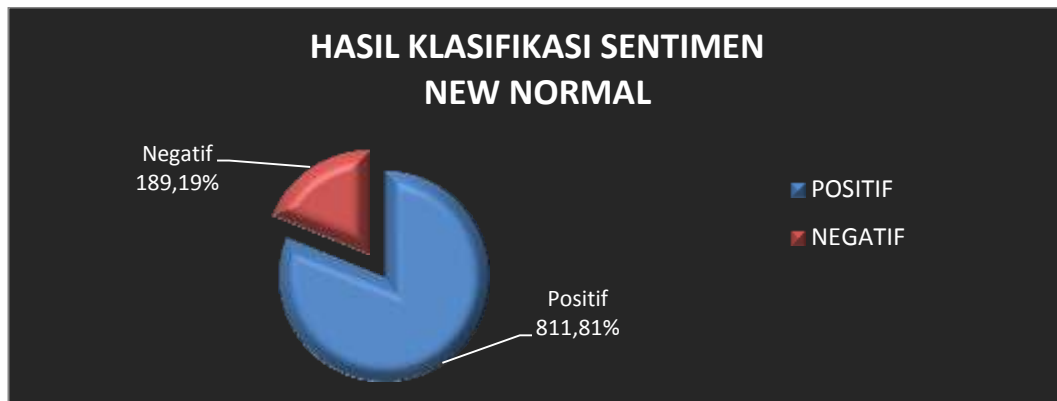
4.2 Hasil

Pada penelitian ini, perhitungan pelabelan sentimen telah mendapatkan hasil perbandingan jumlah data kelas sentimen seperti terlihat pada Tabel 4.13 :

Tabel 4.13 Perbandingan Jumlah Data Kelas Sentimen

Kelas Sentimen Sementara	Jumlah
Positif	568
Negatif	189
Netral	243
Total	1000

Pada Tabel 4.13 terdapat kelas sentimen yaitu positif, negatif dan juga netral, namun data yang dipakai pada penelitian ini cuma memakai sentimen positif dan negatif saja. Selanjutnya akan melakukan proses reduksi pada kelas sentimen netral dimasukan kedalam sentimen positif yang dibuat secara manual. Untuk hasil pelabelan kelas sentimen terhadap *new normal* dapat dilihat sebagai berikut :



Gambar 4.20 Hasil Kelas Sentimen Terhadap New Normal

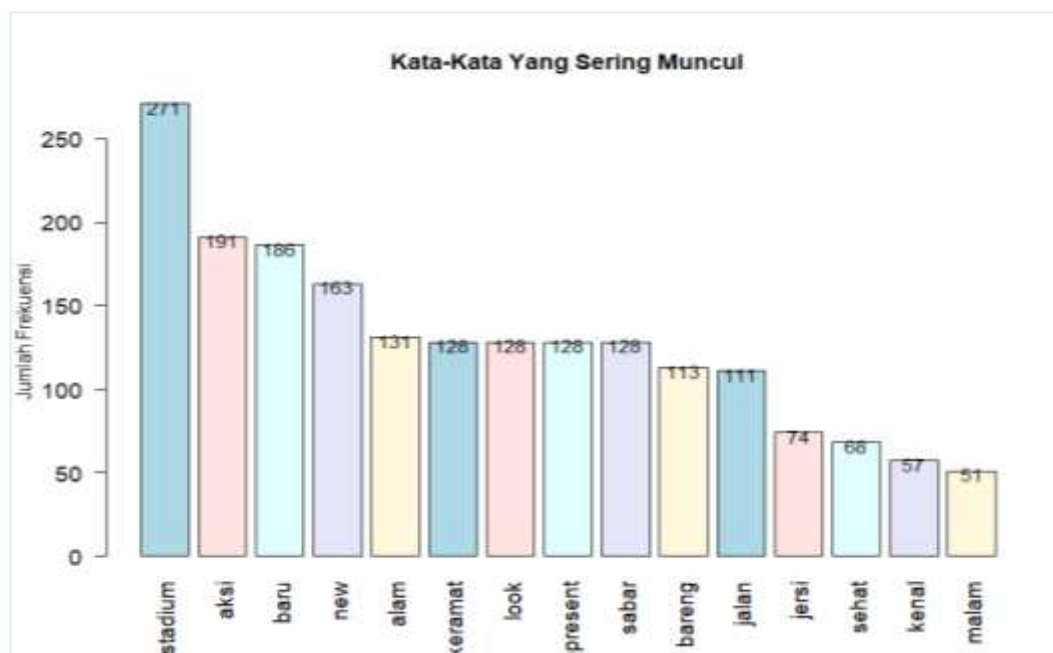
Berdasarkan Gambar 4.20 diatas terlihat jumlah pada data kelas sentimen terhadap *new normal* yang terdiri dari 1000 data opini. Dengan persentase pada kelas sentimen positif mendapatkan 81% dan 19% pada kelas sentimen negatif. Dalam hal ini opini masyarakat terhadap kebijakan *new normal* mendapatkan kelas sentimen positif lebih unggul dibandingkan kelas sentimen negatif. Dengan kata lain masyarakat Indonesia dalam memberi opini atau tanggapan sangat baik terhadap kebijakan *new normal* di era *covid-19*.

4.2.1 Visualisasi

Term Documen Matrix (TDM) merupakan bentuk matematika yang menguraikan frekuensi kata yang selalu terdapat didalam *corpus*. Pada penelitian ini mendapatkan hasil frekuensi kata dari 1000 *tweet* sebesar 1735 kata yang seperti terlihat pada Gambar 4.21.

	word	freq
stadium	stadium	271
aksi	aksi	191
baru	baru	186
new	new	163
alam	alam	131
keramat	keramat	128
look	look	128
represent	represent	128
sabar	sabar	128
bareng	bareng	113
jalan	jalan	111
jersi	jersi	74
sehat	sehat	68
kenal	kenal	57
malam	malam	51
normal	normal	49
semangat	semangat	48
kembali	kembali	47
tetap	tetap	44
masker	masker	41
siap	siap	40
tahan	tahan	38
owner	owner	37

Showing 1 to 25 of 1,735 entries; 2 total

Gambar 4.21 Frekuensi *TDM*

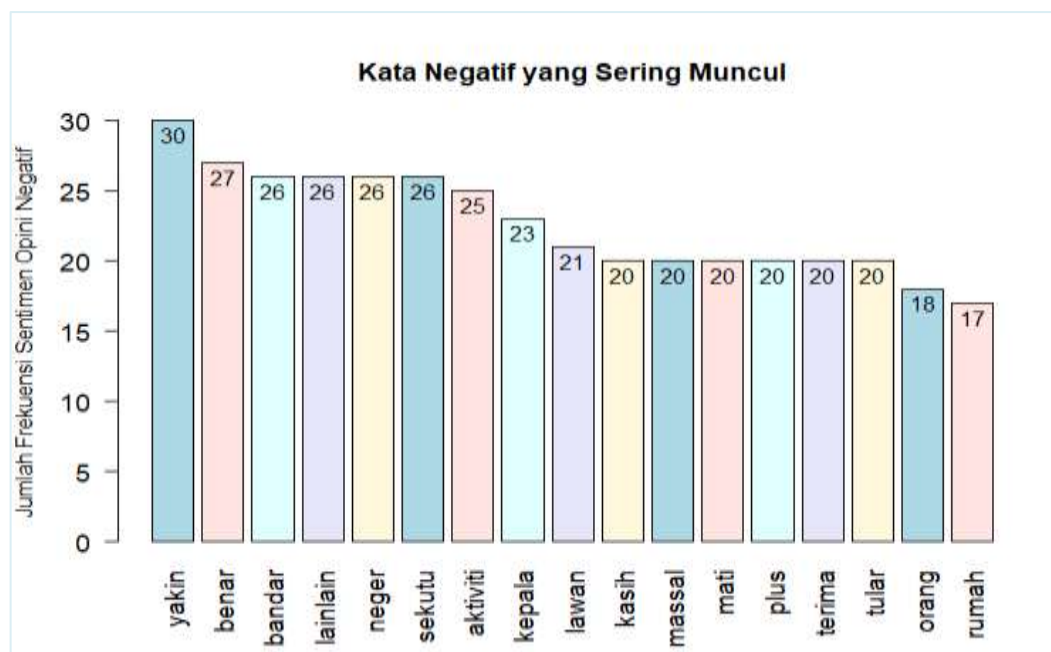
Gambar 4.22 Kata Yang Sering Muncul Pada Keseluruhan Data

Dapat dilihat pada Gambar 4.22 pada hasil klasifikasi opini keseluruhan data sebanyak 1000 data yang telah didapatkan beberapa kata yang paling sering muncul yaitu kata "*stadium*" sebanyak 271 kali, kata "*aksi*" sebanyak 191 kali,

kata "*baru*" sebanyak 186 kali, kata "*new*" sebanyak 163 kali, kata "*sabar*" sebanyak 128 kali, kata "*sehat*" sebanyak 68 kali dan seterusnya.

a) Opini Negatif

Pelabelan yang dilakukan menghasilkan data sentimen negatif dengan menggunakan kamus *lexicon* secara otomatis memakai aplikasi *R*. Hasil pelabelan sentimen pada kelas opini negatif terhadap *new normal* yaitu 189 sentimen dari 1000 data opini. Adapun hasil visualisasi untuk keseluruhan data *tweet* opini bersentimen negatif sebagai berikut :



Gambar 4.23 Kata Yang Sering Muncul Pada Data Kelas Negatif

Dapat dilihat pada Gambar 4.23 kata-kata yang terdapat pada sentimen negatif yang digunakan masyarakat dalam memberikan opini diantaranya adalah kata "*yakin*" sebanyak 30 kali, kata "*benar*" sebanyak 27 kali, kata "*bandar*" sebanyak 26 kali, kata "*lawan*" sebanyak 21 kali, kata "*mati*" sebanyak 20 kali dan seterusnya. Kata-kata yang muncul seperti pada Gambar 4.22 merupakan kata yang memiliki sentimen negatif yang telah menjadi bahasan yang negatif yang paling banyak muncul yang dibicarakan masyarakat pengguna twitter terhadap

new normal. Kumpulan kata-kata negatif yang sering muncul tersebut juga dapat ditampilkan dalam bentuk *wordcloud* seperti terlihat pada Gambar 4.24 berikut :

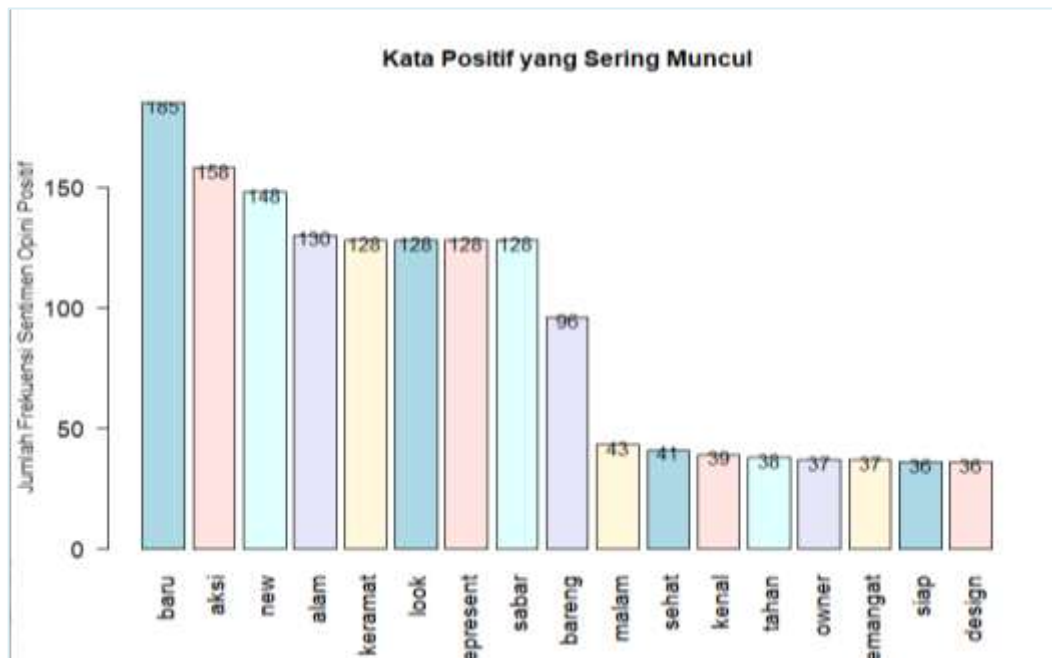


Gambar 4.24 *Wordcloud* Kelas Sentimen Negatif

Berdasarkan Gambar 4.24 visualisasi *wordcloud* dapat dilihat dengan lebih jelas kata-kata sentimen negatif yang sering digunakan oleh masyarakat pengguna twitter dalam memberikan opini atau tanggapan terhadap *new normal*. Semakin besar ukuran pada kata yang ada pada *wordcloud* maka semakin tinggi pula frekuensi atau jumlah kata tersebut, artinya masyarakat pengguna twitter sering menggunakan kata tersebut sebagai topik pembahasan mengenai penilaian negatif terhadap kebijakan pemerintah tentang *new normal* di era *covid-19*.

b) Opini Positif

Data opini positif yang digunakan merupakan data hasil pelabelan yang telah dilakukan dengan penggunaan kamus *lexicon* secara otomatis menggunakan aplikasi *R*. Berdasarkan hasil pelabelan, data opini bersentimen positif terhadap *new normal* lebih banyak dibandingkan dengan jumlah opini bersentimen negatif. Data sentimen positif berjumlah 568 dari 1000 data opini. Hal ini menunjukkan bahwa banyak masyarakat mempunyai persepsi yang terhadap kebijakan pemerintah tentang *new normal*. Adapun hasil visualisasi untuk keseluruhan data *tweet* opini bersentimen positif sebagai berikut :



Gambar 4.25 Kata Yang Sering Muncul Pada Data Kelas Positif

Dapat dilihat pada Gambar 4.25 kata-kata positif yang sering digunakan masyarakat dalam memberikan opini diantaranya adalah kata "baru" sebanyak 185 kali, kata "aksi" sebanyak 158 kali, kata "new" sebanyak 148 kali, kata, "sabar" sebanyak 128 kali, kata "sehat" sebanyak 41 kali dan seterusnya. Kata-kata yang muncul seperti pada Gambar 4.24 merupakan kata yang memiliki sentimen positif serta kata yang telah menjadi topik pembicaraan atau bahasan yang positif yang paling banyak muncul yang dibicarakan masyarakat pengguna twitter terhadap *new normal*. Kumpulan kata-kata positif yang sering muncul tersebut juga dapat ditampilkan dalam bentuk *wordcloud* seperti terlihat pada Gambar 4.26 berikut :

Dari Gambar 4.28 menunjukkan bahwa hasil sistem klasifikasi *K-NN* pada pengujian *use training set* dalam mengelompokkan klasifikasi positif dan negatif yang berhasil teridentifikasi. Jumlah kata positif yang benar yaitu 811, sedangkan negatif berjumlah 189. Pada pengujian *use training set* merupakan pengujian atau pengetesan dengan menggunakan data *training* itu sendiri.

```
Time taken to build model: 0 seconds

=== Summarized cross-validation ===
=== Summary ===

Correctly Classified Instances      809      80.9 %
Incorrectly Classified Instances    191      19.1 %
Kappa statistic                    0.7187
Mean absolute error                 0.0739
Root mean squared error             0.2643
Relative absolute error             10.9416 %
Root relative squared error         82.0668 %
Total Number of Instances          1000

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	AUC Area	Class
	0.986	0.001	0.915	0.998	0.956	0.746	0.913	0.937	Positif
	0.429	0.002	0.983	0.419	0.740	0.746	0.913	0.764	Negatif
Weighted Avg.	0.926	0.309	0.931	0.928	0.919	0.746	0.913	0.903	

```

=== Confusion Matrix ===
  a  b  <-- classified as
 00 11  a = Positif
 72 117  b = Negatif

```

Gambar 4.29 Hasil Klasifikasi *K-NN* dengan *10-Fold Cross-Validation*

Pada pengujian *10-fold cross validation* melakukan pengetesan dengan menggunakan pilihan banyaknya *fold*. Dari Gambar 4.29 menunjukkan bahwa hasil sistem klasifikasi *K-NN* pada pengujian *10-fold cross validation* dalam melakukan klasifikasi positif sebanyak 809 dan untuk label negatif sebanyak 117.

```
=== Evaluation on test split ===

Time taken to test model on test split: 0.12 seconds

=== Summary ===

Correctly Classified Instances      189      94.5 %
Incorrectly Classified Instances     11      5.5 %
Kappa statistic                    0.7246
Mean absolute error                 0.0039
Root mean squared error             0.2249
Relative absolute error             16.8803 %
Root relative squared error         84.3827 %
Total Number of Instances          200

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	AUC Area	Class
	1.000	0.000	0.940	1.000	0.980	0.755	0.932	0.987	Positif
	0.607	0.000	1.000	0.607	0.756	0.755	0.932	0.770	Negatif
Weighted Avg.	0.945	0.000	0.940	0.943	0.939	0.755	0.932	0.931	

```

=== Confusion Matrix ===
  a  b  <-- classified as
 172  0  a = Positif
 11  17  b = Negatif

```

Gambar 4.30 Hasil Klasifikasi *K-NN* dengan *80% Percentage Split*

Dari pengujian 80% *percentage split* melakukan pengetesan dengan $k\%$ dari data, dimana k merupakan proporsi dari *dataset* yang digunakan untuk data *training*. Persentase pada pengujian 80% untuk data latih (*training*) dan 20% untuk data uji (*testing*). Dari Gambar 4.30 menunjukkan bahwa hasil sistem klasifikasi *K-NN* pada pengujian 80% *percentage split* dalam melakukan klasifikasi positif sebanyak 172 dan untuk label negatif sebanyak 17.

Tabel 4.14 Perbandingan Keseluruhan Evaluasi Dari Beberapa Pengujian *Test* Menggunakan Algoritma *K-Nearest Neighbor (K-NN)*

Data Set	Algoritma	Mode Test	Correctly Classified Instances		Incorrectly Classified Instances		Mean Absolute Error
			(n)	%	(n)	%	
New Normal	K-Nearest Neighbor (K-NN) Lazy- IBK	Use Training Set	1000	100%	0	0	0,0005
		10 Fold Cross Validation	926	92,6%	74	7,4%	0,0735
		80% Percentage Split	189	94,5%	11	5,5%	0,0539

Perbandingan dari keseluruhan evaluasi dari beberapa pengujian menggunakan algoritma *K-Nearest Neighbor (K-NN)* dapat dilihat pada Tabel 4.14. Informasi yang terdapat terdiri dari pengujian yang digunakan untuk dataset yang terdiri dari pengujian *use training set*, *10 fold cross validation* dan *80% percentage split*.

Informasi ukuran akurasi juga bisa didapatkan dari Tabel 4.14 pada kolom *correctly classified instance* dan *incorrectly classified instance*. *Mean absolute*

error juga merupakan kolom yang menyediakan informasi rata-rata *error* yang adapada beberapa jenis algoritma ketika membangun model klasifikasi untuk algoritma *K-Nearest Neighbor (K-NN)* dalam Tabel 4.14.

Pada pengujian *use training set* yang ada pada Tabel 4.14 pada penelitian ini nilai *correctly classified instance* yang didapatkan 100% yang terdiri dari 1000 *instance* yang terklasifikasi benar dan memiliki nilai *mean absolute error* sebesar 0,0005 dikarenakan pengujian dilakukan menggunakan data *training* itu sendiri.

Pada pengujian *10-fold cross-validation* mendapatkan hasil *correctly classified instance* yaitu sebesar 92,6% yang terdiri dari 926 *instance* yang terklasifikasi benar dan *incorrectly classified instance* sebesar 7,4%, data yang terklasifikasi salah sebesar 74 *instance* dari total semua *dataset* berjumlah 1000 *instance*, mencapai nilai *mean absolute error* sebesar 0,0735.

Pada 80% *percentage split* menggunakan pembagian nilai jumlah data latih dan data ujialah 80% untuk data latih dan 20% untuk data uji. Mendapatkan hasil *correctly classified instance* yaitu sebesar 94,5% yang terdiri dari 189 *instance* yang terklasifikasi benar dan *incorrectly classified instance* sebesar 5,5% yang dimana klasifikasi data salah hanya sebesar 11 *instance*, mencapai nilai *mean absolute error* sebesar 0,0539.

Tabel 4.15 Waktu Yang Dibutuhkan Untuk Membangun Model

Algoritma	Pengujian Test	New Normal Waktu (Detik)
K-Nearest Neighbor (K-NN) Lazy- IBK	Use Training Set	0,43
	10 Fold Cross Validation	0
	80% Percentage Split	0,12

Tabel 4.15 menampilkan informasi waktu yang diperlukan untuk membangun model pada pengujian menggunakan algoritma *K-Nearest Neighbor (K-NN)*. Satuan waktu yang digunakan yaitu satuan detik, yang dimana pengujian yang digunakan yaitu *use training set*, *10-fold cross-validation* dan *80% percentage split*.

Pada Tabel 4.15 dapat dilihat pada pengujian *use training set* untuk *dataset new normal* menghasilkan catatan waktu sebesar 0,43 detik untuk membuat model klasifikasi. Pada pengujian test 10 *fold cross validation* menghasilkan catatan waktu yang sangat cepat dari lainnya sebesar 0 detik dan 80% *percentage split* untuk membangun model klasifikasi untuk *dataset new normal* memiliki catatan waktu yaitu sebesar 0,12 detik.

4.2.3 Evaluasi Hasil

Evaluasi hasil didalam penelitian ini menggunakan *confusion matrix*. Dari *confusion matrix* akan digunakan untuk menghitung nilai *acuracy*, *precision*, *recall* dan *f-measure* berguna untuk membandingkan pengujian mana yang cukup baik untuk membangun model klasifikasi menggunakan metode algoritma *K-Nearest Neighbor (K-NN)*.

Tabel 4.16 *Confusion Matrix*

Prediksi	Use Training Set		10-Fold Cross Validation		80% Percentage Split	
	Positif	Negatif	Positif	Negatif	Positif	Negatif
Positif	811	0	809	2	172	0
Negatif	0	189	72	117	11	17

Dari data pada Tabel 4.16 *confusion matrix* maka *use training set* dapat dihitung *acuracy*, *precision*, *recall* dan *f-measure* sebagai berikut :

$$Acuracy = \frac{811 + 189}{811 + 0 + 0 + 189} \times 100\% = 100\%$$

$$Precision = \frac{811}{811 + 0} \times 100\% = 100\%$$

$$Recall = \frac{811}{811 + 0} \times 100\% = 100\%$$

$$F - measure = 2 \times \frac{100 \times 100}{100 + 100} = 100\%$$

Dari data pada Tabel 4.16 *confusion matrix* maka 10-fold cross-validation dapat dihitung *accuracy*, *precision*, *recall* dan *f-measure* sebagai berikut :

$$Accuracy = \frac{809 + 117}{809 + 72 + 2 + 117} \times 100\% = 92,6\%$$

$$Precision = \frac{809}{809 + 2} \times 100\% = 99,76\%$$

$$Recall = \frac{809}{809 + 72} \times 100\% = 91,83\%$$

$$F - measure = 2 \times \frac{99,76 \times 91,83}{99,76 + 91,83} = 95,63\%$$

Dari data pada Tabel 4.16 *confusion matrix* maka 80% percentage split dapat dihitung *accuracy*, *precision*, *recall* dan *f-measure* sebagai berikut :

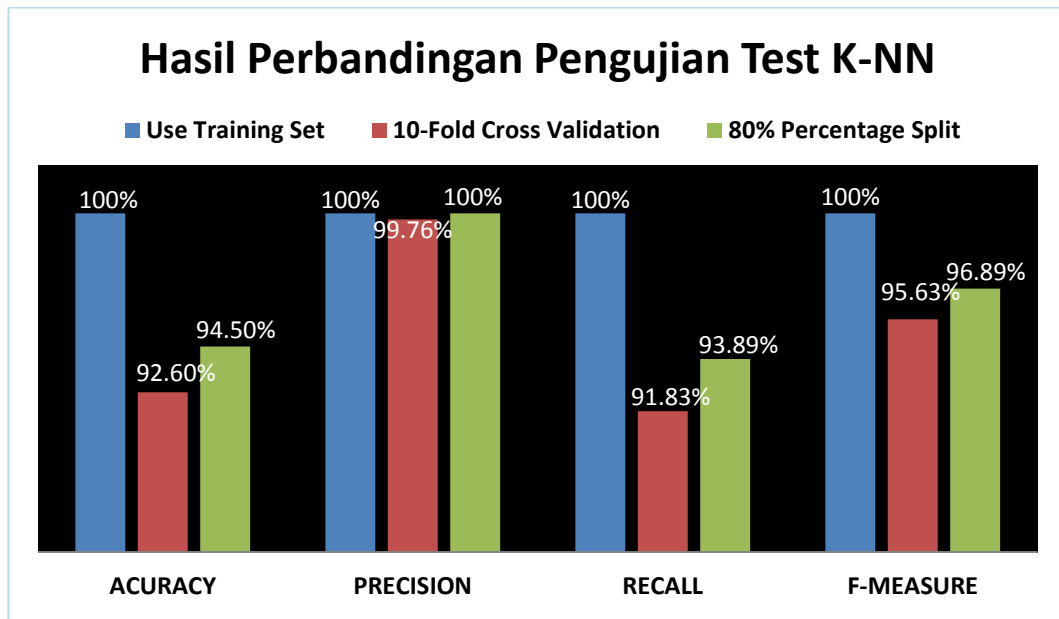
$$Accuracy = \frac{172 + 17}{172 + 11 + 0 + 17} \times 100\% = 94,5\%$$

$$Precision = \frac{172}{172 + 0} \times 100\% = 100\%$$

$$Recall = \frac{172}{172 + 11} \times 100\% = 93,98\%$$

$$F - measure = 2 \times \frac{100 \times 93,98}{100 + 93,98} = 96,89\%$$

Setelah mendapatkan hasil perbandingan pengujian *K-NN* dapat dilihat di gambar diagram pada Gambar 4.31.



Gambar 4.31 Hasil Perbandingan Pengujian *Test K-NN*

Dari hasil diatas yang didapat akurasi *use training set* mendapatkan hasil *accuracy* sebesar 100% , *precision* sebesar 100%, *recall* sebesar 100% dan *f-measure* sebesar 100%, *10-fold cross validation* yang didapat mendapatkan hasil *accuracy* sebesar 92,60%, *precision* sebesar 99,76% *recall* sebesar 91,83% dan *f-measure* sebesar 95,63%, dan yang terakhir 80% *percentage split* mendapatkan hasil *accuracy* sebesar 94,50% , *precision* sebesar 100%, *recall* sebesar 93,89% dan *f-measure* sebesar 96,89%. Metode klasifikasi algoritma *K-Nearest Neighbor* (*K-NN*) menggunakan pengujian 80% *percentage split* sangat baik dalam pengujian klasifikasi memiliki *accuracy*, *precision*, *recall* dan *f-measure* lebih besar dibandingkan dengan pengujian *10-fold cross validation*.

4.2.4 Penerapan

Penerapan sistem pada penelitian yaitu untuk mengetahui bagaimana pandangan atau sentimen masyarakat Indonesia terhadap kebijakan *new normal* di era *covid-19* pada media sosial twitter. Dengan menerapkan metode klasifikasi algoritma *K-Nearest Neighbor* (*KNN*) dan pengujian tingkat akurasi. Penerapan sistem ini juga diharapkan dapat membantu dan mempermudah mahasiswa maupun pelajar dalam menganalisis sebuah sentimen dalam bentuk informasi yang bisa disajikan dalam bentuk artikel-artikel, jurnal ataupun media massa.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari proses analisis sentimen pada penelitian ini diperoleh beberapa kesimpulan sebagai berikut ini :

1. Analisis sentimen terhadap *new normal* di era *covid-19* menggunakan algoritma *K-Nearest Neighbor (K-NN)* mendapatkan hasil penerapan klasifikasinya pada data opini masyarakat terhadap *new normal* pada media sosial twitter menggunakan pelabelan kelas kamus *lexicon* bahwa kelas sentimen positif lebih unggul berjumlah 811 dibandingkan kelas sentimen negatif berjumlah 189. Dengan kata lain masyarakat Indonesia dalam memberi opini atau tanggapan sangat baik terhadap kebijakan *new normal* di era *covid-19*.
2. Dari penelitian ini juga menunjukkan metode klasifikasi algoritma *K-Nearest Neighbor (K-NN)* dengan membandingkan pengujian sangat baik dengan nilai $k=1$ untuk pengujian *use training set* mendapatkan hasil *accuracy* sebesar 100%, *precision* sebesar 100%, *recall* sebesar 100% dan *f-measure* sebesar 100%, hasil *correctly classified instance* yaitu sebesar 100% dimana sistem pengujian *use training set* dapat melakukan klasifikasi dengan sebanyak 1000 *instance* yang terklasifikasi benar dan memiliki nilai *mean absolute error* sebesar 0,0005 dikarenakan pengujian dilakukan menggunakan data *training* itu sendiri. Sedangkan pada pengujian *10-fold cross validation* yang didapat mendapatkan hasil *accuracy* sebesar 92,60%, *precision* sebesar 99,76% *recall* sebesar 91,83% dan *f-measure* sebesar 95,63%, hasil *correctly classified instance* yaitu sebesar 92,6% yang terdiri dari 926 *instance* yang terklasifikasi benar dan *incorrectly classified instance* sebesar 7,4% yang dimana klasifikasi data salah hanya sebesar 74 *instance* dari total keseluruhan *dataset* sebanyak 1000 *instance*, mencapai nilai *mean absolute error* sebesar 0,0735. Dan yang terakhir 80% *percentage split* mendapatkan hasil *accuracy* sebesar 94,50% , *precision* sebesar 100%, *recall* sebesar 93,89% dan *f-measure* sebesar 96,89%,

hasil *correctly classified instance* yaitu sebesar 94,5% yang terdiri dari 189 *instance* yang terklasifikasi benar dan *incorrectly classified instance* sebesar 5,5% yang dimana klasifikasi data salah hanya sebesar 11 *instance*, mencapai nilai *mean absolute error* sebesar 0,0539.

3. Dari hasil penelitian ini penulis menyimpulkan bahwa klasifikasi algoritma *K-Nearest Neighbor (K-NN)* menggunakan pengujian 80% *percentage split* merupakan pengujian yang lebih optimal dibandingkan dengan pengujian lainnya dikarenakan juga memiliki nilai akurasi yang tinggi.

5.2 Saran

Berikut ini adalah saran penulis dalam penelitian ini :

1. Penambahan kamus bahasa Indonesia dan kamus bahasa *slang* dapat ditambahkan untuk penambahan dari kata dasar secara otomatis.
2. Dalam penelitian ini data opini hanya dibataskan untuk opini bahasa Indonesia, maka dari itu diperlukan untuk mengembangkan kembali menggunakan terjemahan bahasa asing.

DAFTAR PUSTAKA

- D'Monte, L., (2009), *Swine Flu's Tweet Tweet Causes Online Flutter*.https://www.business-standard.com/article/technology/swine-flu-s-tweet-tweet-causes-online-flutter-109042900097_1.html. Diakses 15 Agustus 2020. Jam 19:44 WIB
- Deviyanto, A., & Wahyudi, M. D. R. (2018). *Penerapan Analisis Sentimen Pada Pengguna Twitter Menggunakan Metode K-Nearest Neighbor*. JISKA (Jurnal Informatika Sunan Kalijaga), 3(1), 1-13.
- Elcom., 2010, *Twitter Best Sosial Networking*. Andi Publisher : Yogyakarta.
- Faisal, R. M., & Nugraha, D. T. (2019). *Belajar Data Science dengan Pemrograman R*. Scripta Cendekia : Banjarbaru, Kalimantan Selatan.
- Hartono., Abdullah, D., Hartama, D., Furqan, M., Zarlis, M., & Situmorang, Z. (2016). *Sentiment Analysis Using Context Based Fuzzy Linguistic Hedges*. Proceeding International Conference of Computer, Environment, Social Science, Engineering and Technology (ICEST) 2016, 160–162
- Han, J., & Kamber, M. (2006). *Data Mining Concepts and Techniques*. 2nd Edition. In Elsevier : San Francisco.
- Hearst, M. (2003). *What Is Text Mining?* SIMS, UC Berkeley.<https://people.ischool.berkeley.edu/~hearst/text-mining.html>. Diakses 18 September 2020. Jam 15:40 WIB.
- Hutapea, T. & I. (2018). *Penerapan Algoritma Modified K-Nearest Neighbour Pada Pengklasifikasian Penyakit Kejiwaan Skizofrenia*. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 2(10), 3957–3961.
- Gugus Tugas Percepatan Penanganan Covid-19. <https://covid19.go.id/>. Diakses 18 Agustus 2020. Jam 11:44 WIB.
- Jabal Tursina, M. (2019). *Sentimen Analisis Sistem Zonasi Sekolah Pada Media Sosial Youtube Menggunakan Metode K-Nearest Neighbor Dengan Algoritma Levenshtein Distance*. Skripsi Fakultas Sains dan Teknologi Program Studi Teknik Informatika, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

- Kurnia, Y. (2012). *Implementasi Antar Muka Aplikasi Data Mining Algorithm Collection Dan Modul Preprocessing Data*. Skripsi Fakultas Ilmu Komputer, Program Studi Ilmu Komputer Universitas Indonesia.
- Kusuma, Y. (2009). *Pintar Twitter*. PT. Grasindo : Jakarta.
- Kamus Stopwords. Masdevi. <https://github.com/masdevi/ID-Stopwords>. Akses 11 Oktober 2020. Jam 16:29 WIB.
- Kamus OpinionWord. Masdevi. <https://github.com/masdevi/ID-OpinionWords>. Akses 11 Oktober 2020. Jam 16:32 WIB.
- Liu, B. (2015). *Sentiment Analysis, Mining Opinions, Sentiment and Emotions*. In International Journal of Advanced Computer Science and Applications. University of Cambrige : United State of America
- Liu, B., Hu, M., & Cheng, J. (2005). *Opinion observer: Analyzing and Comparing Opinions on the Web* Bing. *Proceedings of the 14th International World Wide Web Conference (WWW-2005)*, May 10-14, Chiba, Japan.
- Lukito, Y., & Chrismanto, A. R. (2015). *Perbandingan Metode-Metode Klasifikasi untuk Indoor Positioning System*. Jurnal Teknik Informatika Dan Sistem Informasi, 1(2), 123–131.
- Mahendrajaya, R., Buntoro, G. A., & Setyawan, M. B. (2019). *Analisis Sentimen Pengguna Gopay Menggunakan Metode Lexicon Based Dan Support Vector Machine*. Komputek : Jurnal Teknik Universitas Muhammadiyah Ponorogo, 3(2), 52–63.
- Manalu, B.U. (2014). *Analisis Sentimen Pada Twitter Menggunakan Text Mining*. Skripsi Teknologi Informasi Fak. Ilmu Komputer Dan Teknologi Informasi, Universitas Sumatera Utara Medan.
- Nasukawa, T., & Yi, J. (2003). *Sentiment analysis: Capturing favorability using natural language processing*. *Proceedings of the 2nd International Conference on Knowledge Capture, K-CAP 2003, January 2003*, 70–77.
- Nimas Safitri, R. (2020). *Analisis Sentimen Review Pelanggan Hotel Menggunakan Metode K-Nearest Neighbor (K-NN) (Studi Kasus : Hotels.com, Booking.com, Agoda.com)*. Skripsi Fakultas Teknologi Dan Informatika, Universitas Dinamika.
- Nugraha , F. A., Harani, N. H., & Habibi, R. (2020). *Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning*. Kreatif Industri Nusantara : Bandung.

- Nurjanah, W. E., Perdana, R. S., & Fauzi, M. A. (2017). *Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode K-Nearest Neighbor dan Pembobotan Jumlah Retweet*. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya, 1(12), 1750–1757.
- Prihantini, A. (2015). *Master Bahasa Indonesia*. PT. Bentang Pustaka : Yogyakarta.
- Purnamasari, D., Henharta, J., Sasmita, Y. P., Ihsani, F., & Wicaksana, I. W. S. (2013). *Machine Learning “Get Easy Using WEKA.”* Cet.I. Dapur Buku : Makasar.
- Permadi, A.V. (2020). *Analisis Sentimen Menggunakan Algoritma Naive Bayes Terhadap Review Restoran Di Singapura*. Jurnal Buana Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, 11(2), 141-151.
- Rizal, M. (2017). *Analisis Sentimen Pengguna Twitter Terhadap Objek Pariwisata di Indonesia Menggunakan Algoritma Pengolahan Deep Natural Language dari IBM Insights untuk Twitter*. Skripsi Fakultas Sains dan Teknologi Universitas Islam Negeri Alauddin Makasar.
- Rizki, M. M. (2019). *Analisis Sentimen Terhadap Produk Otomotif Dari Twitter Menggunakan Kombinasi Algoritma K-Nearest Neighbor Dan Pendekatan Lexicon (Studi Kasus: Mobil Toyota)*. Skripsi Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.
- Santoso, E. B., & Nugroho, A. (2019). *Analisis Sentimen Calon Presiden Indonesia 2019 Berdasarkan Komentar Publik Di Facebook*. Eksplora Informatika, 9(1), 60–69.
- Sari, R. (2020). *Analisis Sentimen Pada Review Objek Wisata Dunia Fantasi Menggunakan Algoritma K-Nearest Neighbor (K-NN)*. EVOLUSI: Jurnal Sains Dan Manajemen, 8(1), 10–17. <https://doi.org/10.31294/evolusi.v8i1.7371>
- Satrio, R. H., & Fauzi, M. A. (2019). *Klasifikasi Tweets Pada Twitter Menggunakan Metode K-Nearest Neighbour (K-NN) Dengan Pembobotan TF-IDF*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. 3(8), 8293–8300.
- Suntoro, J. (2019). *Data Mining : Algoritma dan Implementasi dengan Pemrograman PHP*. PT. Elex Media Komputindo : Jakarta.

- Tala, F. Z., 2003, *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. M.Sc. Thesis. Master of Logic Project.* Institute for Logic, Language and Computation. Universiteit van Amsterdam, The Netherlands.
- Wahid, D. H., & Azhari, S. N., 2016, *Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity. IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 10(2), 207-218.
- Wanto, A., Hasan, M.N., Windarto, A.P., Hartama,D., Ginantra, N.L.W.S.R., Napitupulu, D., Negara, E.S., Lubis, M.R., Dewi, S.V., & Cahyo Prianto., 2020, *Data Mining : Algoritma dan Implementasi.* Penerbit : Yayasan Kita Menulis.
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Practical, Data Mining : Machine Learning Tools and Techniques. 3rd Edition.* In Elsevier : USA.

LAMPIRAN 1
KAMUS STOPWORDS BERBAHASA INDONESIA

ada	bahwasanya	cuma	sehingga
adanya	sebaliknya	percuma	ia
adalah	banyak	dahulu	ialah
adapun	sebanyak	dalam	ibarat
agak	beberapa	dan	ingin
agaknya	seberapa	dapat	inginkah
agar	begini	dari	inginkan
akan	beginian	daripada	ini
akankah	beginikah	dekat	inikah
akhirnya	beginilah	demi	inilah
aku	sebegini	demikian	itu
akulah	begitu	demikianlah	itukah
amat	begitukah	sedemikian	itulah
amatlah	begitulah	dengan	jangan
anda	begitupun	depan	janganakan
andalah	sebegitu	di	janganlah
antar	belum	dia	jika
diantaranya	belumah	dialah	jikalau
antara	sebelum	dini	juga
antaranya	sebelumnya	diri	justru
diantara	sebenarnya	dirinya	kala
apa	berapa	terdiri	kalau
apaan	berapakah	dong	kaulah
mengapa	berapalah	dulu	kalaupun
apabila	berapapun	enggak	kalian
apakah	betulkah	enggaknya	kami
apalagi	sebetulnya	entah	kamilah
apatah	biasa	entahlah	kamu

atau	biasanya	terhadap	kamulah
ataukah	bila	terhadapnya	kan
ataupun	bilakah	hal	kapan
bagai	bisa	hampir	kapankah
bagaikan	bisakah	hanya	kapanpun
sebagai	sebisanya	hanyalah	dikarenakan
sebagainya	boleh	harus	karena
bagaimana	bolehkah	haruslah	karenanya
bagaimanapun	bolehlah	harusnya	ke
sebagaimana	buat	seharusnya	kecil
bagaimanakah	bukan	hendak	kemudian
bagi	bukankah	hendaklah	kenapa
bahkan	bukanlah	hendaknya	kepada
bahwa	bukannya	hingga	kepadanya
ketika	maupun	sambil	sepertinya
seketika	semaunya	sampai	sering
khususnya	memang	sana	seringnya
kini	mereka	sangat	serta
kinilah	merekalah	sangatlah	siapa
kiranya	meski	saya	siapakah
sekiranya	meskipun	sayalah	siapapun
kita	semula	se	disini
kitalah	mungkin	sebab	disinilah
kok	mungkinkah	sebabnya	sini
lagi	nah	sebuah	sinilah
lagian	namun	tersebut	sesuatu
selagi	nanti	tersebutlah	sesuatunya
lah	nantinya	sedang	suatu
lain	nyaris	sedangkan	sesudah
lainnya	oleh	sedikit	sesudahnya
melainkan	olehnya	segala	sudahlah

selaku	seorang	segalanya	supaya
lalu	seseorang	segera	tadi
melalui	pada	sesegea	tadinya
terlalu	padanya	sejak	tak
lama	padahal	sejenak	tanpa
lamanya	paling	sekali	setelah
selama	sepanjang	sekalian	telah
selama	pantas	sekalipun	tentang
selamanya	sepantasnya	sesekali	tentu
lebih	sepantasnyalah	sekaligus	tentulah
terlebih	para	sekarang	tentunya
bermacam	pasti	sekarang	tertentu
macam	pastilah	sekitar	seterusnya
semacam	per	sekitarnya	tapi
maka	pernah	sela	tetapi
makanya	pula	selain	setiap
makin	pun	selalu	tiap
malah	merupakan	seluruh	setidaknya
malahan	rupanya	seluruhnya	tidak
mampu	serupa	semakin	tidakkah
mampukah	saat	sementara	tidaklah
mana	saatnya	sempat	toh
manakala	sesaat	semua	waduh
manalagi	saja	semuanya	wah
masih	sajalah	sendiri	wahai
masihkah	saling	sendirinya	sewaktu
semasih	bersama	seolah	walau
masing	sama	seperti	wong
mau	sesama	sudah	yaitu
walaupun	sedikitnya	sudahkah	yakni dll..

LAMPIRAN 2

LISTING PROGRAM

Script program menggunakan bahasa Pemrograman R

Crawling Data

```
#import library
library(tm)
library(NLP)
library(twitterR)
library(plyr)
library(dplyr)
library(tidyr)

#Crawling Data Twitter
#Masukkan Kode Twitter API
consumer_key <- "8ZJ1SRv5NSGudNbW7Qpc6tiqg"
consumer_secret <- "frSRiYP3wBULd1fibRNGGIDiFaWDIvXHtA1SnjM6QH4
OPevXb0"
access_token <- "1297884879560962049-pRV6CDyKtVCBt07fqr5dXv7Hu7ku8v"
access_secret <- "sxPUqd0cXHzosEqgfQJm87rox7nLolj1a3C62v9f9IoUj"

#Mengambil Data Twitter
setup_twitter_oauth(consumer_key, consumer_secret, access_token,
access_secret)
crawling_tweet = searchTwitter('#NewNormal', n = 1000, lang = 'id')

#save datanya
saveRDS(crawling_tweet, file = 'datamentah.rds')

#Load dataset
crawling_tweet <- readRDS('datamentah.rds')
tweet.id = twListToDF(crawling_tweet)
View(tweet.id)
write.csv(tweet.id, file = 'datamentah#NewNormal.csv')

#Memanggil data
tweet<-read.csv("datamentah#NewNormal.csv", stringsAsFactors = TRUE)
View(tweet) #menampilkan data

#Mengambil tweet text saja
tweet_text <- tweet$text
View(tweet_text)

#save sebagai file csv
write.csv(tweet_text, file = 'crawling_datatweet.csv', row.names=
FALSE)
```

Preprocessing Data

```
library(tm)
library(NLP)
library(stringr)
library(caret)
library(dplyr)
library(devtools)
install_github("nurandi/katadasaR")
library(katadasaR)

#Memanggil data hasil crawling
tweet<-read.csv("crawling_datatweet.csv", stringsAsFactors = TRUE)
View(tweet) #menampilkan data

tw.id <- Corpus(VectorSource(tweet$x))

#Cleaning
remove.mention <- function(x) gsub("@\\S+", " ", x)
tw_mention <- tm_map(tw.id, remove.mention)

remove.hashtag <- function(x) gsub("#\\S+", " ", x)
tw_hashtag <- tm_map(tw_mention, remove.hashtag)

removeURL <- function(x) gsub("http[^[:space:]]*", " ", x)
tw_URL <- tm_map(tw_hashtag, content_transformer(removeURL))

removetitik2 <- function(x) gsub(":", " ", x)
tw_titik2 <- tm_map(tw_URL, removetitik2)

inspect(tw_titik2[1:50])

removetitik3 <- function(x) gsub("...", " ",x)
tw_titik3 <-tm_map(tw_titik2,removetitik3)

removetitikkoma <- function(x) gsub(";", " ", x)
tw_titikkoma <- tm_map(tw_titik3, removetitikkoma)

removeamp <- function(x) gsub("&", " ", x)
tw_amp <- tm_map(tw_titikkoma, removeamp)

removeslashn <- function(x) gsub("\n", " ",x)
tw_slashn <-tm_map(tw_amp,removeslashn)

removeslashn <- function(x) gsub("/", " ",x)
tw_slash <-tm_map(tw_slashn,removeslashn)

removeRT <- function(x) gsub("RT", " ", x)
tw_RT <- tm_map(tw_slash, removeRT)

inspect(tw_RT[1:50])

tw_punctuation<-tm_map(tw_RT,content_transformer(
removePunctuation))
```

```

tw_nonumber<-tm_map(tw_punctuation,
content_transformer(removeNumbers))

inspect(tw_nonumber[1:50])

#casefolding
tw_casefolding<- tm_map(tw_nonumber, content_transformer(
tolower))
inspect(tw_casefolding[1:50])

#Normalisasi
tw_norm1 <- tm_map(tw_casefolding, gsub,
pattern="dlm",replacement="dalam")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="utk",replacement="untuk")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="lha",replacement="lah")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="jd",replacement="jadi")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="gak",replacement="tidak")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="tdk",replacement="tidak")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="tak",replacement="tidak")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="yg",replacement="yang")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="lg",replacement="lagi")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="jgn",replacement="jangan")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="dr",replacement="dari")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="lp",replacement="lupa")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="lha",replacement="lah")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="krl",
replacement="kereta rel listrik")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="krn",
replacement="karena")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="dpt",
replacement="dapat")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="liat",
replacement="melihat")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="maen",
replacement="main")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="amp",
replacement="sampai")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="smp",
replacement="sampai")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="gimana",
replacement="bagaimana")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="adl",
replacement="adalah")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="tp",
replacement="tapi")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="Tp",
replacement="tapi")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="tsb",
replacement="tersebut")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="kl",
replacement="kalau")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="aja",
replacement="saja")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="udah",

```

```

replacement="sudah")
tw_norm1 <- tm_map(tw_norm1, gsub, pattern="lha",
replacement="lah")

inspect(tw_norm1[1:50])

#Stemming
library(caret)
library(tau)
library(tokenizers)

stem_tw<-function(text,mc.cores=1)
{
  stem_string<-function(str)
  {
    str<-tokenize(x=str)
    str<-sapply(str,katadasaR)
    str<-paste(str,collapse = "")
    return(str)
  }
  x<-mclapply(X=text,FUN=stem_string,mc.cores=mc.cores)
  return(unlist(x))
}
tw_stemming<-tm_map(tw_norm1,stem_tw)

inspect(tw_stemming[1:50])

##Filtering
#Stopwords
stopwords_id<-readLines("stopwords_id.txt")
tw_stopword <- tm_map(tw_stemming, removeWords, stopwords_id)

inspect(tw_stopword[1:50])

#Hapus data stopwords tambahan
datastopword <- tm_map(tw_stopword, removeWords,c("wa","br","s",
"jaw","mai","kit","jt","law","uf","i","sa","m","sa","k","url","uecudeu
ddudeudeuddcude","olv","pm","cr","ter","cv","sar","pt","cr","n","ubdub
dubdufufuf","h","rprb","widuriufduffbufdufffb","uddeudduec","mt","ken",
"untuk","jadi","ufuffb","kh","ssaja","ac","bag","gr","w","ce","u","b",
"hp","up","us","j","uududududud","udud","ududeududeudcududbudbudu","ue
cuecuc","lurd","g","shop®","ufuffbuufuffbuaufefufuffbuufefuufefuufefu
","jud","rr","fe","jam","sua","ase","dm","uu","p","lho","ufcuffb","gu
","te","wil","cgk","tni","rp","prio","tuju","ufuf","ocb","ga","ufe","uu
uuu","ufufa","ft","jtv","cred","okt","mil","ala","htt","fc","uffuffe",
"nak","shah","bas","ufufdfufef","uff","grm","akb","nas","ufeufufuff",
"ufuaaufef","isi","tengok"))

inspect(datastopword[1:50])

#Hapus spasi berlebih
tw_whitespace<-tm_map(datastopword,stripWhitespace)
inspect(tw_whitespace[1:50])

```

```
#menyimpan data
dataclean <- data.frame(text=unlist(sapply(tw_whitespace,`[`)),
stringsAsFactors=F)
write.csv(dataclean,file="datatweetbersih.csv")
```

Pelabelan & Menghitung Skor Sentimen

```
library(tm)
library(plyr)
library(stringr)

#Memanggil Data Tweet
datatweet<-read.csv("datatweetbersih.csv",header=TRUE)
View(datatweet)

#skoring label
positif <- scan("sentimen_positif.txt",what="character",
comment.char=";")
negatif <- scan("sentimen_negatif.txt",what="character",
comment.char=";")
kata.positif = c(positif)
kata.negatif = c(negatif)
score.sentiment = function(datatweet, kata.positif,kata.negatif,
.progress='none')

{
  require(plyr)
  require(stringr)
  scores = laply(datatweet, function(kalimat, kata.positif,
kata.negatif)
  {
    kalimat = gsub('[:punct:]', '', kalimat)
    kalimat = gsub('[:cntrl:]', '', kalimat)
    kalimat = gsub('\\d+', '', kalimat)
    kalimat = tolower(kalimat)
    list.kata = str_split(kalimat, '\\s+')
    kata2 = unlist(list.kata)
    positif.matches = match(kata2, kata.positif)
    negatif.matches = match(kata2, kata.negatif)
    positif.matches = !is.na(positif.matches)
    negatif.matches = !is.na(negatif.matches)
    score = sum(positif.matches) - (1*sum(negatif.matches))
    return(score)
  }, kata.positif, kata.negatif, .progress=.progress )
  scores.df = data.frame(score=scores, text=datatweet)
  return(scores.df)
}

hasil = score.sentiment(datatweet$text, kata.positif, kata.negatif)
View(hasil)

#Mengklasifikasikan Data Sentimen
hasil$klasifikasi<- ifelse(hasil$score<0, "Negatif","Positif")
hasil$klasifikasi
```



```

View(hasil)

#Ubah kolom
data <- hasil[c(3,2,1)]
View(data)

#Menyimpan Seluruh Data Sentimen
write.csv(data, file = "Datasantimen.csv")

#Menyimpan Data Positif dan Negatif
data.positif <- hasil[hasil$score>0,]
View(data.positif)
write.csv(data.positif, file = "data_positif.csv")

data.negatif <- hasil[hasil$score<0,]
View(data.negatif)
write.csv(data.negatif, file = "data_negatif.csv")

```

Visualisasi Data

```

library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(stringr)

data <- read.csv("datatweetbersih.csv",header=TRUE)
View(data)

tweet <- Corpus(VectorSource(data$text))

#Hapus data stopwords
data <- tm_map(tweet, removeWords,

c("wa", "br", "s", "jaw", "mai", "kit", "jt", "law", "uf", "i", "sa", "m",
  "sa", "k", "url", "uecudeuddudeudeuddcude", "olv",
  "pm", "cr", "ter", "cv", "sar", "pt", "cr", "n", "ubdubdubdufufuf", "h", "rprb",
  "widuriufduffbufdufffb", "uddeudduec", "mt", "ken", "untuk", "jadi", "ufuffb",
  "kh", "ssaja", "ac", "bag",
  "gr", "w", "ce", "u", "b", "hp", "up", "us", "j", "uududududud", "udud", "ududeud
  udeudcududbudbudu", "uecuec", "lurd", "g", "shop@", "ufuffbuufuffbuaufef
  ufuffbuufefuufefuufefu", "jud", "rr", "fe", "jam", "sua", "ase", "dm", "uu", "p",
  "lho", "ufcuffb", "gu", "te", "wil", "cgk", "tni", "rp", "prio", "tuju", "ufuf",
  "ocb", "ga", "ufe", "uuuuu", "ufufa", "ft", "jtv", "cred", "okt", "mil", "ala",
  "htt", "fc", "uffuffe", "nak", "shah", "bas", "ufufdfufef", "uff",
  "grm", "akb", "nas", "ufeufufuff", "ufuaaufef", "tengok"))

inspect(data[1:50])

#Hapus spasi berlebih
data <- tm_map(data, stripWhitespace)

```

```

inspect(data[1:150])

#Build a term-document matrix

dtm <- TermDocumentMatrix(data)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)

head(d,50)
View(d)

#wordcloud
library(wordcloud)
library(wordcloud2)
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
           max.words=100, random.order=FALSE, rot.per=0.35,
           colors=brewer.pal(8, "Dark2"))

#barplot
k<-barplot(d[1:15,]$freq, las = 2, names.arg =
d[1:15,]$word,cex.axis=1.2,cex.names=1.2,
          main = "Kata-Kata Yang Sering Muncul",
          ylab = "Jumlah Frekuensi",
          col = c("lightblue", "mistyrose", "lightcyan", "lavender",
"cornsilk"))

termFrequency <- rowSums(as.matrix(dtm))
termFrequency <- subset(termFrequency, termFrequency>=50)

text(k,sort(termFrequency, decreasing = T)-
2,labels=sort(termFrequency, decreasing = T),pch = 6, cex =1)

```

Visualisasi Data Positif

```

library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(stringr)

datapositif <-read.csv("data_positif.csv",header=TRUE)
View(datapositif)
datapositif <- Corpus(VectorSource(datapositif$text))

#Hapus data stopwords
datapositif <- tm_map(datapositif, removeWords,

c("ufcuffb","w","stud","uffuffe","y","ua","mai","meyaki","gr","ft","u"
,"tuju","konfe","akb","ufuffbuuffbuaufefufuffbuufefuufefuufefu","ase"
,"sua","j","g","law","prio","jaw","babakanmulya","si","hadeuuhh","la"
,"uf","nak","tengok","stadium","maca","wa","hj","shah","yab","uff","bi

```

```
as","re","uufef","msampaiir","ju","ufuaaufef","jersi","rm","tujua","un
t","h","k","sah","v","de","s","ma","mu","ni","sj","sy","rp","ko","ku",
"jud","uu","ufbufb"))
```

```
#Hapus spasi berlebihan
datapositif <- tm_map(datapositif, stripWhitespace)
```

```
inspect(datapositif[1:50])
```

```
#Build a term-document matrix
```

```
dtm <- TermDocumentMatrix(datapositif)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
```

```
head(d,50)
View(d)
```

```
#wordcloud
library(wordcloud)
library(wordcloud2)
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
           max.words=250, random.order=FALSE, rot.per=0.35,
           col = brewer.pal(8, "Dark2"))
```

```
#barplot
datapositif<-barplot(d[1:17,]$freq, las = 2, names.arg =
d[1:17,]$word,cex.axis=1.2,cex.names=1.2,
                 main ="Kata Positif yang Sering Muncul",
                 ylab = "Jumlah Frekuensi Sentimen Opini Positif",
                 col = c("lightblue", "mistyrose", "lightcyan",
                        "lavender", "cornsilk"))
```

```
termFrequency <- rowSums(as.matrix(dtm))
termFrequency <- subset(termFrequency, termFrequency>=36)
```

```
text(datapositif,sort(termFrequency, decreasing = T)-
1,labels=sort(termFrequency, decreasing = T),pch = 6, cex =1)
```

Visualisasi Data Negatif

```
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(stringr)

datanegatif <- read.csv("data_negatif.csv",header=TRUE)
View(datanegatif)
datanegatif <- Corpus(VectorSource(datanegatif$text))

inspect(datanegatif)
```

```

#Hapus data stopwords
datanegatif <- tm_map(datanegatif, removeWords,
c("uf","ufa","bas","ufuffb","ter","jut","p","mad","s","y","b","wa","ca",
",","uecudeuddudeudeuddcude","uddeudduec","m","jng","bo","rr","ufe","i",
"gu","jtv","uf","me","s","untuk","jadi","jalan","isi","jalanjalan","ay
o"))

inspect(datanegatif)

#Hapus spasi berlebih
datanegatif <- tm_map(datanegatif, stripWhitespace)

inspect(datanegatif[1:50])

#Build a term-document matrix

dtm <- TermDocumentMatrix(datanegatif)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)

head(d,50)
View(d)

#wordcloud
library(wordcloud)
library(wordcloud2)
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words=250, random.order=FALSE, rot.per=0.35,
          col = brewer.pal(8, "Dark2"))

#barplot
datanegatif<-barplot(d[1:17,]$freq, las = 2, names.arg =
d[1:17,]$word,cex.axis=1.2,cex.names=1.2,
          main ="Kata Negatif yang Sering Muncul",
          ylab = "Jumlah Frekuensi Sentimen Opini Negatif",
          col = c("lightblue", "mistyrose", "lightcyan", "lavender",
"cornsilk"))

termFrequency <- rowSums(as.matrix(dtm))
termFrequency <- subset(termFrequency, termFrequency>=15)

text(datanegatif,sort(termFrequency, decreasing = T)-
1,labels=sort(termFrequency, decreasing = T),pch = 6, cex =1)

```

LAMPIRAN 3
DAFTAR RIWAYAT HIDUP



I. Data Pribadi

Nama : Susan Mayang Sari
Tempat/Tanggal Lahir : Kutacane, 03 September 1996
Alamat : Jl. Bukit Barisan I, Glugur Darat I, Kec. Medan
Tim., Kota Medan, Sumatera Utara 20238
Agama : Islam
Jenis Kelamin : Perempuan
Status : Belum Menikah
No. Hp/Telp : 08566214809
Email : susanmys100@gmail.com

II. Pendidikan Formal

1. SDN 1 Lawe Bulan Kutacane, Aceh Tenggara TA. 2004 - 2009
2. MTsN Kutacane, Aceh Tenggara TA. 2009 - 2012
3. SMA Negeri 1 Kutacane, Aceh Tenggara TA. 2012 - 2015
4. S1 Prodi Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam
Negeri Sumatera Utara Medan TA. 2016 - 2020

III. Pengalaman Organisasi

Purna Paskibraka Aceh Tenggara Tahun 2014
HMJ Ilmu Komputer Fakultas Sains dan Teknologi UIN-SU Medan T. 2017











LAMPIRAN 4
KARTU BIMBINGAN SKRIPSI

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik 2016/2020

Nama : Susan Mayang Sari	Pembimbing I : Dr. Mhd. Furqan, S.Si, M. Comp. &
NIM : 0701162003	Pembimbing II : Sriani, S.Kom, M.Kom
Prog. Studi : Ilmu Komputer	SK Pembimbing :
Judul Skripsi : <div style="text-align: center;"> Analisis Sentimen Terhadap New Normal Di era Covid-19 Menggunakan Algoritma K-Nearest Neighbor (K-NN). </div>	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	10/07/20	Revisi Bab I		18/07/20	Revisi Bab I	
II	24/07/20	Acc. Bab I		29/07/20	Acc. Bab I	
III	12/08/20	Revisi Bab II		01/08/20	Revisi Bab II	
IV	28/08/20	Acc. Bab II		29/08/20	Acc. Bab II	
V	18/09/20	Revisi Bab III		14/08/20	Revisi Bab III	

VI	24/09/20	Acc. Bab III Lanjut Seminar Proposal		21/09/20	Acc. Bab III Lanjut Seminar Proposal	
VII	15/02/21	Revisi Bab IV * Bab V		14/02/21	Revisi Bab IV * Bab V	
VIII	14/03/21	Acc Bab IV * Bab V lanjut sedang Munagaseyah		05/03/21	Acc Bab IV * Bab V lanjut sedang Munagaseyah	
IX	15/03/21	Revisi Daftar Pustaka		16/03/21	Revisi bab IV*V (sistematika penulisan)	
X	20/03/21	Revisi abstrak		18/03/21	Acc Bab IV * V	

Medan, 20 Maret 2021

An. Dekan

Ketua Jurusan/Program Studi

Ilmu Komputer



Ilka Zufria, S.Kom., M.Kom
NIP. 198506042015031006