

**ANALISIS SENTIMEN MASYARAKAT TERHADAP  
PEMBELAJARAN DARING DI ERA PANDEMI COVID-19  
PADA MEDIA SOSIAL TWITTER MENGGUNAKAN  
EKSTRAKSI FITUR *COUNTVECTORIZER* DAN ALGORITME  
*K-NEAREST NEIGHBOR***

**TUGAS AKHIR**



**Oleh:**

**MUS PRIANDI  
NIM : 1711501559**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA  
2021**

**ANALISIS SENTIMEN MASYARAKAT TERHADAP  
PEMBELAJARAN DARING DI ERA PANDEMI COVID-19  
PADA MEDIA SOSIAL TWITTER MENGGUNAKAN  
EKSTRAKSI FITUR *COUNTVECTORIZER* DAN ALGORITME  
*K-NEAREST NEIGHBOR***

**Diajukan untuk memenuhi salah satu persyaratan  
memperoleh gelar Sarjana Komputer (S.Kom)**

**TUGAS AKHIR**



**Oleh:**

**MUS PRIANDI  
NIM : 1711501559**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA  
2021**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

---

**PERSETUJUAN TUGAS AKHIR**

Nama : MUS PRIANDI  
Nomor Induk Mahasiswa : 1711501559  
Program Studi : Teknik Informatika  
Bidang Peminatan : *Programming Expert*  
Jenjang Studi : Strata 1  
Judul : Analisis Sentimen Masyarakat Terhadap Pembelajaran Daring Di Era Pandemi Covid-19 Pada Media Sosial Twitter Menggunakan Ekstraksi Fitur *CountVectorizer* Dan Algoritme *K-Nearest Neighbor*

Disetujui untuk dipertahankan dalam sidang Tugas Akhir periode semester Gasal tahun ajaran 2020/2021

Jakarta, 08 Februari 2021

Dosen Pembimbing

( Painem, S.Kom., M.Kom. )

## ABSTRAK

### ANALISIS SENTIMEN MASYARAKAT TERHADAP PEMBELAJARAN DARING DI ERA PANDEMI COVID-19 PADA MEDIA SOSIAL TWITTER MENGGUNAKAN EKSTRAKSI FITUR *COUNTVECTORIZER* DAN ALGORITME *K-NEAREST NEIGHBOR*

Oleh : Mus Priandi (1711501559)

Pemerintah Indonesia telah mengeluarkan kebijakan Pembatasan Sosial Berskala Besar (PSBB) untuk mencegah penyebaran Covid-19. Kebijakan tersebut berdampak merubah sistem pembelajaran konvensional menjadi pembelajaran jarak jauh. Sistem pembelajaran jarak jauh dilakukan secara daring dengan memanfaatkan media komunikasi dan informasi, tanpa dibatasi oleh kendala waktu, ruang dan tempat serta keterbatasan sistem pembelajaran konvensional. Kurangnya kesiapan dalam menerapkan sistem pembelajaran baru tersebut memaksa banyak pihak untuk dapat beradaptasi dalam waktu yang cepat. Sistem pembelajaran yang semula dianggap sebagai solusi mulai menuai beragam pendapat dari masyarakat. Penelitian ini bertujuan untuk melakukan analisis pandangan masyarakat terhadap sistem pembelajaran daring pada media sosial Twitter. Metode yang digunakan adalah dengan melakukan analisis sentimen melalui pendekatan *machine learning* disertai fitur kamus sentimen, dengan ekstraksi fitur menggunakan *CountVectorizer* dan algoritme klasifikasi *K-Nearest Neighbor*. *Dataset* yang digunakan bersumber dari media sosial Twitter berupa kicauan (*tweet*) berbahasa Indonesia yang diperoleh melalui fitur pencarian dengan kata kunci ‘pembelajaran jarak jauh’, ‘belajar dari rumah’, ‘#belajaronline’, ‘#belajardarirumah’, ‘#belajardirumah’, dan ‘#kuliahonline’. Hasil analisis berdasarkan 3.954 *tweet* menunjukkan bahwa sentimen positif sebesar 76.56% dan sentimen negatif sebesar 23.44% pada periode Desember 2020, sementara hasil pengujian terbaik diperoleh menggunakan nilai  $K=7$ , dengan nilai akurasi sebesar 71%, presisi sebesar 74% dan *recall* sebesar 86%.

**Kata kunci :** analisis sentimen, twitter, pembelajaran daring, *countvectorizer*, *k-nearest neighbor*

xiv + 90 halaman; 56 gambar; 21 tabel



## SURAT PERNYATAAN TIDAK PLAGIAT DAN PERSETUJUAN PUBLIKASI

Saya yang bertanda tangan di bawah ini:

Nama : MUS PRIANDI  
NIM : 1711501559  
Program Studi : TEKNIK INFORMATIKA  
Bidang Peminatan : PROGRAMMING EXPERT  
Jenjang Studi : STRATA 1  
Fakultas : TEKNOLOGI INFORMASI

menyatakan bahwa TUGAS AKHIR yang berjudul:

ANALISIS SENTIMEN MASYARAKAT TERHADAP PEMBELAJARAN  
DARING DI ERA PANDEMI COVID-19 PADA MEDIA SOSIAL  
TWITTER MENGGUNAKAN EKSTRAKSI FITUR COUNTVECTORIZER  
PAN ALGORITME K-NEAREST NEIGHBOR

Merupakan:

1. Karya tulis saya sebagai laporan tugas akhir yang asli dan belum pernah diajukan untuk mendapatkan gelar akademik apapun, baik di Universitas Budi Luhur maupun di perguruan tinggi lainnya.
2. Karya tulis ini bukan saduran / terjemahan, dan murni gagasan, rumusan dan pelaksanaan penelitian / implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan pembimbing di organisasi tempat riset.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Saya menyerahkan hak milik atas karya tulis ini kepada Universitas Budi Luhur, dan oleh karenanya Universitas Budi Luhur berhak melakukan pengelolaan atas karya tulis ini sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh berdasarkan karya tulis ini, serta sanksi lainnya sesuai dengan norma di Universitas Budi Luhur dan Undang-Undang yang berlaku.

Jakarta, 08 Februari 2021



Mus Priandi

## KATA PENGANTAR

Puji serta syukur Alhamdulillah, penulis panjatkan kehadiran Allah Subhanahu Wa Ta'ala yang telah melimpahkan rahmat dan karunia-Nya, sehingga pada akhirnya penulis dapat menyelesaikan tugas akhir ini dengan baik. Adapun tugas akhir ini disusun untuk memenuhi persyaratan dalam menyelesaikan tingkat pendidikan Strata 1 (S1) pada program studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Budi Luhur dengan judul tugas akhir yang penulis angkat yaitu “Analisis Sentimen Masyarakat Terhadap Pembelajaran Daring Di Era Pandemi Covid-19 Pada Media Sosial Twitter Menggunakan Ekstraksi Fitur *CountVectorizer* Dan Algoritme *K-Nearest Neighbor*”.

Penulis berharap tugas akhir ini dapat memberikan manfaat kepada para pembaca. terselesaikannya penelitian ini tidak lepas dari bantuan berbagai pihak, rasa terima kasih yang mendalam juga penulis sampaikan kepada mereka yang telah berjasa dalam membantu penyusunan tugas akhir ini, terkhusus kepada:

1. Allah Subhanahu Wa Ta'ala, atas segala petunjuk, kemudahan, serta nikmat-Nya yang diberikan sehingga penulis dapat menyelesaikan penyusunan tugas akhir ini dengan baik.
2. Segenap keluarga penulis, khususnya orang tua tercinta, bapak dan ibu, serta adik, yang telah memberikan banyak dukungan baik berupa moral maupun material, juga do'a yang selalu dipanjatkan.
3. Bapak Dr. Ir. Wendi Usino, M.Sc. M.M., selaku Rektor Universitas Budi Luhur.
4. Bapak Dr. Deni Mahdiana, M.M. M.Kom, selaku Dekan Fakultas Teknologi Informasi Universitas Budi Luhur.
5. Bapak Subandi, Sp. Pd., M.M. selaku Dosen Penasehat Akademik.
6. Ibu Painem, S.Kom., M.Kom. selaku Dosen Pembimbing Tugas Akhir sekaligus Kepala Laboratorium ICT Universitas Budi Luhur, yang selalu memberikan arahan dan ilmu selama penulis mengabdikan di LAB ICT hingga menyelesaikan tugas akhir ini.
7. Bapak dan Ibu dosen-dosen Universitas Budi Luhur yang telah memberikan ilmu yang bermanfaat juga motivasi bagi penulis.
8. Rekan-rekan Asisten Laboratorium ICT Terpadu Universitas Budi Luhur khususnya angkatan 2017, sebagai rekan kerja selama 3 tahun mengabdikan di LAB ICT.
9. Semua pihak yang terkait, baik secara langsung maupun tidak langsung dalam penulisan tugas akhir ini dari tahap awal sampai tahap akhir penyusunan.

Jakarta, 08 Februari 2021

Penulis

## DAFTAR TABEL

Tabel 2.1 Confusion matrix .....	10
Tabel 2.2 Studi Literatur .....	10
Tabel 3.1 Sampel data tweet .....	18
Tabel 4.1 Proses perhitungan skor .....	34
Tabel 4.2 Sampel data latih .....	36
Tabel 4.3 List kata .....	37
Tabel 4.4 Fitur kata .....	37
Tabel 4.5 Vektor kosong latih .....	38
Tabel 4.6 Representasi vektor latih .....	39
Tabel 4.7 File JSON model latih .....	39
Tabel 4.8 Sampel data uji .....	41
Tabel 4.9 Vektor kosong uji .....	42
Tabel 4.10 Representasi vektor uji .....	42
Tabel 4.11 Hasil jarak euclidean distance .....	43
Tabel 4.12 Pengurutan jarak tetangga .....	44
Tabel 4.13 Data K tetangga terdekat .....	45
Tabel 4.14 Nilai probabilitas data uji .....	46
Tabel 4.15 Sampel data hasil prediksi .....	79
Tabel 4.16 Confusion matrix pengujian K=3 .....	80
Tabel 4.17 Nilai pengujian K=3 .....	80
Tabel 4.18 Hasil pengujian .....	81


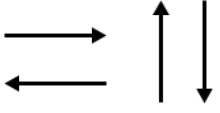

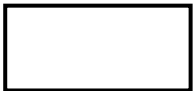

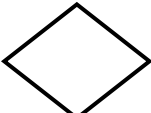
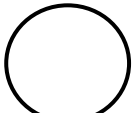
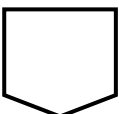

## DAFTAR GAMBAR

Gambar 2.1 Pelabelan kelas sentimen .....	8
Gambar 3.1 Tahapan metode .....	19
Gambar 3.2 Tahap pengumpulan data .....	20
Gambar 3.3 Tahap preprocessing .....	21
Gambar 3.4 Tahap labeling .....	24
Gambar 3.5 Tahap pembagian data .....	24
Gambar 3.6 Tahap modeling .....	25
Gambar 3.7 Tahap klasifikasi .....	26
Gambar 4.1 Proses case folding.....	30
Gambar 4.2 Proses menghapus URL.....	31
Gambar 4.3 Proses menghapus mention.....	31
Gambar 4.4 Proses menghapus hastag.....	32
Gambar 4.5 Proses menghapus selain huruf.....	32
Gambar 4.6 Proses menghapus spasi berlebih.....	32
Gambar 4.7 Proses merubah slang word .....	33
Gambar 4.8 Proses menghapus stop word .....	33
Gambar 4.9 Proses stemming .....	34
Gambar 4.10 Flowchart keseluruhan proses sistem.....	47
Gambar 4.11 Flowchart proses crawling .....	48
Gambar 4.12 Flowchart proses preprocessing.....	48
Gambar 4.13 Flowchart proses case folding.....	49
Gambar 4.14 Flowchart proses cleansing .....	49
Gambar 4.15 Flowchart proses slang word .....	50
Gambar 4.16 Flowchart proses stop word .....	50
Gambar 4.17 Flowchart proses stemming .....	51
Gambar 4.18 Flowchart proses labeling .....	51
Gambar 4.19 Flowchart proses pembagian data .....	52
Gambar 4.20 Flowchart proses modeling .....	53
Gambar 4.21 Flowchart proses klasifikasi.....	55
Gambar 4.22 Flowchart menu masuk aplikasi.....	55
Gambar 4.23 Flowchart menu aplikasi .....	56
Gambar 4.24 Flowchart menu beranda.....	57
Gambar 4.25 Flowchart menu slang word.....	57
Gambar 4.26 Flowchart menu stop word.....	58
Gambar 4.27 Flowchart menu kata positif.....	59
Gambar 4.28 Flowchart menu kata negatif.....	59
Gambar 4.29 Flowchart menu crawling .....	60
Gambar 4.30 Flowchart menu preprocessing .....	60
Gambar 4.31 Flowchart menu labeling.....	61
Gambar 4.32 Flowchart menu pembagian data .....	61
Gambar 4.33 Flowchart menu modeling .....	62



Gambar 4.34 Flowchart menu pengujian.....	62
Gambar 4.35 Flowchart menu visualisasi hasil .....	63
Gambar 4.36 Tampilan layar masuk aplikasi .....	81
Gambar 4.37 Tampilan layar beranda.....	82
Gambar 4.38 Tampilan layar kamus slang word .....	82
Gambar 4.39 Tampilan layar kamus stop word .....	83
Gambar 4.40 Tampilan layar kamus kata positif .....	83
Gambar 4.41 Tampilan layar kamus kata negatif .....	84
Gambar 4.42 Tampilan layar crawling .....	84
Gambar 4.43 Tampilan layar preprocessing .....	85
Gambar 4.44 Tampilan layar labeling .....	85
Gambar 4.45 Tampilan layar pembagian data .....	86
Gambar 4.46 Tampilan layar modeling .....	86
Gambar 4.47 Tampilan layar pengujian .....	87
Gambar 4.48 Tampilan layar visualisasi hasil .....	87

### DAFTAR SIMBOL *FLOWCHART*

SIMBOL	NAMA	KETERANGAN
	<i>Terminal Point Symbol</i> / Simbol Titik Terminal	Simbol untuk permulaan atau akhir dari suatu program
	<i>Flow Direction Symbol</i> / Simbol Arus	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain ( <i>connecting line</i> )
	<i>Input-Output</i> / Simbol Keluar-Masuk	Simbol yang menyatakan proses input atau <i>output</i>
	<i>Processing Symbol</i> / Simbol Proses	Simbol proses yang menunjukkan pengolahan yang dilakukan sistem
	<i>Predefined Process</i> / Simbol Proses Terdefinisi	Simbol yang Menggambarkan proses – proses yang masih dapat dijabarkan
	<i>Decision Symbol</i> / Simbol Keputusan	Simbol kondisi yang akan menghasilkan <i>output true</i> atau <i>false</i>
	<i>Connector (On-page)</i>	Simbol keluar atau masuk prosedur atau proses dalam lembar atau halaman yang sama
	<i>Connector (Off-page)</i>	Simbol yang digunakan untuk menghubungkan simbol dalam halaman berbeda
	<i>Document Symbol</i> / Simbol Dokumen	Simbol yang menyatakan proses input atau <i>output</i> yang melibatkan dokumen atau file

## DAFTAR ISI

ABSTRAK.....	iv
KATA PENGANTAR .....	vi
DAFTAR TABEL.....	vii
DAFTAR GAMBAR .....	viii
DAFTAR SIMBOL <i>FLOWCHART</i> .....	x
DAFTAR ISI.....	xi
BAB I    PENDAHULUAN.....	1
1. 1.    Latar Belakang.....	1
1. 2.    Perumusan Masalah.....	2
1. 3.    Batasan Masalah .....	2
1. 4.    Tujuan.....	3
1. 5.    Manfaat.....	3
1. 6.    Sistematika Penulisan .....	3
BAB II    LANDASAN TEORI ...	5
2. 1. <i>Text Mining</i> .....	5
2. 2.    Analisis Sentimen .....	5
2. 3.    Media Sosial .....	5
2. 4. <i>Crawling</i> .....	6
2. 5. <i>Preprocessing</i> .....	6
2. 5. 1. <i>Case folding</i> .....	6
2. 5. 2. <i>Cleansing</i> .....	6
2. 5. 3. <i>Mengubah slang word</i> .....	6
2. 5. 4. <i>Menghapus stop word</i> .....	7
2. 5. 5. <i>Stemming</i> .....	7
2. 6. <i>Labeling</i> .....	7
2. 6. 1.    Perhitungan skor sentimen .....	7
2. 6. 2.    Pemberian kelas sentimen.....	8
2. 7. <i>CountVectorizer</i> .....	8
2. 8. <i>Modeling</i> .....	8
2. 9. <i>K-Nearest Neighbor</i> .....	9
2. 10.    Pengujian dan Evaluasi.....	9

2. 11. Studi Literatur .....	10
BAB III METODOLOGI PENELITIAN .....	18
3. 1. Data Penelitian.....	18
3. 2. Penerapan Metode .....	19
3. 2. 1. Pengumpulan data.....	20
3. 2. 2. <i>Preprocessing</i> .....	21
3. 2. 3. <i>Labeling</i> .....	23
3. 2. 4. Pembagian data .....	24
3. 2. 5. <i>Modeling</i> .....	25
3. 2. 6. Klasifikasi <i>K-Nearest Neighbor</i> .....	26
3. 3. Rancangan Pengujian .....	27
3. 3. 1. Akurasi .....	27
3. 3. 2. Presisi .....	27
3. 3. 3. <i>Recall</i> .....	27
BAB IV HASIL DAN PEMBAHASAN .....	29
4. 1. Lingkungan Percobaan .....	29
4. 1. 1. Spesifikasi perangkat keras.....	29
4. 1. 2. Spesifikasi perangkat lunak .....	29
4. 2. Implementasi Metode .....	29
4. 2. 1. Tahap pengumpulan data .....	29
4. 2. 2. Tahap <i>preprocessing</i> .....	30
4. 2. 3. Tahap <i>labeling</i> .....	34
4. 2. 4. Tahap pembagian data .....	35
4. 2. 5. Tahap ekstraksi fitur <i>CountVectorizer</i> .....	35
4. 2. 6. Tahap klasifikasi <i>K-Nearest Neighbor</i> .....	41
4. 3. <i>Flowchart</i> Tahapan Metode .....	47
4. 3. 1. <i>Flowchart</i> keseluruhan proses sistem .....	47
4. 3. 2. <i>Flowchart</i> proses <i>crawling</i> .....	48
4. 3. 3. <i>Flowchart</i> proses <i>preprocessing</i> .....	48
4. 3. 4. <i>Flowchart</i> proses <i>labeling</i> .....	51
4. 3. 5. <i>Flowchart</i> proses pembagian data .....	52
4. 3. 6. <i>Flowchart</i> proses <i>modeling</i> .....	52



4. 3. 7.	<i>Flowchart</i> proses klasifikasi .....	54
4. 3. 8.	<i>Flowchart</i> menu masuk aplikasi .....	55
4. 3. 9.	<i>Flowchart</i> menu aplikasi.....	56
4. 3. 10.	<i>Flowchart</i> menu beranda .....	56
4. 3. 11.	<i>Flowchart</i> menu <i>slang word</i> .....	57
4. 3. 12.	<i>Flowchart</i> menu <i>stop word</i> .....	58
4. 3. 13.	<i>Flowchart</i> menu kata positif .....	58
4. 3. 14.	<i>Flowchart</i> menu kata negatif .....	59
4. 3. 15.	<i>Flowchart</i> menu <i>crawling</i> .....	60
4. 3. 16.	<i>Flowchart</i> menu <i>preprocessing</i> .....	60
4. 3. 17.	<i>Flowchart</i> menu <i>labeling</i> .....	61
4. 3. 18.	<i>Flowchart</i> menu pembagian data.....	61
4. 3. 19.	<i>Flowchart</i> menu <i>modeling</i> .....	62
4. 3. 20.	<i>Flowchart</i> menu pengujian .....	62
4. 3. 21.	<i>Flowchart</i> menu visualisasi hasil.....	63
4. 4.	Algoritme Tahapan Metode.....	63
4. 4. 1.	Algoritme keseluruhan proses sistem .....	63
4. 4. 2.	Algoritme proses <i>crawling</i> .....	64
4. 4. 3.	Algoritme proses <i>preprocessing</i> .....	64
4. 4. 4.	Algoritme proses <i>labeling</i> .....	66
4. 4. 5.	Algoritme proses pembagian data.....	67
4. 4. 6.	Algoritme proses <i>modeling</i> .....	68
4. 4. 7.	Algoritme proses klasifikasi .....	69
4. 4. 8.	Algoritme menu masuk aplikasi .....	71
4. 4. 9.	Algoritme menu aplikasi.....	71
4. 4. 10.	Algoritme menu beranda.....	72
4. 4. 11.	Algoritme menu <i>slang word</i> .....	72
4. 4. 12.	Algoritme menu <i>stop word</i> .....	73
4. 4. 13.	Algoritme menu kata positif .....	74
4. 4. 14.	Algoritme menu kata negatif .....	75
4. 4. 15.	Algoritme menu <i>crawling</i> .....	76
4. 4. 16.	Algoritme menu <i>preprocessing</i> .....	76

4. 4. 17.	Algoritme menu <i>labeling</i> .....	77
4. 4. 18.	Algoritme menu pembagian data .....	77
4. 4. 19.	Algoritme menu <i>modeling</i> .....	78
4. 4. 20.	Algoritme menu pengujian .....	78
4. 4. 21.	Algoritme menu visualisasi hasil .....	79
4. 5.	Pengujian .....	79
4. 6.	Tampilan Layar Aplikasi .....	81
4. 6. 1.	Tampilan layar masuk aplikasi .....	81
4. 6. 2.	Tampilan layar beranda.....	82
4. 6. 3.	Tampilan layar kamus kata .....	82
4. 6. 4.	Tampilan layar <i>crawling</i> .....	84
4. 6. 5.	Tampilan layar <i>preprocessing</i> .....	84
4. 6. 6.	Tampilan layar <i>labeling</i> .....	85
4. 6. 7.	Tampilan layar pembagian data .....	85
4. 6. 8.	Tampilan layar <i>modeling</i> .....	86
4. 6. 9.	Tampilan layar pengujian .....	86
4. 6. 10.	Tampilan layar visualisasi hasil .....	87
BAB V	PENUTUP .....	88
5. 1.	Kesimpulan.....	88
5. 2.	Saran .....	88
DAFTAR PUSTAKA	.....	89

## **BAB I**

### **PENDAHULUAN**

#### **1. 1. Latar Belakang**

Organisasi Kesehatan Dunia (WHO) telah menyatakan bahwa *Coronavirus disease* 2019 atau Covid-19 dikategorikan sebagai pandemi global pada 11 Maret 2020 (Watrianthos, 2020). Pandemi tersebut menyebar dengan sangat cepat dan telah melanda 215 negara di dunia (Sadikin and Hamidah, 2020). Penyebaran virus melalui kontak fisik memaksa semua negara untuk menerapkan *social distancing* dan *physical distancing* guna mengurangi interaksi antara orang-orang. Pemerintah Indonesia melalui Presiden Jokowi telah mengeluarkan pernyataan terkait *social distancing* dan *physical distancing* ini dengan dikeluarkannya kebijakan Pembatasan Sosial Berskala Besar (PSBB) untuk mencegah penyebaran virus (Ristyawati, 2020).

Pendidikan merupakan salah satu bidang yang terkena dampak pandemi Covid-19. Kementerian Pendidikan dan Kebudayaan Republik Indonesia melalui menterinya Nadiem Makarim telah mengeluarkan kebijakan tentang pelaksanaan pendidikan dalam masa darurat Covid-19. Kebijakan tersebut menjelaskan tentang pelaksanaan proses Belajar Dari Rumah (BRD) secara daring atau *online*. Karenanya, seluruh institusi pendidikan diminta untuk menghentikan proses belajar mengajar baik di sekolah maupun di kampus dan menggantinya dengan sistem belajar jarak jauh. Hal ini mengakibatkan semua model pembelajaran saat ini harus berlangsung secara daring atau *online* dengan bantuan alat perantara seperti *handphone*, komputer, atau laptop (Watrianthos, 2020). Pengajar, murid, dan orang tua harus menyesuaikan diri dengan model pembelajaran tersebut. Bagi sebagian pengajar yang tidak mahir dalam penggunaan teknologi akan merasa terkejut dan harus segera beradaptasi, demikian juga murid juga orang tua. Menurut Hadion Wijoyo (Wijoyo, 2020), diketahui bahwa pengajar menyenangi kelas daring sebesar 67% sedangkan yang lainnya lebih menyenangi kelas luring dikarenakan membutuhkan waktu lebih dalam mempersiapkan bahan ajar di kelas daring termasuk pemahaman perangkat IT yang digunakan. Sistem pembelajaran yang semula dianggap sebagai solusi mulai menuai beragam pendapat dari masyarakat.

Menurut Ronal Watrianthos (Watrianthos, 2020) melalui penelitian yang berjudul Analisis Pembelajaran Daring di Era Pandemic Covid-19, hasil penelitian menunjukkan pendapat (sentimen) masyarakat terhadap pembelajaran daring cenderung mengarah pada hasil sentimen yang negatif sebesar 83% pada bulan Juli 2020. Dalam penelitian tersebut juga dilakukan analisis emosi, menunjukkan bahwa '*trust*' atau kepercayaan sangat mendominasi yang menandakan kepercayaan terhadap pembelajaran daring telah jauh menurun. Penelitian lain yang pernah dilakukan terkait analisis sentimen diantaranya adalah Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan

K-Nearest Neighbor (Septian, Fahrudin and Nugroho, 2019), menyatakan bahwa algoritme *K-Nearest Neighbor* (KNN) mampu memperoleh nilai akurasi 79.99% dengan nilai  $K=23$ . Penelitian lain yang berjudul Perbandingan Metode Naive Bayes, KNN Dan Decision Tree Terhadap Analisis Sentimen Transportasi KRL Commuter Line (Romadloni, Santoso and Budilaksono, 2019), menyatakan bahwa KNN dapat digunakan untuk analisis sentimen dengan nilai akurasi sebesar 80% terhadap 127 data dan mampu mengimbangi algoritme *Naive Bayes Classifier*. KNN juga digunakan oleh Novelty dan Adiwijaya (Daeli and Adiwijaya, 2020) dalam penelitian yang berjudul *Sentiment Analysis on Movie Reviews Using Information Gain and K-Nearest Neighbor*, untuk melakukan analisis sentimen terhadap dataset *review* film dengan total 2000 data, memperoleh hasil yang baik pada  $K=3$  dengan nilai akurasi sebesar 96.8%.

Penelitian ini bertujuan untuk melakukan analisis sentimen masyarakat terhadap pembelajaran daring pada periode Desember 2020. Metode yang digunakan adalah dengan melakukan analisis sentimen melalui pendekatan *machine learning* disertai fitur kamus sentimen, dengan ekstraksi fitur menggunakan *CountVectorizer* dan algoritme klasifikasi *K-Nearest Neighbor*. *Dataset* yang digunakan berupa teks kicauan (*tweet*) yang bersumber pada media sosial Twitter dengan kata kunci ‘pembelajaran jarak jauh’, ‘belajar dari rumah’, ‘#belajaronline’, ‘#belajardarirumah’, ‘#belajardirumah’, dan ‘#kuliahonline’. Pengumpulan *dataset* dilakukan pada tanggal 1 Desember 2020 hingga 31 Desember 2020. Tercatat sebanyak 4.314 *tweet* yang diperoleh menggunakan kata kunci dan rentang tanggal yang diusulkan.

### 1. 2. Perumusan Masalah

Berdasarkan uraian latar belakang yang telah disampaikan, maka dapat disimpulkan rumusan masalah sebagai berikut:

- a. Bagaimana persentase pandangan (sentimen) masyarakat Indonesia terhadap pembelajaran daring pada periode waktu 1 Desember 2020 hingga 31 Desember 2020?
- b. Bagaimana cara menganalisis sentimen berdasarkan pendapat masyarakat Indonesia melalui media sosial Twitter?
- c. Berapa nilai akurasi yang diperoleh algoritme *K-Nearest Neighbor* dalam melakukan analisis sentimen?

### 1. 3. Batasan Masalah

Adapun batasan atau ruang lingkup masalah dalam penelitian ini adalah sebagai berikut:

- a. *Platform* yang digunakan hanya berbasis *web*.
- b. *Dataset* bersumber pada Twitter, terbatas pada *tweet* berbahasa Indonesia kata kunci ‘pembelajaran jarak jauh’, ‘belajar dari rumah’, ‘#belajaronline’, ‘#belajardarirumah’, ‘#belajardirumah’, dan



‘#kuliahonline’ pada rentang tanggal 1 Desember 2020 sampai dengan 31 Desember 2020.

- c. Fitur *import* hanya dapat mengenali file masukan berupa *excel* dengan ekstensi *.xls* atau *.xlsx*.
- d. Aplikasi hanya mengklasifikasikan tweet menjadi dua buah kategori sentimen, yaitu: “positif” dan “negatif”.

#### **1. 4. Tujuan**

Adapun tujuan dari dilakukan penelitian ini adalah sebagai berikut:

- a. Melakukan analisis sentimen masyarakat terhadap pembelajaran daring melalui media sosial Twitter.
- b. Merancang sebuah *model* penelitian untuk menganalisis sentimen dengan topik terkait pembelajaran daring.
- c. Menguji keakuratan algoritme *K-Nearest Neighbor* dalam melakukan analisis sentimen.

#### **1. 5. Manfaat**

Adapun manfaat dari penelitian adalah untuk menganalisis pandangan (sentimen) masyarakat Indonesia berdasarkan *tweet* yang dipublikasikan melalui media sosial Twitter. Sehingga dapat diperoleh gambaran sentimen masyarakat terkait topik pembelajaran daring di tengah pandemi Covid-19. Hasil penelitian ini juga diharapkan menjadi bahan evaluasi untuk sistem pembelajaran daring yang akan berlangsung. Penelitian ini juga dilakukan untuk menguji kinerja dan nilai akurasi algoritme *K-Nearest Neighbor* untuk analisis sentimen.

#### **1. 6. Sistematika Penulisan**

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut:

### **BAB I: PENDAHULUAN**

Bagian ini berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan juga membahas mengenai sistematika penulisan.

### **BAB II: LANDASAN TEORI**

Bagian ini berisi tentang algoritme dan metode yang akan dibahas, serta teori-teori yang berkaitan dengan penelitian ini, antara lain pengertian dan pemahaman materi terkait *text mining*, analisis sentimen, Twitter, *crawling*, *preprocessing*, *labeling*, *CountVectorizer*, *K-Nearest Neighbor*, *modeling*, dan pengujian serta studi literatur.

### **BAB III: METODOLOGI PENELITIAN**

Bagian ini berisi tentang sumber data penelitian, penerapan dan tahapan metode yang digunakan, serta rancangan pengujian yang akan dilakukan.

### **BAB IV: HASIL DAN PEMBAHASAN**

Bagian ini berisi mengenai lingkungan percobaan dari sistem yang dibuat, implementasi metode, flowchart tahapan metode, dan uraian algoritme pada proses, serta analisis pengujian sistem yang telah dibangun pada sisi akurasi, presisi, dan *recall*.

### **BAB V: PENUTUP**

Bagian ini berisi tentang kesimpulan yang dapat diambil dari penelitian dan saran untuk pengembangan lebih lanjut mengenai topik terkait untuk penelitian berikutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2. 1. Text Mining**

*Text mining* merupakan proses *mining* atau menambang suatu informasi dari data yang tersaji dalam jumlah besar, dalam hal ini adalah teks. Proses ini dilakukan dalam rangka penggalian, pengolahan, serta pengaturan pada informasi dengan menganalisis keterkaitan antara informasi satu dengan yang lainnya (Sudiantoro and Zuliarso, 2018). Dalam definisi lain, *text mining* adalah proses penambangan yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, dan tidak diketahui sebelumnya, atau menemukan kembali informasi yang tersirat secara implisit (Sari and Wibowo, 2019).

#### **2. 2. Analisis Sentimen**

Analisis sentimen merupakan bidang penelitian yang sedang berlangsung di bidang *text mining*. Tujuan dari analisis sentimen adalah untuk menganalisis opini, sentimen, dan subjektivitas teks. Analisis sentimen juga dapat disamakan dengan *opinion mining* karena berfokus kepada pendapat, sikap, emosi yang mewakili pandangan individu terkait peristiwa atau topik tertentu (Afrizal *et al.*, 2019) (Medhat, Hassan and Korashy, 2014). Saat ini, analisis sentimen banyak digunakan oleh peneliti sebagai salah satu cabang riset dalam ilmu komputer seiring dengan ledakan informasi di internet. Twitter merupakan salah satu media sosial yang paling populer untuk digunakan sebagai sumber data pada analisis teks (Watrianthos, 2020) (Ferdiana *et al.*, 2019).

#### **2. 3. Media Sosial**

Media sosial merupakan media penyampaian informasi yang banyak menjadi pilihan masyarakat, dengan adanya media sosial pengguna dapat memanfaatkan akun yang dimiliki untuk mengungkapkan perasaan baik atau buruk terhadap suatu peristiwa atau objek tertentu (Oktasari, Chrisnanto and Yuniarti, 2016).

##### **2. 3. 1. Twitter**

Twitter merupakan jejaring sosial daring dan layanan *microblogging* yang memungkinkan pengguna terdaftar untuk membaca dan memposting pesan singkat yang disebut dengan kicauan (*tweet*) (Aribowo, 2018) (Septian, Fahrudin and Nugroho, 2019). Twitter juga merupakan media sosial yang populer kalangan masyarakat Indonesia, menurut penelitian dan analisis oleh statista.com tercatat negara Indonesia menempati peringkat ke-7 dengan 13.2 miliar pengguna pada Oktober 2020 (Statista Research Department, 2020). Pada umumnya *tweet* diunggah untuk menyampaikan sebuah berita atau informasi terkait peristiwa tertentu, isi *tweet* juga dapat mengekspresikan sebuah pendapat dari penggunanya. Karena hal tersebut, Twitter banyak digunakan sebagai objek penelitian. Hal ini karena tulisan-tulisan pada media

sosial Twitter (*tweet*), memiliki struktur yang sangat cocok untuk digunakan pada analisis (Ferdiana *et al.*, 2019).

#### 2. 4. *Crawling*

*Crawling* merupakan proses mengumpulkan data dari sebuah laman dan menyimpannya untuk diatur dan dianalisis lebih lanjut (Nurulbaiti and Retno Subekti, 2020). Dalam penelitian ini proses *crawling* dilakukan menggunakan *standard search* API Twitter dengan pustaka Tweepy. Penggunaan pustaka Tweepy bertujuan untuk memperoleh data *tweet* pada Twitter dengan akses menggunakan API Key yang didapatkan melalui akun *developer* Twitter. Dalam penelitian ini, 4.314 *dataset* berhasil dikumpulkan berdasarkan enam (6) parameter kata kunci antara lain: ‘pembelajaran jarak jauh’, ‘belajar dari rumah’, ‘#belajaronline’, ‘#belajardarirumah’, ‘#belajardirumah’, dan ‘#kuliahonline’ dalam periode waktu 1 Desember 2020 hingga 31 Desember 2020.

#### 2. 5. *Preprocessing*

*Preprocessing* merupakan bagian dari *text mining* yang dilakukan untuk menghapus *noise* pada dokumen atau kalimat. Selain itu, proses ini bertujuan untuk menghindari data yang kurang sempurna; gangguan pada data; dan data yang tidak konsisten (Sari and Wibowo, 2019). Proses pengubahan data teks yang tidak terstruktur menjadi data teks yang terstruktur sangat diperlukan sehingga perlu adanya proses pra-pemrosesan data (Sudiantoro and Zuliarso, 2018). Merujuk pada penelitian yang telah dilakukan (Watrianthos, 2020) (Santoso and Nugroho, 2019) (Fitriyyah, Safriadi and Pratama, 2019) (Antinasari, Perdana and Fauzi, 2017) maka dalam penelitian ini akan dilakukan beberapa tahapan *preprocessing* teks antara lain: *case folding*, *cleansing*, mengubah *slang word*, menghapus *stop word*, dan *stemming*.

##### 2. 5. 1. *Case folding*

*Case folding* merupakan proses yang bertujuan untuk mengubah seluruh huruf menjadi huruf kecil (*lowercase*) (Santoso and Nugroho, 2019) (Fitriyyah, Safriadi and Pratama, 2019).

##### 2. 5. 2. *Cleansing*

*Cleansing* merupakan proses yang bertujuan untuk menghapus atribut yang tidak diperlukan untuk proses analisis (Watrianthos, 2020) (Santoso and Nugroho, 2019) (Fitriyyah, Safriadi and Pratama, 2019). *Cleansing* yang dilakukan dalam penelitian terdiri atas beberapa tahapan antara lain: menghapus URL, *mention* (@*mention*), *hashtag* (#*hashtag*), selain huruf (a-z) dan spasi berlebih.

##### 2. 5. 3. *Mengubah slang word*

*Slang word* merupakan kata yang tidak sesuai dengan ejaan bahasa Indonesia yang baku (EYD) baik berupa kata singkatan, kata gaul atau modern, ataupun kesalahan salah eja (Antinasari,



Perdana and Fauzi, 2017). *Slang word* tersebut sebanyak mungkin akan ditampung ke dalam kamus *slang word*. Kamus tersebut kemudian digunakan sebagai pengetahuan untuk melakukan *replace* atau mengubah kata *slang* menjadi kata dengan bahasa Indonesia yang baku sesuai EYD.

#### 2. 5. 4. Menghapus *stop word*

*Stop word* merupakan kata yang tidak berpengaruh atau kurang bermakna namun sering ditemui dalam dokumen atau kalimat, seperti kata ‘saya’, ‘dan’, ‘atau’ (Watrianthos, 2020) (Santoso and Nugroho, 2019). Dalam proses ini, kata yang tergolong ke dalam *stop word* akan ditampung ke dalam kamus *stop word*. Kamus tersebut kemudian digunakan sebagai pembanding untuk menghapus sebuah kata dalam dokumen atau kalimat yang tergolong ke dalam *stop word*.

#### 2. 5. 5. *Stemming*

*Stemming* merupakan proses mengubah kata berimbuhan menjadi kata dasar (Watrianthos, 2020) (Fitriyyah, Safriadi and Pratama, 2019). Dalam penelitian ini proses *stemming* dilakukan dengan menggunakan pustaka Sastrawi melalui dengan paket StemmerFactory.

### 2. 6. *Labeling*

*Labeling* atau pelabelan merupakan proses pemberian kelas berdasarkan ciri atau karakteristik yang terkandung dalam sebuah dokumen atau kalimat. Performa pembagian kelas lebih baik terbagi menjadi dua (2) kelas kelas sentimen, yakni sentimen positif dan sentimen negatif dibandingkan pembagian ke tiga buah kelas (Fitriyyah, Safriadi and Pratama, 2019). Dalam penelitian ini proses pelabelan akan memberikan kelas pada tiap *tweet* dengan positif atau negatif (2 kelas) yang dapat dilakukan dengan dua (2) cara antara lain: pelabelan manual dengan melabeli kalimat berdasarkan subjektivitas dan pelabelan dengan pendekatan kamus sentimen. Tahapan melakukan pelabelan dengan pendekatan kamus sentimen antara lain perhitungan skor sentimen dan pemberian kelas sentimen.

#### 2. 6. 1. Perhitungan skor sentimen

Perhitungan skor sentimen merupakan proses pelabelan dengan cara pendekatan kamus sentimen. Kamus tersebut berisikan kata opini positif dan kata opini negatif. Skor suatu kata akan bernilai +1 jika kata tersebut adalah kata opini positif, dan bernilai -1 jika kata tersebut adalah kata opini negatif (Buntoro, 2017) (Liu, Hu and Cheng, 2005). Perhitungan skor ini didasarkan pada frekuensi kemunculan kata positif dan negatif pada sebuah dokumen atau kalimat. Maka dapat diketahui bahwa nilai skor sentimen dapat diperoleh menggunakan rumus:

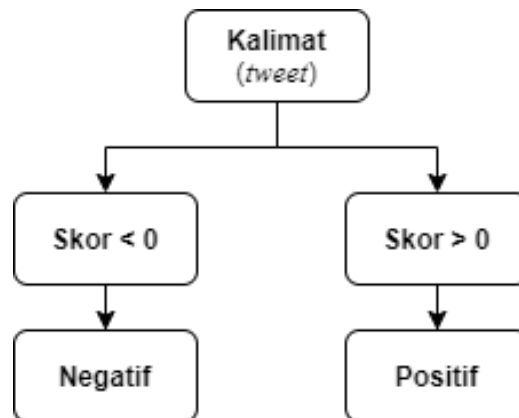
$$Skor_x = \left( \sum kata\ positif_x \right) - \left( \sum kata\ negatif_x \right) \quad (2.1)$$

Keterangan:

$x$  = sebuah dokumen atau kalimat  
*kata positif* = bilangan bulat positif atau nol  
*kata negatif* = bilangan bulat positif atau nol

#### 2. 6. 2. Pemberian kelas sentimen

Setelah melakukan proses perhitungan skor sentimen dan diketahui nilai skor dari suatu kalimat. Proses selanjutnya dalam pelabelan dengan pendekatan kamus sentimen adalah pemberian kelas pada kalimat(*tweet*) berdasarkan skor. Jika kalimat mempunyai skor  $> 0$  akan masuk ke dalam kelas positif, jika kalimat mempunyai skor  $< 0$  akan masuk ke dalam kelas negatif (Santoso and Nugroho, 2019) (Buntoro, 2017) (Nurulbaiti and Retno Subekti, 2020), sedangkan jika kalimat mempunyai skor  $= 0$  maka akan diabaikan sehingga penentuan kelas sentimen dilakukan secara manual. Adapun proses pelabelan dapat dilihat pada Gambar 2.1 di bawah ini:



Gambar 2.1 Pelabelan kelas sentimen

#### 2. 7. *CountVectorizer*

*CountVectorizer* merupakan proses pengolahan dokumen atau teks menjadi bentuk vektor. *CountVectorizer* digunakan untuk menghitung frekuensi kata dalam dokumen atau kalimat kemudian direpresentasikan ke dalam bentuk vektor (Munawar, 2019).

#### 2. 8. *Modeling*

*Modeling* merupakan proses pembuatan pengetahuan berdasarkan data latih yang telah tersedia. Data latih yang dijadikan *model* dipilih dengan teknik sampling kuota (*quota sampling*). *Quota Sampling* merupakan teknik sampling yang menentukan jumlah sampel dari populasi yang memiliki ciri

atau kriteria tertentu hingga jumlah kuota yang diinginkan tercapai (Sari and Wibowo, 2019).

## 2. 9. *K-Nearest Neighbor*

*K-Nearest Neighbor* (KNN) adalah algoritme klasifikasi *supervised learning* berbasis jarak. Algoritme ini bekerja dengan cara membandingkan jarak antara data uji dengan semua data latih yang ada (Romadloni, Santoso and Budilaksono, 2019) (Septian, Fahrudin and Nugroho, 2019) (Daeli and Adiwijaya, 2020). Untuk menghitung jarak antara data digunakan perhitungan *euclidean distance* dengan rumus:

$$d_{(x,y)} = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (2. 2)$$

Keterangan:

$D_{(x,y)}$	= Jarak antara data uji dengan data latih
$n$	= jumlah fitur
$X_i$	= Fitur ke- $i$ dalam data uji
$Y_i$	= Fitur ke- $i$ dalam data latih

Proses selanjutnya setelah menghitung jarak untuk setiap data latih adalah mencari data latih dengan nilai jarak terkecil (ketetanggaan terdekat) sebanyak nilai  $K$  yang telah ditentukan. Proses akhir setelah ditemukannya data tetangga terdekat adalah pemungutan suara (*voting*). *Voting* bertujuan untuk menentukan kelas atau label dari suatu data uji (Daeli and Adiwijaya, 2020).

## 2. 10. Pengujian dan Evaluasi

Pengujian merupakan hal penting untuk memastikan bahwa suatu algoritma yang telah dirancang dapat berjalan sesuai dengan harapan. Pengujian klasifikasi sentimen dilakukan dengan menguji aplikasi yang telah dibangun dengan membandingkan antara data prediksi dan data aktual. Data prediksi berupa hasil klasifikasi yang dihasilkan oleh aplikasi yang dibangun, sedangkan data aktual berupa yang didapatkan melalui proses pelabelan (Wahid and SN, 2017). Dalam penelitian ini, pengujian dilakukan pada sebuah *model* terhadap data uji yang tersedia. Hasil dari pengujian tersebut akan dievaluasi menggunakan *confusion matrix* untuk mengukur tingkat akurasi, presisi dan *recall*. Ilustrasi *confusion matrix* dapat dilihat pada Tabel 2.1 berikut:

**Tabel 2.1 Confusion matrix**

		Nilai Aktual	
		TRUE ( <i>positive</i> )	FALSE ( <i>negative</i> )
Nilai Prediksi	TRUE ( <i>positive</i> )	TP ( <i>True Positive</i> )	FP ( <i>False Positif</i> )
	FALSE ( <i>negative</i> )	FN ( <i>False Negative</i> )	TN ( <i>True Negative</i> )

## 2. 11. Studi Literatur

Berdasarkan landasan teori yang telah dijelaskan, terdapat penelitian yang sudah ada sebelumnya, dirangkum dalam Tabel 2.2 berikut :

**Tabel 2.2 Studi Literatur**

No	Penulis	Judul	Terbitan	Deskripsi
1	Ronal Watrianthos	Analisis Pembelajaran Daring di Era Pandemic Covid-19	Green Press, Hal 55-64, 2018, P-ISBN: 978-623-93614-2-6, e-ISBN: 978-623-93614-3-3	Melakukan analisis terhadap pembelajaran daring melalui sosial media Twitter, berdasarkan kata kunci pada tanggal 1 Juli - 31 Juli 2020. Menggunakan metode analisis sentimen dengan Naive Bayes. Hasil analisis menunjukkan sentimen negatif sangat tinggi mencapai 83%; 16% sentimen positif; 1% sentimen netral dan pada periode Juli 2020.
2	Siti Mujilawati	<i>Pre-Processing Text Mining</i> Pada Data Twitter	Seminar Nasional Teknologi Informasi dan Komunikasi 2016 (SENTIK A 2016), Maret 2016,	Melakukan penelitian mengenai teknik penanganan data <i>tweet</i> (Twitter) dengan pre-processing. Hasil penelitian kemudian diuji sebagai bahan pengklasifikasian layanan perusahaan telekomunikasi dan didapatkan hasil akurasi mencapai

No	Penulis	Judul	Terbitan	Deskripsi
			ISSN: 2089-9815	93,11% dengan 450 data uji.
3	Eko Budi Santoso, Aryo Nugroho	Analisis Sentimen Calon Presiden Indonesia 2019 Berdasarkan Komentar Publik di Facebook	Jurnal Eksplora Informatika, Vol. 9, No. 1, Hal 60-69, September 2019, P-ISSN: 2089-1814, e-ISSN: 2460-3694	Melakukan analisis komentar masyarakat pada media sosial Facebook terhadap popularitas dari seorang calon presiden. Metode klasifikasi yang digunakan adalah Naive Bayes disertai dengan proses asosiasi teks, juga menggunakan fitur kamus ( <i>lexicon</i> ) pada proses pelabelan kelas sentimen. Penelitian ini menghasilkan persentase sentimen (positif dan negatif) tiap pasangan calon presiden dan serta pengujian akurasi untuk metode Naive Bayes Classifier yaitu sebesar 86,4%.
4	Fransiska Vina Sari, Arief Wibowo	Analisis Sentimen Pelanggan Toko Online JD.Id Menggunakan Metode Naive Bayes Classifier Berbasis Konversi Ikon Emosi	Jurnal SIMETRI S, Vol. 10, No. 2, November 2019, P-ISSN: 2252-4983, e-ISSN: 2549-3108	Melakukan analisis terhadap opini pelanggan atau konsumen terkait toko online JD.id. Menggunakan data yang bersumber pada media sosial Twitter dengan metode klasifikasi Naive Bayes dan pembobotan TF-IDF disertai fitur konversi ikon emoji ( <i>emoticon</i> ). Hasil penelitian menunjukkan bahwa

No	Penulis	Judul	Terbitan	Deskripsi
				metode Naïve Bayes tanpa penambahan fitur mampu mengklasifikasi sentimen dengan nilai akurasi sebesar 96,44%, sementara jika ditambahkan fitur pembobotan TF-IDF disertai konversi ikon emosi mampu meningkatkan nilai akurasi menjadi 98%.
5	Novelty Octaviani Faomasi Daeli, Adiwijaya	<i>Sentiment Analysis on Movie Reviews Using Information Gain and K-Nearest Neighbor</i>	J. Data SCI APPL, Vol. 3, No. 1, Hal. 001-007, Januari 2020, e-ISSN 2614-7408	Melakukan pengujian untuk mencari nilai K yang optimal untuk K-Nearest Neighbor (KNN) dengan perhitungan jarak euclidean distance. Dataset yang digunakan adalah dataset review film Cornell Polarity v2.0 dengan total data 1000 dokumen negatif dan 1000 dokumen positif. Dengan melibatkan Information Gain, nilai K optimal yang diperoleh untuk KNN adalah 3 (K=3) dengan memberikan akurasi sebesar 96.8%.
6	Nova Tri Romadloni, Imam Santoso, Sularso Budilaksono	Perbandingan Metode Naive Bayes, KNN Dan Decision Tree Terhadap Analisis Sentimen Transportasi	Jurnal IKRA-ITH Informatika, Vol. 3, No. 2, Juli 2019, ISSN: 2580-4316	Melakukan perbandingan metode Naive Bayes, K-Nearest Neighbor (KNN) dan Decision Tree untuk melakukan analisis sentimen pada data media sosial Twitter. Pengujian dilakukan terhadap



No	Penulis	Judul	Terbitan	Deskripsi
		KRL Commuter Line		127 data yang telah diberikan label positif atau negatif, menghasilkan akurasi 80% menggunakan Naive Bayes; 80% menggunakan KNN; 100% menggunakan Decision Tree.
7	Muhammad Syarifuddin	Analisis Sentimen Opini Publik Mengenai Covid-19 Pada Twitter Menggunakan Metode Naïve Bayes Dan KNN	Inti Nusa Mandiri, Vol. 15, Agustus 2020, P-ISSN: 0216-6933, e-ISSN: 2685-807X	Melakukan analisis pendapat masyarakat yang bersumber dari media sosial Twitter. Menggunakan 1098 tweet dengan kata kunci Covid-19, memperoleh nilai akurasi tertinggi menggunakan metode Naive Bayes sebesar 63.21% sedangkan metode KNN sebesar 58.10%, dan kecenderungan opini masyarakat di Twitter condong ke positif dengan jumlah opini positif sebesar 610 sedangkan negatif 488.
8	Ghulam Asrofi Buntoro	Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter	Integer Journal, Vol 2, No 1, Maret 2017, Hal. 32-41, P-ISSN: 2477-5274, e-ISSN: 2579-566X	Melakukan analisis terkait opini masyarakat terhadap pemilihan gubernur DKI Jakarta tahun 2017 pada media sosial Twitter. Proses penentuan sentimen menggunakan metode <i>Lexicon-Based</i> dan proses klasifikasinya menggunakan metode Naïve Bayes Classifier (NBC) dan Support Vector Machine

No	Penulis	Judul	Terbitan	Deskripsi
				(SVM). Akurasi tertinggi didapat saat menggunakan metode klasifikasi Naïve Bayes Classifier (NBC), dengan nilai rata-rata akurasi mencapai 95%, nilai presisi 95%, nilai recall 95% nilai TP rate 96,8% dan nilai TN rate 84,6%.
9	Walaa Medhat, Ahmed Hassan, Hoda Korashy	<i>Sentiment Analysis Algorithms and Applications: A Survey</i>	Ain Shams Engineering Journal, Vol 5, No. 4, Hal. 1093–1113, Desember 2014, <a href="https://doi.org/10.1016/j.asej.2014.04.011">https://doi.org/10.1016/j.asej.2014.04.011</a>	Melakukan penelitian terkait analisis sentimen. Meliputi proses melakukan analisis sentimen menggunakan pendekatan <i>machine learning</i> dan <i>lexicon based</i> . Penelitian ini juga membahas macam-macam teknik klasifikasi sentimen dan cara pengaplikasiannya secara singkat untuk mengolah data teks.
10	Jeremy Andre Septian, Tresna Maulana Fahrudin, Aryo Nugroho	Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor	Journal of Intelligent Systems And Computation, Vol. 1 No. 1, Oktober 2019, P-ISSN: 2621-9220, e-ISSN: 2722-1962	Melakukan analisis sentimen pada setiap kalimat dari pengguna twitter terhadap persepakbolaan Indonesia apakah memiliki sentimen negatif atau positif menggunakan K-Nearest Neighbor (KNN) dengan pembobotan kata TF-IDF. Data yang digunakan dalam didapatkan dari hasil crawling dari media sosial twitter terkait

No	Penulis	Judul	Terbitan	Deskripsi
				persepakbolaan di Indonesia yang diambil dari akun twitter resmi PSSI. Dari 2000 data tweet berbahasa indonesia didapatkan hasil akurasi optimal pada nilai $k=23$ sejumlah 79.99%.
11	Sitti Nurul Jannah Fitriyyah, Novi Safriadi, Enda Esyudha Pratama	Analisis Sentimen Calon Presiden Indonesia 2019 dari Media Sosial Twitter Menggunakan Metode Naive Bayes	JEPIN (Jurnal Edukasi dan Penelitian Informatika) Vol. 5, No. 3, Desember 2019, P-ISSN: 2460-0741, e-ISSN: 2548-9364	Melakukan analisis sentimen terhadap pasangan calon(paslon) presiden melalui media sosial Twitter. Penelitian ini juga melakukan penerapan metode Naive Bayes untuk klasifikasi sentimen pengguna Twitter dengan dua kelas sentimen (negatif, positif) dan tiga kelas sentimen (negatif, positif, netral). Hasil dari penelitian ini menunjukkan metode Naive Bayes memiliki performa lebih baik dalam mengklasifikasikan 2 kelas sentimen (negatif, positif) dibandingkan pengujian dengan 3 kelas sentimen.
12	Agus Sasmito Aribowo	Analisis Sentimen Publik pada Program Kesehatan Masyarakat menggunakan Twitter	Seminar Nasional Informatika Medis, Hal. 17-23, 2018, ISSN: 9-	Melakukan penelitian untuk mengembangkan model untuk mengetahui sentimen publik terhadap enam macam program kebijakan pemerintah

No	Penulis	Judul	Terbitan	Deskripsi
		<i>Opinion Mining</i>	772301-936005	yaitu imunisasi, asuransi kesehatan, stunting, gizi buruk, pelayanan kesehatan, dan jaminan kesehatan masyarakat. Metodenya adalah dengan melakukan ekstraksi pengetahuan dari opini di media sosial menggunakan analisis sentimen berbasis leksikon. Dataset yang diperoleh dalam kurun waktu 3 - 9 Agustus 2018 sebanyak total 3311 data. Hasil penelitian berupa sentimen yang dituangkan ke dalam bentuk grafik.
13	Bing Liu, Mingqing Hu, Junsheng Cheng	<i>Opinion Observer: Analyzing and Comparing Opinions on the Web</i>	Proceedings of the 14th International World Wide Web Conference (WWW-2005), May 10-14, Chiba, Japan	Melakukan penelitian untuk menganalisis pendapat konsumen terhadap suatu produk. Mengelompokkan data pendapat berdasarkan ulasan konsumen ke dalam bentuk ulasan positif atau negatif, kemudian dijadikan sebuah pengetahuan untuk dibandingkan dengan ulasan lainnya. Penelitian ini juga membuahkan daftar kata positif dan negatif yang dapat digunakan kembali untuk proses klasifikasi pendapat.

No	Penulis	Judul	Terbitan	Deskripsi
14	Adhi Viky Sudiantoro, Eri Zuliarso	Analisis Sentimen Twitter Menggunakan Text Mining Dengan Algoritma Naïve Bayes Classifier	Dinamika Informatika Vol.10, No.2, Oktober 2018, Hal. 69-73, P- ISSN: 2085- 3343, e- ISSN : 2714-8769	Melakukan analisis dengan tujuan untuk mengklasifikasi data tweet menjadi dua sentimen yaitu positif dan negatif. Dataset bersumber dari tweet teks berbahasa Indonesia yang terdapat di sosial media Twitter, kemudian digunakan sebagai bahan analisis sentimen untuk mengetahui sentimen masyarakat terhadap pilkada Jawa Barat. Hasil pengujian akurasi terhadap 100 data uji, Naïve Bayes Classifier memberikan nilai akurasi sebesar 84%.

### BAB III METODOLOGI PENELITIAN

#### 3.1. Data Penelitian

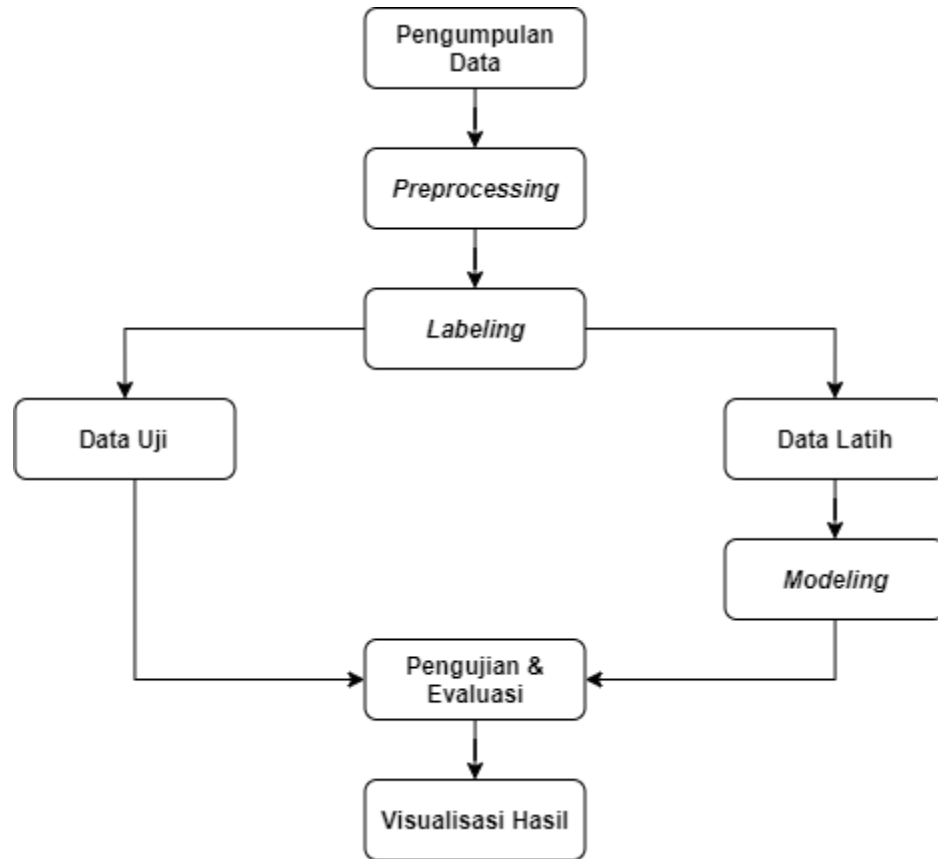
*Dataset* atau data yang digunakan dalam penelitian ini bersumber dari Twitter berupa data teks kicauan (*tweet*) yang diperoleh mulai tanggal 1 Desember 2020 hingga 31 Desember 2020 sejumlah 4.314 data. Data tersebut diperoleh menggunakan pustaka Tweepy melalui proses *crawling*. *Dataset* yang diperoleh dikumpulkan berdasarkan enam (6) parameter kata kunci yang terkait dengan sistem pembelajaran daring antara lain: 'pembelajaran jarak jauh', 'belajar dari rumah', '#belajaronline', '#belajardarirumah', '#belajardirumah', dan '#kuliahonline'. Berikut beberapa contoh data *tweet* hasil dari proses *crawling* yang dapat dilihat pada Tabel 3.1 berikut:

**Tabel 3.1 Sampel data tweet**

<i>Tweet ID</i>	<i>Username</i>	<i>Tweet</i>	Waktu <i>tweet</i>
13359893547 92103936	LRomdani	Tetap memakai masker meski dirumah sendiri Tetap semangat belajar dari rumah dimasa pandemi #DiktiMengajarDariRumah #DiktiDutaEdukasiPerubahan Prilaku <a href="https://t.co/c1WMa5SVSj">https://t.co/c1WMa5SVSj</a>	2020-12- 07 16:47:38
13365204602 55724032	kelaskitadot com	Gunakan hak suara kamu dengan bijak, ya! Selamat memilih! #kelaskita #carabarubelajarseru #belajardirumah #elearning #belajaronline #dirumahaja #quotes <a href="https://t.co/1anyTiETIA">https://t.co/1anyTiETIA</a>	2020-12- 09 03:58:03
13367424941 22340096	fandimas16	@collegemenfess 1. Jenuh banget di rumah 2. Gw dri dulu suka ama suasana kelas, dan suasana itu mendukung gw untuk belajar dan memahami suatu materi	2020-12- 09 18:40:20
13380037305 87812096	kumparan	Tanpa smartphone di masa pandemi, bisa berarti putus sekolah, karena kini dilakukan belajar online atau pembelajaran jarak jauh. <a href="https://t.co/rVW6xOgrfI">https://t.co/rVW6xOgrfI</a>	2020-12- 13 06:12:02

### 3. 2. Penerapan Metode

Dalam membangun aplikasi analisis sentimen yang dilakukan pada penelitian ini, terdapat beberapa tahapan yang dilakukan. Tahapan tersebut merepresentasikan setiap proses dan rancangan dalam penelitian, dari awal hingga akhir aplikasi berjalan. Tahapan yang dilakukan dapat dilihat pada Gambar 3.1 berikut:



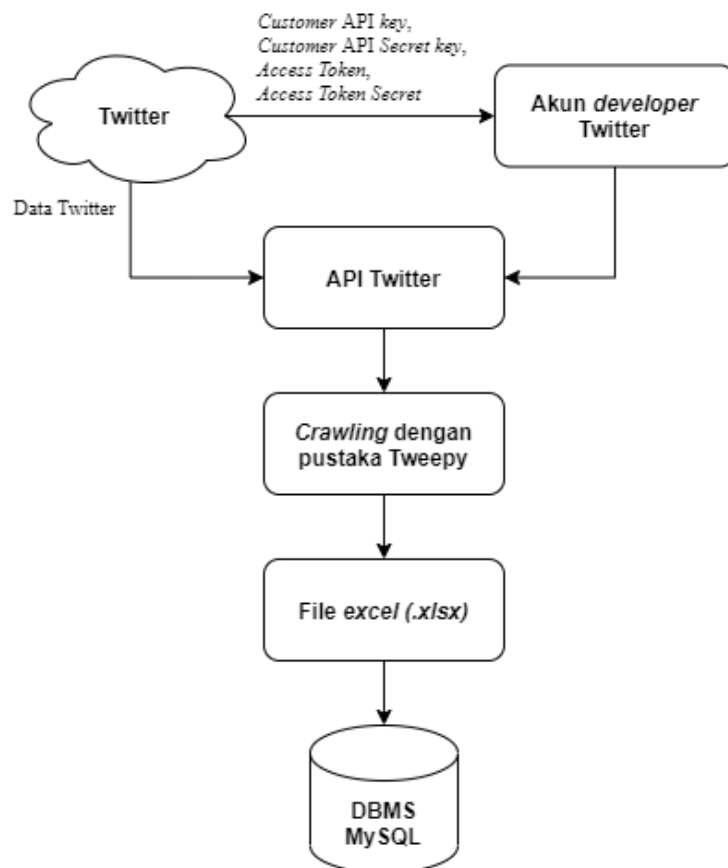
Gambar 3.1 Tahapan metode

Pada Gambar 3.1, pengumpulan data dilakukan melalui proses *crawling* untuk mendapatkan *dataset* berupa kicauan (*tweet*). Selanjutnya, *tweet* yang telah diperoleh dalam bentuk *excel*, kemudian dimasukkan ke dalam basis data (*database*) untuk dilakukan proses *preprocessing*, pada proses *preprocessing* dilakukan penyaringan, pembuangan dan perbaikan kata. Hasil dari proses *preprocessing* menghasilkan kalimat yang lebih terstruktur (*clean text*) yang kemudian digunakan pada tahap selanjutnya. *Clean text* yang diperoleh dari proses *preprocessing* akan diproses dalam tahap *labeling* untuk menentukan kelas (*label*) berupa sentimen positif atau negatif, kemudian *tweet* yang telah berlabel akan dibagi menjadi dua (2) buah bagian antara lain: data uji dan data latih. Data latih merupakan data yang berfungsi sebagai pembangun pengetahuan untuk proses klasifikasi, proses pembangunan pengetahuan tersebut dilakukan melalui proses *modeling* dan menghasilkan sebuah model latih menggunakan data latih

yang tersedia. Sementara data uji merupakan data yang disiapkan untuk menguji tingkat keakuratan model latih yang dihasilkan oleh proses *modeling*, proses pengujian tingkat akurasi tersebut dilakukan melalui proses pengujian dan evaluasi. Setelah dilakukan proses pengujian dan evaluasi, hasil penerapan metode akan divisualisasikan ke dalam bentuk persentase dan grafik.

### 3. 2. 1. Pengumpulan data

Pada tahapan pengumpulan data dilakukan melalui proses *crawling*. Proses tersebut meliputi: mendapatkan API key Twitter melalui akun *developer* Twitter (<https://developer.twitter.com/>). API key Twitter yang diperoleh antara lain: *Customer API key*, *Customer API Secret key*, *Access Token*, dan *Access Token Secret*. Proses selanjutnya adalah penambangan data yang bersumber pada media sosial Twitter menggunakan pustaka Tweepy dengan akses dari API key yang telah didapatkan. Data *tweet* yang berhasil dikumpulkan akan disimpan ke dalam sebuah file *excel* (.xlsx), yang kemudian dimasukkan ke dalam basis data (*database*) MySQL. Ilustrasi tahap pengumpulan data dapat dilihat pada Gambar 3.2 berikut:

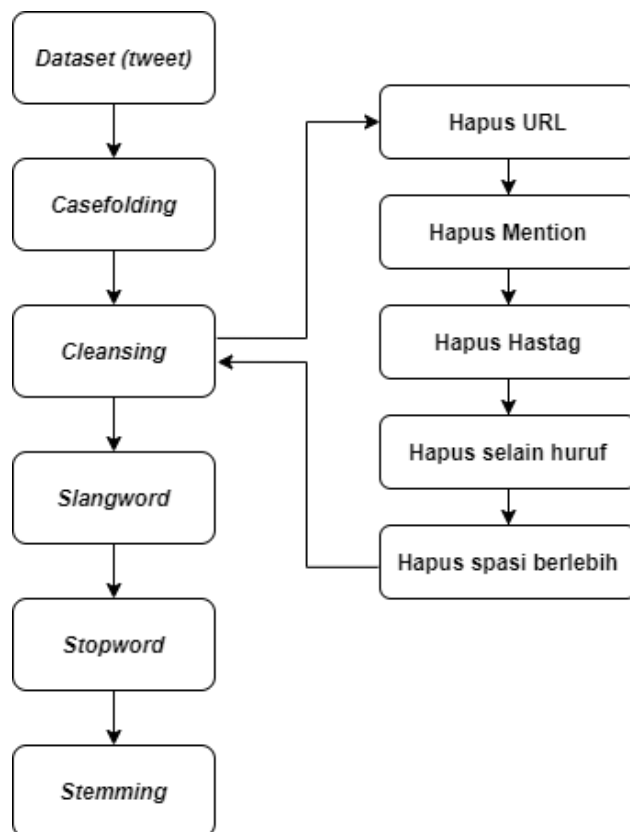


Gambar 3.2 Tahap pengumpulan data



### 3. 2. 2. *Preprocessing*

Pada tahapan *preprocessing* dilakukan penyaringan, pembuangan dan perbaikan kata melalui beberapa proses. Hal tersebut dimaksudkan untuk menghasilkan data kicauan (*tweet*) yang lebih terstruktur atau disebut dengan *clean text*. Berdasarkan teori yang telah dijelaskan pada sub bab (2.5), proses yang dilakukan dalam tahap *preprocessing* antara lain: *case folding*, *cleansing*, mengubah *slang word*, menghapus *stop word*, dan *stemming*. Ilustrasi proses tahap *preprocessing* data dilihat pada Gambar 3.3 berikut:



Gambar 3.3 Tahap *preprocessing*

#### a. *Case folding*

Proses *case folding* dilakukan penyetaraan teks menjadi huruf kecil (*lowercase*) secara keseluruhan, misalnya kata 'BELAJAR' atau 'Belajar', maka akan diubah menjadi 'belajar'.

#### b. *Cleansing*

Pada proses *cleansing* dilakukan penyaringan dan pembuangan teks. Proses *cleansing* terdiri atas beberapa tahapan antara lain: menghapus URL, *mention* (@*mention*), *hashtag* (#*hashtag*), selain huruf (a-z) dan spasi berlebih.

1) Menghapus URL

Proses penghapusan URL akan menghapus semua teks yang diawali dengan kata 'http', misalnya: 'https://google.com', maka akan diubah menjadi ' '.

2) Menghapus *mention* (@*mention*)

Proses penghapusan *mention* (@*mention*) akan menghapus semua teks yang diawali dengan '@', misalnya: '@kominfo', maka akan diubah menjadi ' '.

3) Menghapus *hashtag* (#*hashtag*)

Pada Gambar 3.6 proses penghapusan tagar atau *hashtag* (#*hashtag*) akan menghapus semua teks yang diawali dengan '#', misalnya: '#pjj', maka akan diubah menjadi ' '.

4) Menghapus selain huruf

Proses penghapusan selain huruf akan menghapus karakter selain huruf a-z pada teks, misalnya terdapat teks: 'tanggal: 01 desember 2020', maka akan diubah menjadi 'tanggal desember '.

5) Menghapus spasi berlebih

Proses penghapusan spasi atau (*whitespace*) berlebih akan menghapus *whitespace* berlebih pada teks, misalnya terdapat teks: 'tanggal desember ', maka akan diubah menjadi 'tanggal desember'.

c. Merubah *slang word*

Proses merubah *slang word* akan merubah setiap kata gaul, kata singkatan atau kata tidak baku ke bentuk bakunya, misalnya kata: 'utk' menjadi 'untuk', 'yng' menjadi 'yang', 'apotik' menjadi 'apotek'. Proses pengubahan tersebut melibatkan kamus *slang word* yang terdapat dalam basis data (*database*), diperoleh melalui penelitian sebelumnya yang dipublikasikan pada laman github ([https://github.com/louisowen6/NLP\\_bahasa\\_resources](https://github.com/louisowen6/NLP_bahasa_resources)).

d. Menghapus *stop word*

Proses menghapus *stop word* akan menghapus setiap kata yang kurang memiliki makna namun sering dijumpai dalam sebuah teks, misalnya penghapusan kata: 'untuk', 'yang', 'apa'. Proses penghapusan tersebut melibatkan kamus *stop word* yang terdapat dalam basis data (*database*), diperoleh melalui penelitian sebelumnya yang dipublikasikan pada laman github ([https://github.com/louisowen6/NLP\\_bahasa\\_resources](https://github.com/louisowen6/NLP_bahasa_resources)).

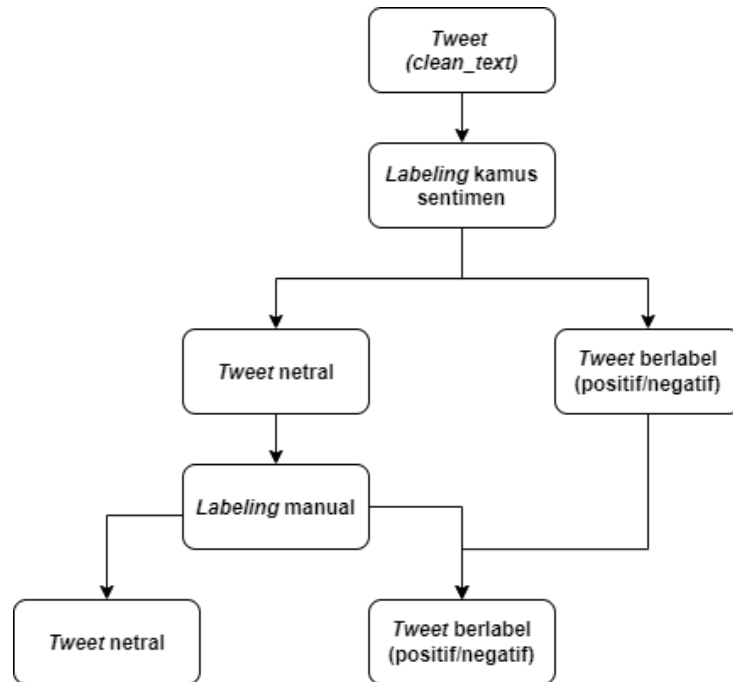
e. *Stemming*

Proses *stemming* akan mengubah kata berimbuhan menjadi kata dasar dengan melibatkan pustaka Sastrawi, misalnya kata: ‘membantu’ menjadi ‘bantu’, ‘pembelajaran’ menjadi ‘ajar’. Proses pengubahan tersebut dilakukan dengan melibatkan pustaka Sastrawi, menggunakan paket StemmerFactory.

3. 2. 3. *Labeling*

Pada tahapan *labeling* dilakukan pemberian *label* (kelas) berdasarkan ciri atau karakteristik yang terkandung dalam sebuah dokumen atau kalimat. Pada tahap ini, *tweet* yang telah melalui proses *preprocessing* dan menghasilkan *clean text* akan diberikan kelas positif atau negatif. Kelas positif dimaksudkan untuk teks *tweet* tersebut yang mengandung pernyataan yang setuju, mendukung atau menerima proses berjalanya pembelajaran daring. Sedangkan kelas negatif dimaksudkan untuk teks *tweet* yang cenderung tidak setuju atau menolak proses berjalanya pembelajaran daring.

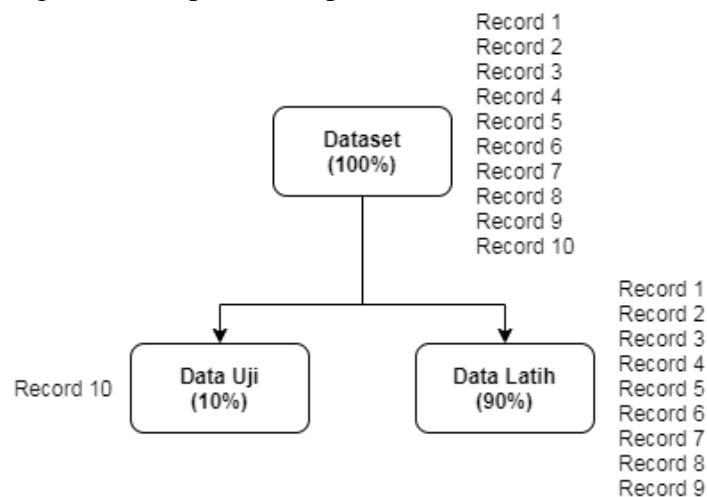
Berdasarkan teori yang telah dijelaskan dalam sub bab (2. 6), bahwa proses *labeling* dapat dilakukan dengan dua (2) buah cara, antara lain: *labeling* manual dan *labeling* dengan kamus sentimen. *Labeling* manual merupakan proses pemberian kelas berdasarkan subjektivitas beberapa orang. Sedangkan *labeling* dengan kamus sentimen merupakan proses pemberian kelas secara otomatis berdasarkan kamus sentimen, dimana prosesnya melibatkan kamus kata positif dan kamus kata negatif yang terdapat dalam basis data (*database*), diperoleh melalui penelitian sebelumnya yang dipublikasikan pada laman github (<https://github.com/masdevid/ID-OpinionWords>). Proses *labeling* utama dilakukan dengan cara *labeling* dengan kamus sentimen, kemudian untuk *tweet* yang gagal mendapatkan *label* (*tweet* netral) akan dievaluasi ulang menggunakan cara *labeling* manual berdasarkan subjektivitas beberapa orang, dengan mengabaikan *tweet* yang dinilai sebagai netral. Ilustrasi tahap *labeling* dapat dilihat pada Gambar 3.4 berikut:



**Gambar 3.4 Tahap labeling**

#### 3. 2. 4. Pembagian data

Pada tahapan pembagian data, *tweet* yang telah berlabel akan dibagi menjadi dua (2) buah bagian antara lain: data uji dan data latih. Proses pembagian data dilakukan dengan membagi *dataset* menjadi 90% data latih dan 10% menjadi data uji. Ilustrasi tahap pembagian data dapat dilihat pada Gambar 3.5 berikut:



**Gambar 3.5 Tahap pembagian data**

##### a. Data Latih

Data latih merupakan data yang berfungsi sebagai *model* latih, yaitu pembangun pengetahuan untuk proses klasifikasi.

b. Data Uji

Data uji merupakan data yang disiapkan untuk menguji tingkat keakuratan algoritme klasifikasi berdasarkan *model* latih.

3. 2. 5. *Modeling*

Tahap modeling dilakukan untuk mengekstraksi *tweet* data latih menjadi representasi vektor menggunakan *CountVectorizer*. Pada tahap ini terdapat lima (5) proses utama yang dilalui untuk menghasilkan sebuah *model* latih, lima proses itu antara lain: seleksi data latih, pembuatan list kata, pencarian fitur kata, pembuatan vektor kosong dan membuat representasi vektor. Ilustrasi proses tersebut dapat dilihat pada Gambar 3.6 berikut:



**Gambar 3.6 Tahap *modeling***

Pada Gambar 3.6, penyeleksian data latih dilakukan dengan menggunakan teknik *sampling* kuota (*quota sampling*), dimaksudkan untuk mendapatkan keseimbangan pada data latih. Tahapan seleksi tersebut dilakukan sebanyak kriteria tertentu. Data latih telah terseleksi akan diproses (ekstraksi fitur) menggunakan *CountVectorizer* sehingga dapat diperoleh daftar fitur dan

representasi vektor angka untuk tiap data latih. Daftar fitur dan vektor angka tersebut kemudian akan disimpan dan dijadikan sebagai sebuah *model* pengetahuan (*model* latih) dalam bentuk file JSON (.json).

### 3. 2. 6. Klasifikasi *K-Nearest Neighbor*

Tahap klasifikasi menggunakan *K-Nearest Neighbor* (KNN) merupakan tahapan yang dapat dilakukan setelah terdapat satu atau lebih *model* latih. *Model* latih tersebut merupakan data latih yang telah melalui tahap *modeling* yang dijelaskan pada sub-sub bab (3. 2. 5). Untuk menerapkan *model* klasifikasi menggunakan KNN, terdapat empat (4) buah proses utama yaitu: membuat representasi vektor uji, menghitung jarak antar data, mencari tetangga terdekat berdasarkan nilai K, dan menghitung nilai probabilitas *label* sentimen. Ilustrasi tahapan klasifikasi dapat dilihat pada Gambar 3.7 berikut:



Gambar 3.7 Tahap klasifikasi

Pada Gambar 3.7, pembuatan representasi vektor uji dilakukan menggunakan *model* latih yang dipilih, sehingga terbentuk representasi vektor uji yang sesuai dengan pengetahuan *model* latih. Hasil vektor uji tersebut akan dihitung tingkat kedekatannya (jarak) dengan vektor pada *model* latih, proses tersebut melibatkan perhitungan *euclidean distance*. Hasil perhitungan *euclidean distance* akan menghasilkan nilai jarak, yang kemudian akan

disaring berdasarkan K tetangga terdekat. Selanjutnya dilakukan *voting* untuk menentukan *label* prediksi (positif atau negatif) berdasarkan dominasi *label* pada K tetangga terdekatnya.

### 3.3. Rancangan Pengujian

Pengujian dilakukan untuk mengetahui nilai atau tingkat akurasi, presisi, dan *recall* dari *model* latih menggunakan algoritme yang diusulkan. Pada penelitian ini, pengujian dilakukan dengan cara membandingkan beberapa data hasil prediksi (data hasil tahap klasifikasi) dengan sekumpulan data aktual (data hasil tahap *labeling*). Adapun dimaksud dengan beberapa data hasil prediksi merupakan sekumpulan data yang telah diproses melalui algoritme *K-Nearest Neighbor* (KNN) dengan variasi nilai K, yaitu: K=3, K=5, K=7, K=9, dan K=11.

#### 3.3.1. Akurasi

Akurasi merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual persamaan (3. 1).

#### 3.3.2. Presisi

Presisi merupakan tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan oleh sistem persamaan (3. 2).

#### 3.3.3. Recall

*Recall* merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi persamaan (3. 3).

Berdasarkan pada sub bab (2. 10), pengukuran tingkat akurasi, presisi, dan *recall* dapat diketahui melalui *confusion matrix* dengan persamaan sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3. 1)$$

$$Presisi = \frac{TP}{TP + FP} \quad (3. 2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3. 3)$$

dengan,

- a. *True Positive* (TP) merupakan data positif yang diprediksi benar. Misalnya: *tweet* 1 berlabel positif dan dari *model* latih yang dibuat memprediksi *tweet* 1 bernilai positif juga.
- b. *True Negative* (TN) merupakan data negatif yang diprediksi benar. Misalnya: *tweet* 1 berlabel negatif dan dari *model* latih yang dibuat memprediksi *tweet* 1 bernilai negatif juga.

- c. *False Positive* (FP) merupakan data negatif namun diprediksi sebagai data positif. Misalnya: *tweet* 1 berlabel negatif namun dari *model* latihan yang dibuat memprediksi *tweet* 1 bernilai positif.
- d. *False Negative* (FN) merupakan data positif namun diprediksi sebagai data negatif. Misalnya: *tweet* 1 berlabel positif namun dari *model* latihan yang dibuat memprediksi *tweet* 1 bernilai negatif.



## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4. 1. Lingkungan Percobaan**

Agar aplikasi yang telah dikembangkan dapat berjalan dengan semestinya, dibutuhkan perangkat dengan spesifikasi tertentu, adapun dalam penelitian ini menggunakan spesifikasi perangkat diantaranya.

##### **4. 1. 1. Spesifikasi perangkat keras**

Daftar perangkat keras yang mendukung aplikasi ini untuk berjalan dengan baik adalah sebagai berikut:

- a. Processor : Intel(R) Core(TM) i3 CPU M 380 @ 2.53GHz
- b. RAM : 2,00 GB
- c. Harddisk : 500 GB
- d. VGA : Intel(R) HD Graphics

##### **4. 1. 2. Spesifikasi perangkat lunak**

Daftar perangkat lunak yang mendukung aplikasi ini untuk berjalan dengan baik adalah sebagai berikut:

- a. Sistem Operasi : Windows 7 Professional
- b. Bahasa Program Utama : Python 3.8 (32-bit)
- c. IDE : Visual Studio Code v1.52.1
- d. DBMS : MySQL Database
- e. Browser : Google Chrome
- f. Lainnya : XAMPP v7.3.9, Ms. Excel 2013

#### **4. 2. Implementasi Metode**

Implementasi metode dalam penelitian ini dilakukan dengan enam (6) tahapan utama. Tahapan utama tersebut diproses secara berurutan, tahapan utama yang dimaksud antara lain: Tahap pengumpulan data, tahap *preprocessing*, tahap *labeling*, tahap pembagian data, tahap ekstraksi fitur *CountVectorizer* dan tahap klasifikasi *K-Nearest Neighbor*.

##### **4. 2. 1. Tahap pengumpulan data**

Berdasarkan sub bab (3. 1), *dataset* penelitian bersumber dari media sosial Twitter berupa data teks kicauan (*tweet*). *Dataset* tersebut diperoleh secara rutin melalui proses *crawling* menggunakan pustaka Tweepy, dimulai pada tanggal 1 Desember 2020 hingga 31 Desember 2020, sehingga diperoleh total *dataset* sejumlah 4.314 data (*tweet*). Kata kunci yang digunakan antara lain: ‘pembelajaran jarak jauh’, ‘belajar dari rumah’, ‘#belajaronline’, ‘#belajardarirumah’, ‘#belajardirumah’, dan ‘#kuliahonline’.

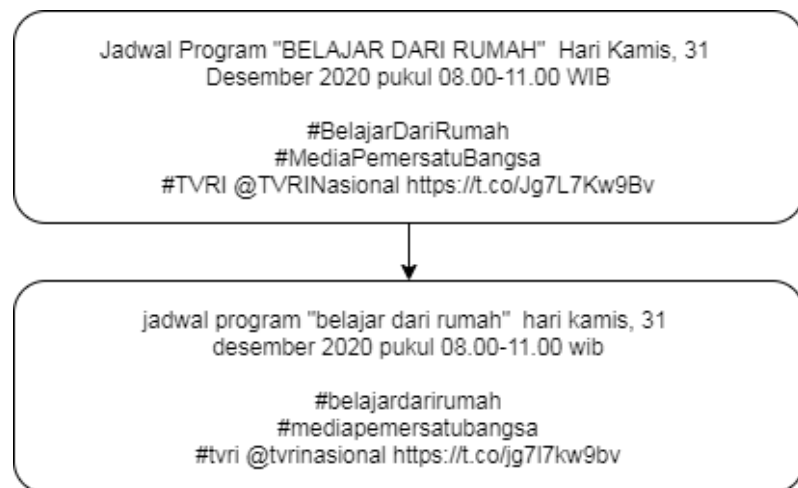
*Dataset* yang berhasil dikumpulkan melalui proses *crawling* dengan informasi antara lain: *tweet id*, *full\_text (tweet)*, *created\_at*, *user.screen\_name (username)*, akan disimpan ke dalam file *excel (.xlsx)*, yang kemudian akan dimasukkan ke dalam basis data (*database*) MySQL untuk tahap selanjutnya (*preprocessing*).

#### 4. 2. 2. Tahap *preprocessing*

Tahap *preprocessing* merupakan tahapan yang hanya dapat dilakukan setelah tersedianya satu atau lebih *dataset* pada basis data (*database*) hasil dari tahapan pengumpulan data. Berdasarkan pada sub sub bab (3. 2. 2), tahapan ini terdiri atas lima (5) proses utama antara lain: *case folding*, *cleansing*, mengubah *slang word*, menghapus *stop word*, dan *stemming*. Berikut penjabaran dari tahap *preprocessing*:

##### a. Case folding

Pada Gambar 4.1 proses *case folding* dilakukan penyetaraan teks menjadi huruf kecil secara keseluruhan, misalnya: 'Jadwal' akan diubah menjadi 'jadwal', 'BELAJAR DARI RUMAH' akan diubah menjadi 'belajar dari rumah', dan seterusnya.



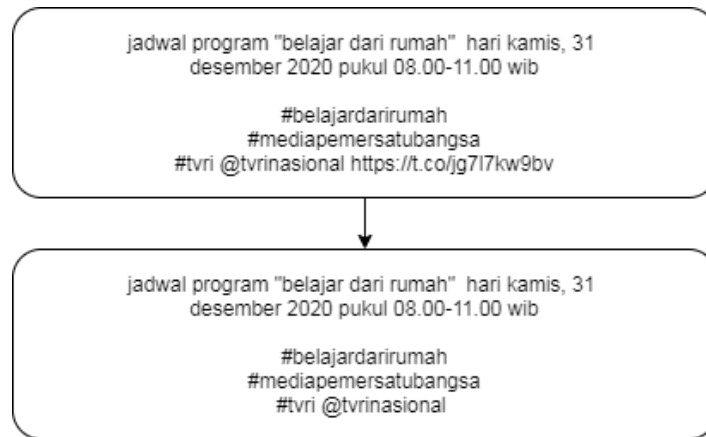
**Gambar 4.1** Proses *case folding*

##### b. Cleansing

Pada proses *cleansing* dilakukan penyaringan dan pembuangan teks. Proses *cleansing* terdiri atas beberapa tahapan antara lain: menghapus URL, *mention (@mention)*, *hashtag (#hashtag)*, selain huruf (a-z) dan spasi berlebih.

##### 1) Menghapus URL

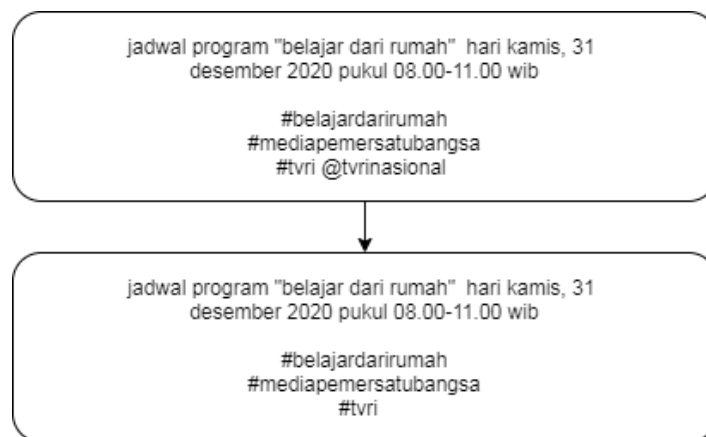
Pada Gambar 4.2 proses penghapusan URL akan menghapus semua teks yang diawali dengan 'http'.



**Gambar 4.2 Proses menghapus URL**

2) Menghapus *mention* (@*mention*)

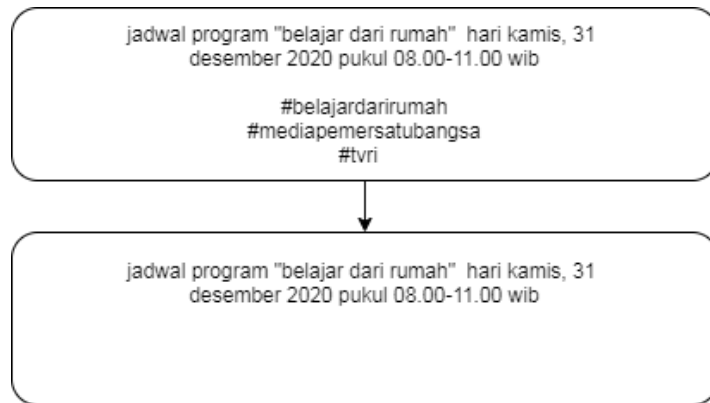
Pada Gambar 4.3 proses penghapusan *mention* (@*mention*) akan menghapus semua teks yang diawali dengan '@'.



**Gambar 4.3 Proses menghapus *mention***

3) Menghapus *hashtag* (#*hashtag*)

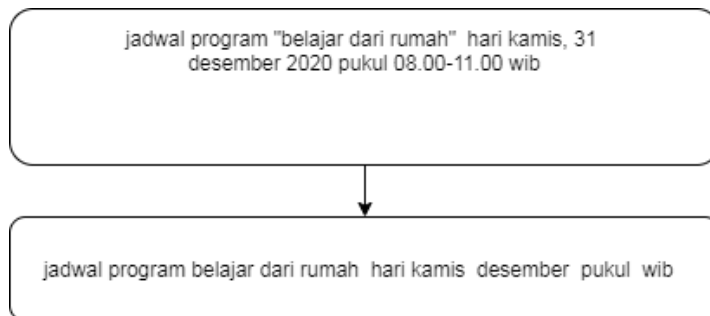
Pada Gambar 4.4 proses penghapusan tagar atau *hashtag* (#*hashtag*) akan menghapus semua teks yang diawali dengan '#'.



**Gambar 4.4 Proses menghapus *hashtag***

4) Menghapus selain huruf

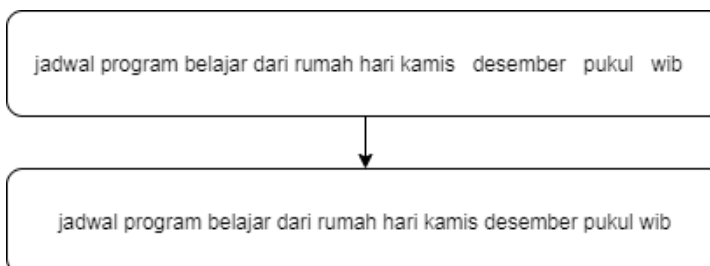
Pada Gambar 4.5 proses penghapusan selain huruf akan menghapus karakter selain huruf a-z pada teks, misalnya: '31 desember 2020' menjadi ' desember '.



**Gambar 4.5 Proses menghapus selain huruf**

5) Menghapus spasi berlebih

Pada Gambar 4.6 proses penghapusan spasi atau (*whitespace*) berlebih akan menghapus *whitespace* berlebih pada teks, misalnya: 'pukul wib' menjadi 'pukul wib'.

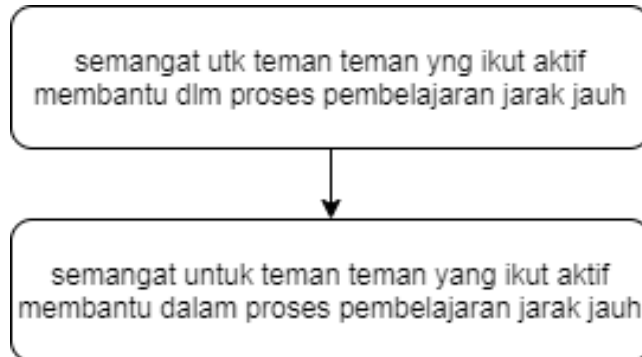


**Gambar 4.6 Proses menghapus spasi berlebih**

c. Merubah slang word

Pada Gambar 4.7 proses merubah *slang word* akan merubah setiap kata gaul, kata singkatan atau kata tidak baku

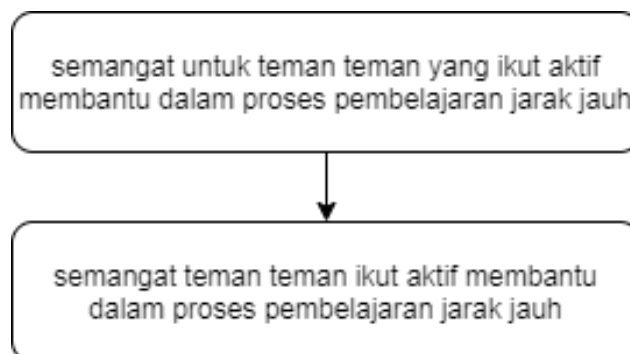
ke bentuk bakunya, misalnya: ‘utk’ menjadi ‘untuk’, ‘yng’ menjadi ‘yang’ dan seterusnya. Proses perubahan tersebut melibatkan kamus *slang word* yang terdapat dalam basis data (*database*).



**Gambar 4.7 Proses merubah *slang word***

d. Menghapus stop word

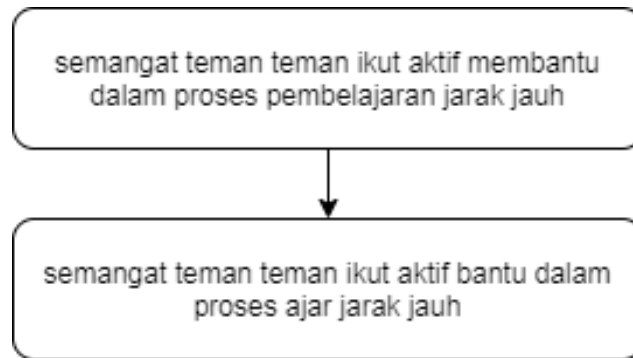
Pada Gambar 4.8 proses menghapus *stop word* akan menghapus setiap kata yang kurang memiliki makna namun sering dijumpai dalam sebuah teks, misalnya kata: ‘untuk’ ‘yang’, dan seterusnya. Proses penghapusan tersebut melibatkan kamus *stop word* yang terdapat dalam basis data (*database*).



**Gambar 4.8 Proses menghapus *stop word***

e. Stemming

Pada Gambar 4.9 proses *stemming* akan mengubah kata berimbuhan menjadi kata dasar dengan melibatkan pustaka Sastrawi, misalnya kata: ‘membantu’ menjadi ‘bantu’, ‘pembelajaran’ menjadi ‘ajar’, dan seterusnya.



**Gambar 4.9 Proses stemming**

Dalam penelitian ini, tahap *preprocessing* akan mengolah dan menghasilkan 4.314 data (*tweet*) yang lebih terstruktur atau disebut dengan *clean text*, yang kemudian akan disimpan ke dalam basis data (*database*) untuk tahap selanjutnya (*labeling*).

#### 4. 2. 3. Tahap *labeling*

Tahap *labeling* merupakan tahapan yang hanya dapat dilakukan setelah tersedianya satu atau lebih data *clean text* pada basis data (*database*) hasil dari tahapan *preprocessing*. Berdasarkan sub sub bab (3. 2. 3), tahap *labeling* utama dilakukan dengan cara menggunakan kamus sentimen, berdasarkan sub bab (2. 6), tahap *labeling* dengan kamus sentimen dilakukan dengan proses perhitungan skor sentimen dan pemberian kelas sentimen. Proses perhitungan skor sentimen pada sebuah *tweet* dapat dilihat pada Tabel 4.1 berikut:

**Tabel 4.1 Proses perhitungan skor**

<i>Dataset (clean text)</i>	<b>Kata Positif</b>	<b>Kata Negatif</b>
semangat teman teman ikut aktif bantu dalam proses ajar jarak jauh	semangat (1) teman (2) ikut (1) aktif (1) bantu (1) proses (1) ajar (1)	semangat (1) jauh (1)
<b>Jumlah</b>	8	2

Berdasarkan Tabel 4.1, menggunakan persamaan (2.1) maka dapat diperoleh perhitungan skor untuk *tweet* ‘semangat teman teman ikut aktif bantu dalam proses ajar jarak jauh’ yaitu sebagai berikut:

$$\begin{aligned}\text{skor} &= (\text{jumlah kata positif}) - (\text{jumlah kata negatif}) \\ \text{skor} &= 8 - 2 \\ \text{skor} &= 6\end{aligned}$$

Setelah diketahui nilai skor, proses selanjutnya adalah pemberian kelas sentimen berdasarkan aturan yang telah dijelaskan pada sub sub bab (2. 6. 2) sebagai berikut:

```
if skor > 0:
    kelas = 'positif'
elif skor < 0:
    kelas = 'negatif'
else:
    continue
```

Maka dapat disimpulkan bahwa *tweet* ‘semangat teman teman ikut aktif bantu dalam proses ajar jarak jauh’ akan mendapatkan kelas positif, karena nilai skor > 0.

Dalam penelitian ini, diperoleh 3.954 *tweet* berlabel. Sebanyak 3.733 *tweet* berlabel tersebut diperoleh menggunakan cara *labeling* kamus sentimen secara otomatis, sementara 221 *tweet* lainnya diperoleh dengan *labeling* secara manual dengan melibatkan subjektifitas beberapa orang (empat orang), dengan mengabaikan *tweet* yang dinilai sebagai netral.

#### 4. 2. 4. Tahap pembagian data

Tahap pembagian data merupakan tahapan yang dilakukan setelah tersedianya satu atau lebih data (*tweet*) berlabel pada basis data (*database*) hasil dari tahapan *labeling*. Berdasarkan sub sub bab (3. 2. 4), *tweet* berlabel akan dibagi menjadi dua (2) bagian, antara lain data uji dan data latih. Pembagian data dilakukan menggunakan rasio yang telah ditentukan yaitu 1:9 (data uji : data latih) atau 10% data uji, 90% data latih.

Dalam penelitian ini, tahap pembagian data dilakukan terhadap 3.954 *tweet* berlabel. Dengan rasio data 1:9, maka dapat diketahui jumlah data uji yang diperoleh sebanyak 395 *tweet* berlabel, sementara jumlah data latih yang diperoleh sebanyak 3.559 *tweet* berlabel. *Tweet* berlabel tersebut kemudian akan disimpan ke dalam basis data (*database*) untuk tahap selanjutnya (ekstraksi fitur).

#### 4. 2. 5. Tahap ekstraksi fitur *CountVectorizer*

Tahap ekstraksi fitur menggunakan *CountVectorizer* (*modeling*) merupakan tahapan yang dilakukan setelah *tweet* melalui proses *preprocessing*, *labeling*, dan pembagian data. Tahapan ini bertujuan untuk memperoleh *model* latih atau

pengetahuan melalui data latih yang ada. Berdasarkan sub sub bab (3. 2. 5), tahap ini terdapat lima (5) proses utama antara lain: seleksi data latih, pembuatan list kata, pencarian fitur kata, pembuatan vektor kosong dan membuat representasi vektor. Berikut penjabaran dari tahap *modeling*:

a. Seleksi data latih

Seleksi data latih dilakukan setelah data melalui proses *preprocessing*, *labeling*, dan pembagian data. Menggunakan teknik sampling kuota (*quota sampling*) seperti yang telah dijelaskan pada sub bab (2. 8) dan sub-sub bab (3. 2. 5), tahap pertama dalam *modeling* adalah pengambilan sampel dari populasi data latih untuk dijadikan sebagai pengetahuan berdasarkan kriteria tertentu, kriteria yang dimaksud adalah dengan menyamakan jumlah antara data berlabel positif dengan data berlabel negatif. Sampel data latih yang diambil dapat dilihat pada Tabel 4.2 berikut:

**Tabel 4.2 Sampel data latih**

<i>Tweet</i> ( $T_{latih-i}$ )	<i>Clean Text</i>	<i>Sentiment Type</i>
latih-1	ajar efektif kelas pintar semangat gratis	positif
latih-2	pagi tetap semangat ajar aktivitas rabu pintar ayo simak jadwal acara	positif
latih-3	pelita bangsa tengah pandemi covid bangkit semangat wujud merdeka ajar	positif
latih-4	susah sulit kerja tugas bingung tanya tanya kelas pintar akibat covid	negatif
latih-5	covid ajar jarak jauh sulit didik tugas banyak	negatif
latih-6	pagi susah kerja lama lama ajar jarak jauh penuh drama	negatif

Sampel data latih pada Tabel 4.2 terdapat tiga (3) kolom yaitu: *Tweet* ( $T_{latih-i}$ ) yang berarti urutan *Tweet* latih ke-i, *Clean Text* yang berarti teks *tweet* yang telah terstruktur setelah melalui proses *preprocessing*, dan *Sentiment Type* yang berarti jenis kategori (*label*) *tweet* yang diperoleh setelah melalui proses *labeling*.



b. Pembuatan list kata

Dari sampel data latih pada Tabel 4.2 kemudian akan dipisahkan menjadi satuan kata. Pemisahan menjadi kata dilakukan berdasarkan spasi (*whitespace*), kemudian hasilnya akan ditampilkan dalam sebuah wadah *list*. Hasil proses ini dapat dilihat pada Tabel 4.3 berikut:

**Tabel 4.3 List kata**

List kata
['ajar', 'efektif', 'kelas', 'pintar', 'semangat', 'gratis', 'pagi', 'tetap', 'semangat', 'ajar', 'aktivitas', 'rabu', 'pintar', 'ayo', 'simak', 'jadwal', 'acara', 'pelita', 'bangsa', 'tengah', 'pandemi', 'covid', 'bangkit', 'semangat', 'wujud', 'merdeka', 'ajar', 'susah', 'sulit', 'kerja', 'tugas', 'bingung', 'tanya', 'tanya', 'kelas', 'pintar', 'akibat', 'covid', 'covid', 'ajar', 'jarak', 'jauh', 'sulit', 'didik', 'tugas', 'banyak', 'pagi', 'susah', 'kerja', 'lama', 'lama', 'ajar', 'jarak', 'jauh', 'penuh', 'drama']

List kata pada Tabel 4.3 merupakan hasil dari proses pemisahan kata dari kolom *clean text* pada *tweet* berdasarkan pada Tabel 4.2 Sampel data latih.

c. Pencarian fitur kata

Pencarian fitur kata dilakukan dengan cara melakukan pencarian dan pendataan setiap kata unik (*unique*) dengan membuang kata duplikat yang terdapat dalam *list* kata. Sehingga diperoleh *list* fitur kata seperti pada Tabel 4.4 berikut:

**Tabel 4.4 Fitur kata**

Fitur kata
['ajar', 'efektif', 'kelas', 'pintar', 'semangat', 'gratis', 'pagi', 'tetap', 'aktivitas', 'rabu', 'ayo', 'simak', 'jadwal', 'acara', 'pelita', 'bangsa', 'tengah', 'pandemi', 'covid', 'bangkit', 'wujud', 'merdeka', 'susah', 'sulit', 'kerja', 'tugas', 'bingung', 'tanya', 'akibat', 'jarak', 'jauh', 'didik', 'banyak', 'lama', 'penuh', 'drama']

Fitur kata pada Tabel 4.4 merupakan hasil dari proses pencarian fitur berdasarkan pada Tabel 4.3 *list* kata.

d. Membuat vektor kosong latihan

Membuat vektor kosong dimaksudkan untuk menyiapkan wadah berbentuk vektor dengan isian nilai awal yaitu angka (*integer*) nol (0). Wadah vektor tersebut dibentuk dengan panjang berdasarkan jumlah fitur kata. Berdasarkan Tabel 4.2 Sampel data latih dan Tabel 4.4 Fitur kata, maka vektor kosong yang dihasilkan seperti pada Tabel 4.5 berikut:

### Tabel 4.5 Vektor kosong latihan

<i>Tweet</i> <i>(T<sub>latih-i</sub>)</i>	Vektor Kosong
latih-1	[0, 0]
latih-2	[0, 0]
latih-3	[0, 0]
latih-4	[0, 0]
latih-5	[0, 0]
latih-6	[0, 0]

e. Membuat vektor kata latih

Proses membuat vektor kata dilakukan dengan pengubahan nilai pada vektor kosong berdasarkan frekuensi kemunculan fitur pada tiap kata dalam *tweet*. Nilai representasi vektor diperoleh berdasarkan jumlah kemunculan fitur dalam *tweet*. Berdasarkan pada Tabel 4.4 Fitur kata dan Tabel 4.5 Vektor kosong, maka vektor kata yang dihasilkan seperti pada Tabel 4.6 berikut:

<i>Tweet</i> <i>(T<sub>latih-i</sub>)</i>	Representasi vektor
latih-1	[1, 1, 1, 1, 1, 1, 0]
latih-2	[1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
latih-3	[1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
latih-4	[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 2, 1, 0, 0, 0, 0]
latih-5	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0]
latih-6	[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 2, 1, 1]

Sebelum beralih ke tahap selanjutnya (tahap klasifikasi), representasi vektor dan atribut lain yang dihasilkan pada tahap *modeling* akan disimpan ke dalam sebuah wadah berbentuk file dengan format JSON (.json). File JSON tersebut digunakan untuk menampung *model* latih seperti pada Tabel 4.7 berikut:

<i>key</i>	<i>value</i>
<i>teks_list</i>	[ 'ajar efektif kelas pintar semangat gratis', 'pagi tetap semangat ajar aktivitas rabu pintar ayo simak jadwal acara', 'pelita bangsa tengah pandemi covid bangkit semangat wujud merdeka ajar', 'susah sulit kerja tugas bingung tanya tanya kelas pintar akibat covid', 'covid



representasi vektor dari Tabel 4.6 Representasi vektor secara berurutan berdasarkan *Tweet* latih ke-*i* pada Tabel 4.2.

Dalam t ini, dari sejumlah 3.559 data latih, hanya sebanyak 1500 data latih berlabel yang dipilih menggunakan teknik *quota sampling*, dengan ciri: 750 *tweet* latih positif dan 750 *tweet* latih negatif. *Tweet* tersebut kemudian akan dijadikan sebuah pengetahuan (*model* latih) dan disimpan ke dalam sebuah file dengan format JSON (.json) untuk tahap selanjutnya (klasifikasi).

#### 4. 2. 6. Tahap klasifikasi *K-Nearest Neighbor*

Tahap klasifikasi menggunakan *K-Nearest Neighbor* (KNN) merupakan tahapan yang dilakukan setelah tahap ekstraksi fitur (*modeling*). Tahapan ini bertujuan untuk memprediksikan *label* untuk setiap data uji berdasarkan *model* latih yang dihasilkan pada tahap *modeling*. Berikut penjabaran dari tahap klasifikasi KNN:

##### a. Persiapan data

Proses persiapan data merupakan proses pemilihan nilai *K* tetangga terdekat dan *model* latih. *Model* latih yang terpilih selanjutnya akan dijadikan sebagai landasan dalam melakukan klasifikasi untuk data uji yang tersedia. Pada tahap klasifikasi dalam penulisan ini, *model* latih yang dipilih merupakan *model* latih hasil dari sub-sub bab (4. 2. 1), sementara untuk data uji akan digunakan adalah sampel data uji seperti pada Tabel 4.8 berikut:

**Tabel 4.8 Sampel data uji**

<i>Tweet</i> ( $T_{uji-i}$ )	<i>Clean Text</i>	<i>Sentiment</i> <i>Type</i>
uji-1	semangat ikut kelas pintar ajar jarak jauh tengah pandemi	positif
uji-2	susah sulit ajar jarak jauh pandemi covid covid tetap semangat	negatif

##### b. Membuat representasi vektor uji

Pembuatan representasi vektor uji menggunakan pengetahuan yang bersumber dari *model* latih. Pembuatan representasi vektor uji ini terdiri atas dua (2) proses antara lain: membuat vektor kosong dan membuat vektor kata.

##### 1) Membuat vektor kosong uji

Dalam proses ini akan dibuat wadah vektor kosong seperti yang dijelaskan pada sub-sub bab (4. 2. 1) bagian d, vektor kosong dibuat berdasarkan pada *feature\_list* (*model* latih) dan jumlah data uji. Berdasarkan jumlah

fitur pada *model* latih dan Tabel 4.8 Sampel data uji, maka vektor kosong akan dibuat seperti pada Tabel 4.9 berikut:

**Tabel 4.9 Vektor kosong uji**

[illegible]

2) Membuat vektor kata uji

Dalam proses ini akan dibuat representasi vektor untuk setiap data seperti yang dijelaskan pada sub-sub bab (4. 2. 1) bagian e, vektor kata dibuat berdasarkan jumlah kemunculan *feature\_list* (*model* latih) dengan tiap kata dalam *tweet* data uji. Maka vektor kata akan dibuat seperti pada Tabel 4.10 berikut:

**Tabel 4.10 Representasi vektor uji**

<i>Tweet</i> ( $T_{uji-i}$ )	Representasi vektor
uji-1	[1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0]
uji-2	[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0]

c. Menghitung jarak antar data

Perhitungan jarak dilakukan menggunakan data vektor uji (Tabel 4.10 Representasi vektor uji) dan data *vector\_list* pada *model* latih. Berdasarkan pada sub bab (2. 9), proses perhitungan jarak melibatkan *euclidean distance* dengan persamaan (2. 2). Berikut contoh penerapan *euclidean distance* ( $d$ ) dalam menghitung jarak pada vektor uji-1 ( $x$ ) dengan vektor latih-1 ( $y$ ):

$$d_{(uji\ 1, latih1)} = \sqrt{\begin{matrix} (1-1)^2 + (0-1)^2 + (1-1)^2 + (1-1)^2 \\ + (1-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 \\ + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 \\ + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (0-0)^2 + (1-0)^2 + (1-0)^2 + (0-0)^2 \\ + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

$$d_{(uji\ 1, latih1)} = \sqrt{\begin{matrix} 0 + 1 + 0 + 0 + 0 + 1 + 0 + 0 + 0 \\ + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 1 \\ + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ + 0 + 0 + 1 + 1 + 0 + 0 + 0 + 0 + 0 \end{matrix}}$$

$$d_{(uji\ 1, latih1)} = \sqrt{6}$$

$$d_{(uji\ 1, latih1)} = 2.449489742783178$$

Berdasarkan contoh sebelumnya, maka hasil perhitungan untuk setiap jarak antara vektor uji dengan vektor latih adalah seperti dalam Tabel 4.11 berikut:

**Tabel 4.11 Hasil jarak euclidean distance**

<i>Tweet</i> ( $T_{uji-i}$ )	<i>Tweet</i> ( $T_{latih-i}$ )	<i>Euclidean Distance</i> ( $d_{(uji-i, latih-i)}$ )
uji-1	latih-1	$d_{(1,1)} = 2.449489742783178$
	latih-2	$d_{(1,2)} = 3.605551275463989$
	latih-3	$d_{(1,3)} = 3.1622776601683795$
	latih-4	$d_{(1,4)} = 4.123105625617661$
	latih-5	$d_{(1,5)} = 3.1622776601683795$
	latih-6	$d_{(1,6)} = 3.7416573867739413$
uji-2	latih-1	$d_{(2,1)} = 3.7416573867739413$

<i>Tweet</i> ( $T_{uji-i}$ )	<i>Tweet</i> ( $T_{latih-i}$ )	<i>Euclidean Distance</i> ( $d_{(uji-i, latihan-i)}$ )
	latih-2	$d_{(2,2)} = 4.123105625617661$
	latih-3	$d_{(2,3)} = 3.4641016151377544$
	latih-4	$d_{(2,4)} = 4.123105625617661$
	latih-5	$d_{(2,5)} = 2.8284271247461903$
	latih-6	$d_{(2,6)} = 4.0$

d. Mencari tetangga terdekat

Proses pencarian tetangga terdekat melibatkan nilai K. Nilai K dalam *K-nearest neighbor* (KNN) merupakan jumlah data ketetanggaan terdekat yang hendak diperoleh. Dalam penelitian ini nilai K yang dapat digunakan telah ditentukan, yaitu: K=3, K=5, K=7, K=9, dan K=11. Sementara pada penulisan ini nilai K yang dipilih adalah K=3.

Proses pencarian tetangga terdekat dilakukan dengan melalui dua (2) proses, antara lain: mengurutkan nilai jarak dan mengambil K data tetangga terdekat.

1) Mengurutkan nilai jarak

Dalam proses ini, nilai dari Tabel 4.11 Hasil jarak *euclidean distance* akan diurutkan secara urut menaik (*ascending*) berdasarkan jarak. Sehingga hasil urutan dapat diperoleh seperti pada Tabel 4.12 berikut:

**Tabel 4.12 Pengurutan jarak tetangga**

Urutan	Jarak ( $d_{(uji-i, latihan-i)}$ )	<i>Tweet</i> ( $T_{uji-i, latihan-i}$ )
1	2.449489742783178	uji-1, latihan-1
2	3.1622776601683795	uji-1, latihan-3
3	3.1622776601683795	uji-1, latihan-5



Urutan	Jarak ( $d_{(uji-i, \text{latih-i})}$ )	<i>Tweet</i> ( $T_{uji-i, \text{latih-i}}$ )
4	3.605551275463989	uji-1, latih-2
5	3.7416573867739413	uji-1, latih-6
6	4.123105625617661	uji-1, latih-4
1	2.8284271247461903	uji-2, latih-5
2	3.4641016151377544	uji-2, latih-3
3	3.7416573867739413	uji-2, latih-1
4	4.0	uji-2, latih-6
5	4.123105625617661	uji-2, latih-2
6	4.123105625617661	uji-2, latih-4

2) Mengambil K data tetangga terdekat

Setelah melalui proses pengurutan, data dari Tabel 4.12 Pengurutan jarak tetangga akan diambil sebanyak K buah data, dengan nilai K yang telah ditentukan dan dipilih. Dengan nilai  $K=3$ , sehingga diperoleh hasil tetangga terdekat seperti Tabel 4.13 berikut:

**Tabel 4.13 Data K tetangga terdekat**

Urutan	<i>Tweet</i> ( $T_{uji-i, \text{latih-i}}$ )	Jarak ( $d_{(uji-i, \text{latih-i})}$ )
1	uji-1, latih-1	2.449489742783178
2	uji-1, latih-3	3.1622776601683795
3	uji-1, latih-5	3.1622776601683795

Urutan	<i>Tweet</i> ( $T_{uji-i, latih-i}$ )	Jarak ( $d_{(uji-i, latih-i)}$ )
1	uji-2, latih-5	2.8284271247461903
2	uji-2, latih-3	3.4641016151377544
3	uji-2, latih-1	3.7416573867739413

e. Menghitung nilai probabilitas

Nilai probabilitas diperoleh dengan cara melihat probabilitas *label* yang muncul pada data K tetangga terdekat. Nilai probabilitas yang dicari adalah nilai probabilitas *tweet* uji akan berlabel positif dan nilai probabilitas *tweet* uji akan berlabel negatif. Hal tersebut dapat diketahui melalui *label\_list* pada *model* latih dan Tabel 4.13 Data K tetangga terdekat, bahwa nilai probabilitas yang dihasilkan adalah seperti pada Tabel 4.14 berikut:

**Tabel 4.14 Nilai probabilitas data uji**

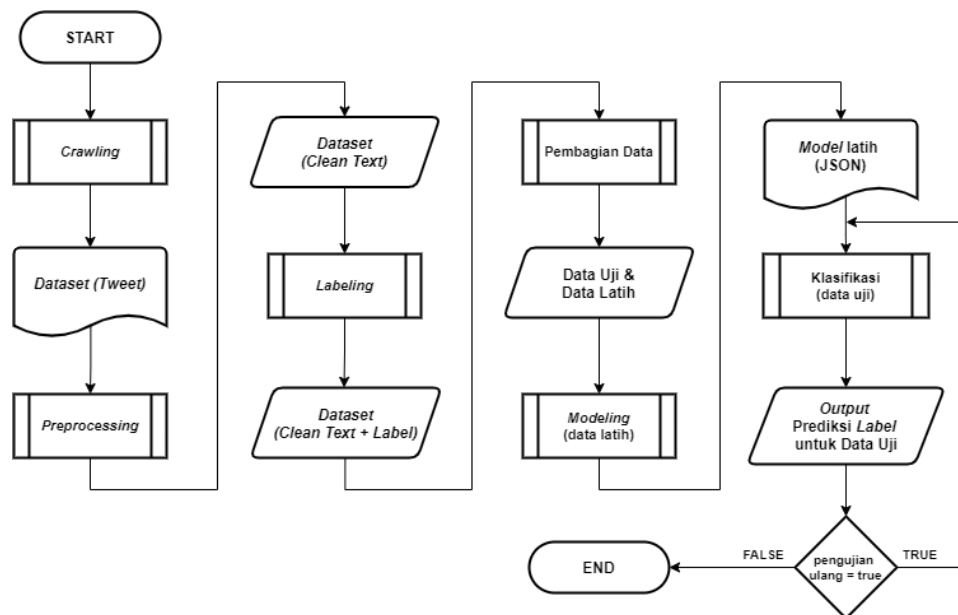
<i>Tweet</i> ( $T_{uji-i}$ )	<i>Tweet</i> ( $T_{latih-i}$ )	<i>Sentiment</i> <i>Type</i>	Probabilitas positif	Probabilitas negatif
uji-1	latih-1	positif	1	0
	latih-3	positif	1	0
	latih-5	negatif	0	1
Jumlah			2 (0.667)	1 (0.333)
uji-2	latih-5	negatif	0	1
	latih-3	positif	1	0
	latih-1	positif	1	0
Jumlah			2 (0.667)	1 (0.333)

Berdasarkan Tabel 4.14 Nilai probabilitas data uji, dapat diketahui dengan  $K=3$ , pada pengujian dengan *tweet* uji-1 dan *tweet* uji-2 keduanya akan sama-sama diprediksikan berlabel positif dengan nilai probabilitas yang sama yaitu 0.667 atau 66.7%.

#### 4. 3. Flowchart Tahapan Metode

*Flowchart* merupakan suatu bagan atau simbol yang menggambarkan alur kerja atau urutan proses pada suatu program. Berikut adalah penjabaran *flowchart* dalam penelitian ini:

##### 4. 3. 1. Flowchart keseluruhan proses sistem

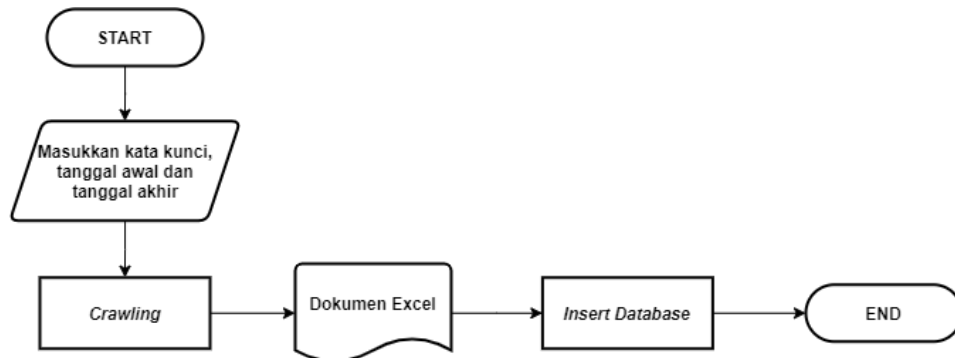


**Gambar 4.10 Flowchart keseluruhan proses sistem**

Pada Gambar 4.10 *Flowchart* keseluruhan proses sistem, menjelaskan proses-proses yang dilalui sistem yang dibuat. Dimulai dari tahap *crawling* sehingga menghasilkan *dataset* berupa *tweet*, kemudian tahap *preprocessing* untuk menghasilkan kolom *clean text*, selanjutnya tahap *labeling* untuk memberikan *label* berupa positif atau negatif, hasil tahap *labeling* akan menghasilkan kolom *label*, kemudian pembagian data untuk membagi *dataset* berlabel antara data uji dan data latih berdasarkan rasio 1:9, lalu tahap *modeling* menggunakan data latih berlabel untuk menghasilkan sebuah *model* latih, yang kemudian akan diuji beberapa kali dengan nilai variasi nilai  $K$  yang berbeda pada tahap klasifikasi. Hasil dari tahapan tersebut akan berupa *label* sentimen prediksi untuk setiap data uji.

#### 4. 3. 2. Flowchart proses crawling

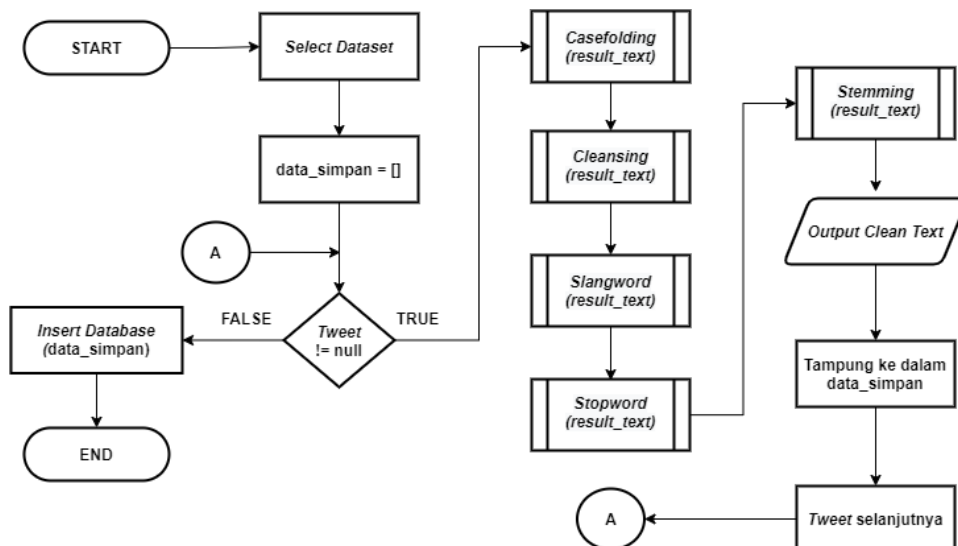
Pada *flowchart* ini, menjelaskan proses pengumpulan data atau *crawling* data tweet dimulai dari memasukkan kata kunci dan tanggal awal dan tanggal akhir, cara lainnya adalah dengan fitur import file *excel*, kemudian dilakukan pencarian *tweet* berdasarkan parameter menggunakan pustaka Tweepy. Hasil pengumpulan data akan disimpan dalam bentuk file Excel (.xlsx) sebelum kemudian dimasukkan ke dalam *database*. *Flowchart* proses *crawling* dapat dilihat pada Gambar 4.11 berikut:



Gambar 4.11 Flowchart proses crawling

#### 4. 3. 3. Flowchart proses preprocessing

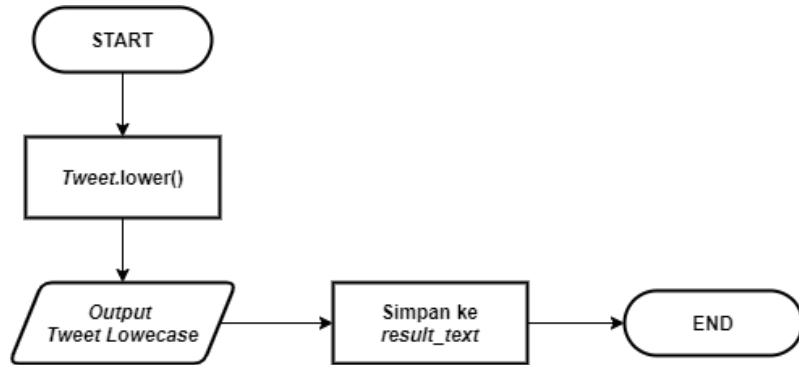
Pada *flowchart* ini, menjelaskan proses perubahan data perubahan data teks *tweet* menjadi terstruktur atau setara, guna mendukung proses klasifikasi agar berjalan dengan baik. Hasil proses ini berupa teks bersih, yang kemudian akan disimpan ke dalam *database* untuk proses selanjutnya. *Flowchart* proses *preprocessing* dapat dilihat pada Gambar 4.12 berikut:



Gambar 4.12 Flowchart proses preprocessing

a. Flowchart proses case folding

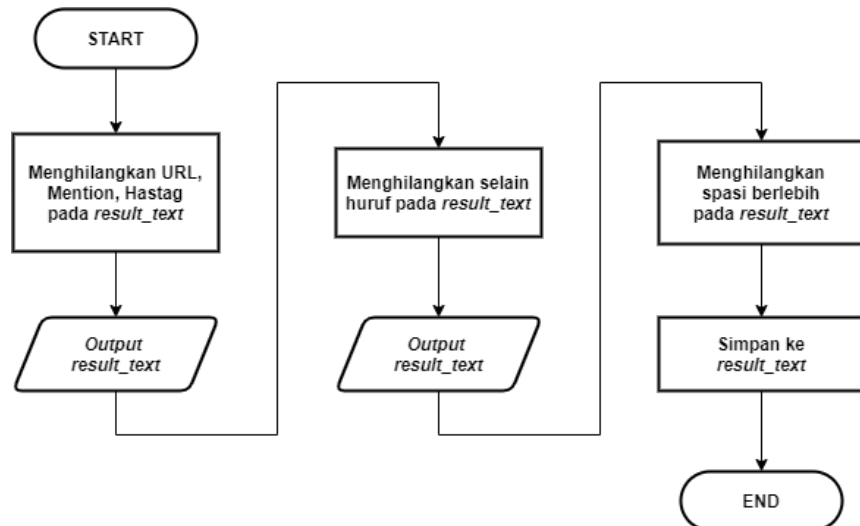
Pada *flowchart* ini dilakukan proses pengubahan teks *tweet* menjadi huruf kecil secara keseluruhan. Hasil dari proses ini ditampung ke dalam sebuah variabel bernama *result\_text* untuk proses selanjutnya. *Flowchart* proses *case folding* dapat dilihat pada Gambar 4.13 berikut:



Gambar 4.13 Flowchart proses case folding

b. Flowchart proses cleansing

Pada *flowchart* ini dilakukan proses penyaringan dan pembuangan atribut pada teks *result\_text*, diantaranya penghapusan atau penghilangan URL, *mention*, *hashtag*, selain huruf dan spasi atau baris berlebih. Hasil dari proses ini ditampung ke dalam sebuah variabel bernama *result\_text* untuk proses selanjutnya. *Flowchart* proses *cleansing* dapat dilihat pada Gambar 4.14 berikut:

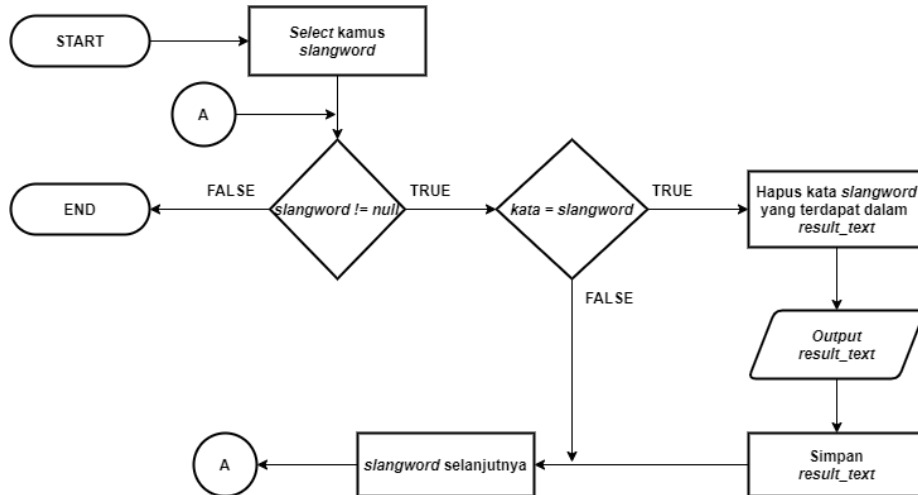


Gambar 4.14 Flowchart proses cleansing

c. Flowchart proses slang word

Pada *flowchart* ini dilakukan perubahan atribut pada teks *result\_text* yang mengandung kata *slang* dari kamus *slang*

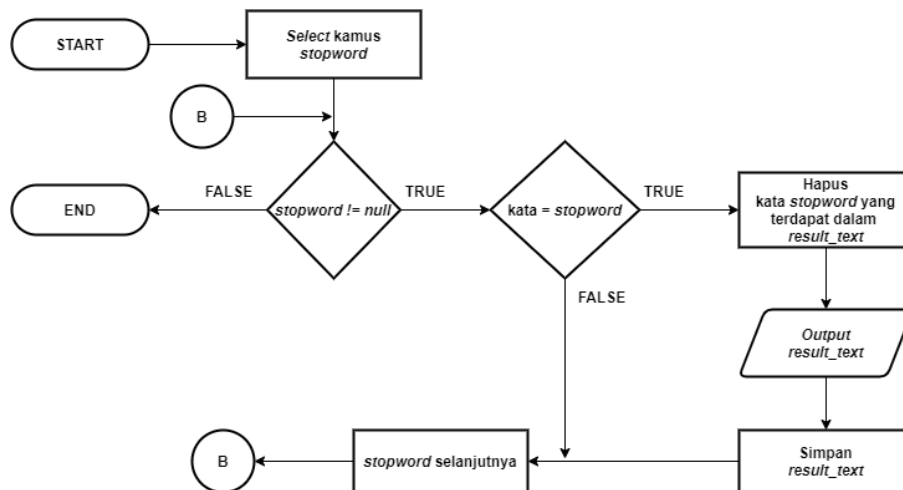
*word* ke bentuk kata asli atau kata bakunya. Hasil dari proses ini ditampung ke dalam sebuah variabel bernama *result\_text* untuk proses selanjutnya. *Flowchart* proses *slang word* dapat dilihat pada Gambar 4.15 berikut:



**Gambar 4.15** *Flowchart* proses *slang word*

d. *Flowchart* proses *stop word*

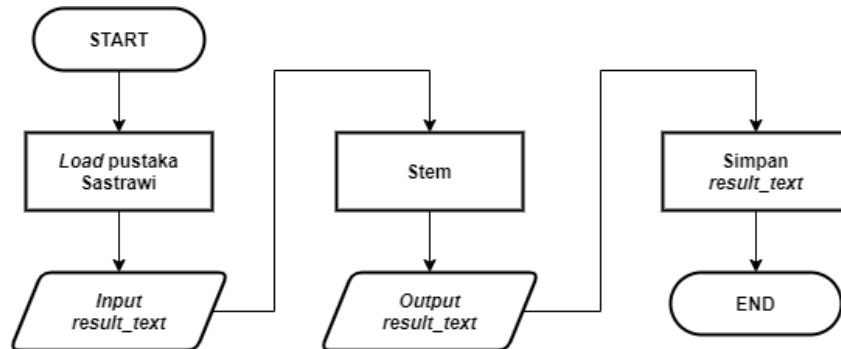
Pada *flowchart* ini dilakukan proses penghapusan atribut pada teks *result\_text* yang mengandung *stop word* dari kamus *stop word* karena dianggap kurang memiliki makna untuk proses klasifikasi. Hasil dari proses ini ditampung ke dalam sebuah variabel bernama *result\_text* untuk proses selanjutnya. *Flowchart* proses *stop word* dapat dilihat pada Gambar 4.16 berikut:



**Gambar 4.16** *Flowchart* proses *stop word*

e. Flowchart proses stemming

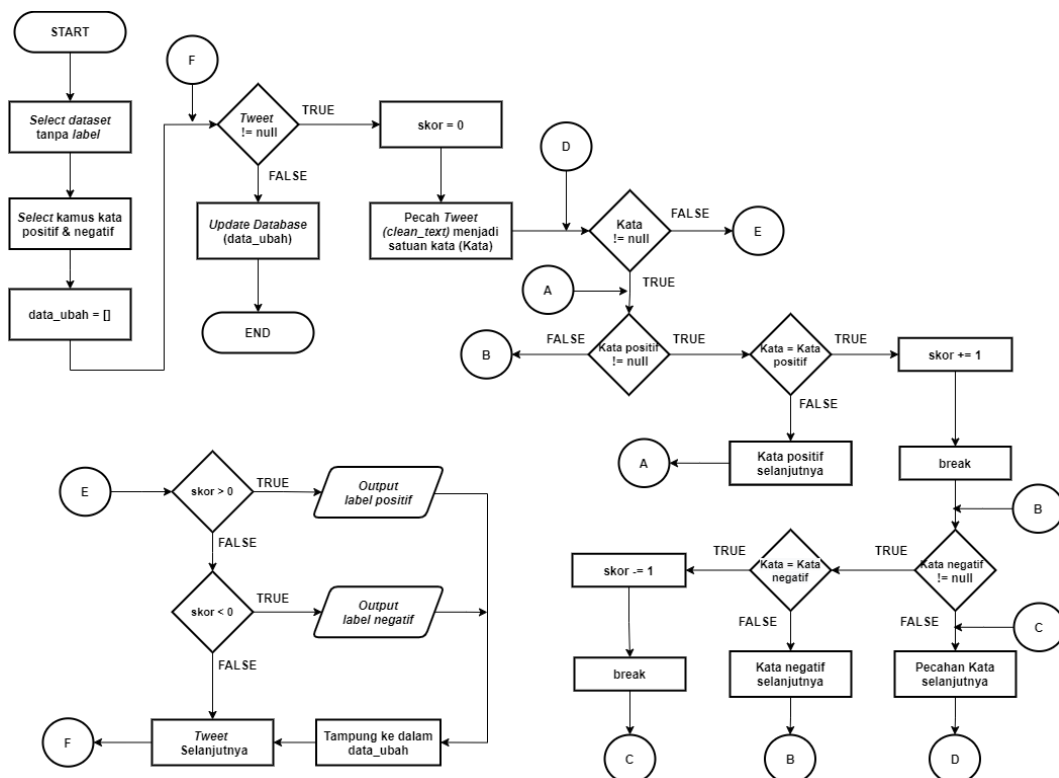
Pada *flowchart* ini dilakukan perubahan atribut pada teks *result\_text* yang mengandung kata berimbuhan ke bentuk kata asalnya, menggunakan pustaka Sastrawi. *Flowchart* proses *stemming* dapat dilihat pada Gambar 4.17 berikut:



Gambar 4.17 Flowchart proses stemming

4. 3. 4. Flowchart proses labeling

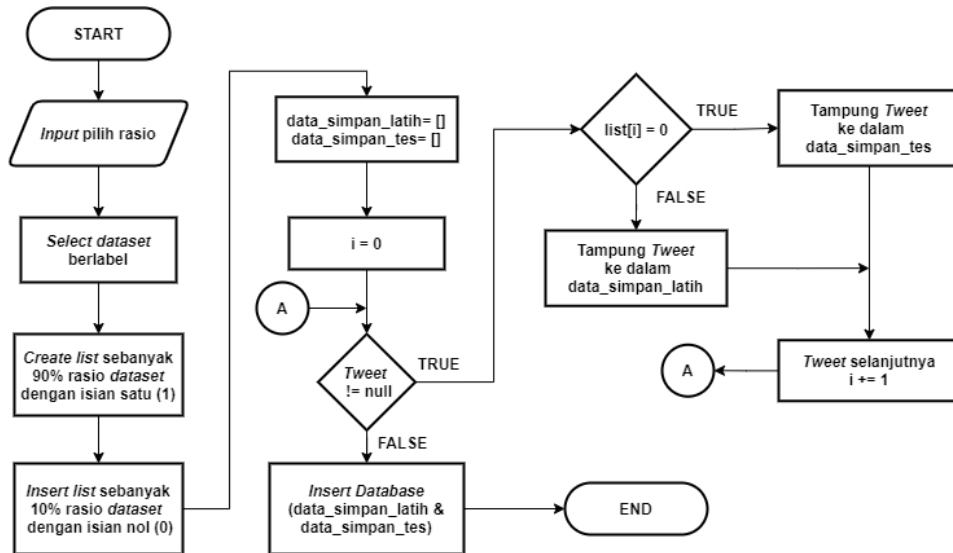
Pada *flowchart* ini, menjelaskan proses *labeling* yang dilakukan dengan cara pendekatan kamus sentimen, dimana diawali dengan perhitungan skor menggunakan kamus positif dan negatif, lalu penentuan kelas atau *label* berdasarkan nilai skor. *Flowchart* proses *labeling* dapat dilihat pada Gambar 4.18 berikut:



Gambar 4.18 Flowchart proses labeling

#### 4. 3. 5. Flowchart proses pembagian data

Pada *flowchart* ini, menjelaskan proses pembagian data ke dua (2) buah bagian, yaitu: data uji dan data latih, menggunakan rasio 1:9 (data uji : data latih). *Flowchart* proses pembagian data dapat dilihat pada Gambar 4.19 berikut:

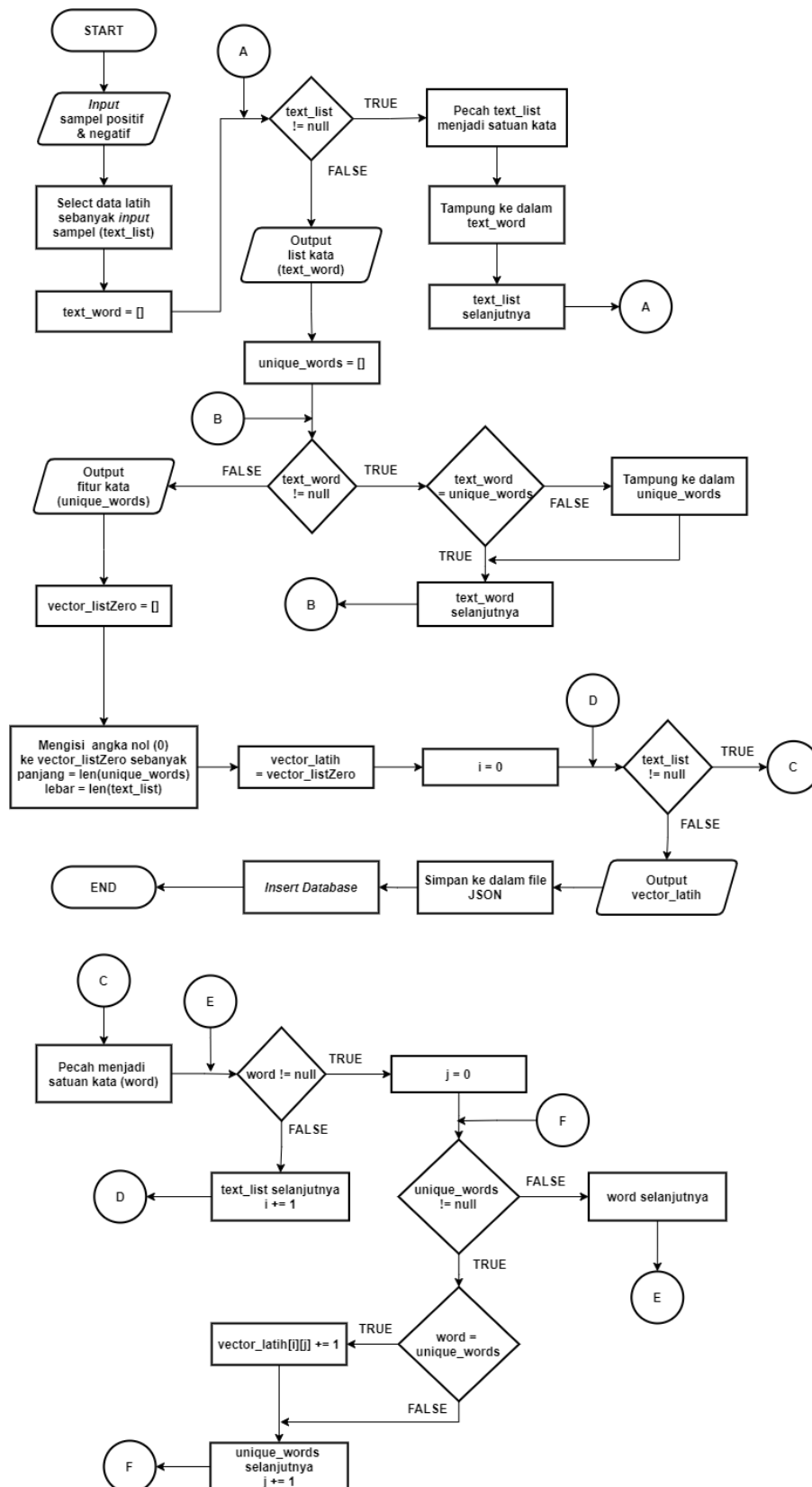


Gambar 4.19 Flowchart proses pembagian data

#### 4. 3. 6. Flowchart proses modeling

Pada *flowchart* ini, menjelaskan proses ekstraksi fitur dengan *CountVectorizer*. Dimulai dari proses seleksi data latih, pembuatan list kata, pencarian fitur kata, pembuatan vektor kosong, dan pembuatan vektor kata. Sehingga dapat diperoleh hasil berupa data *model* latih, yang kemudian disimpan ke dalam file JSON. *Flowchart* proses *modeling* dapat dilihat pada Gambar 4.20 berikut:

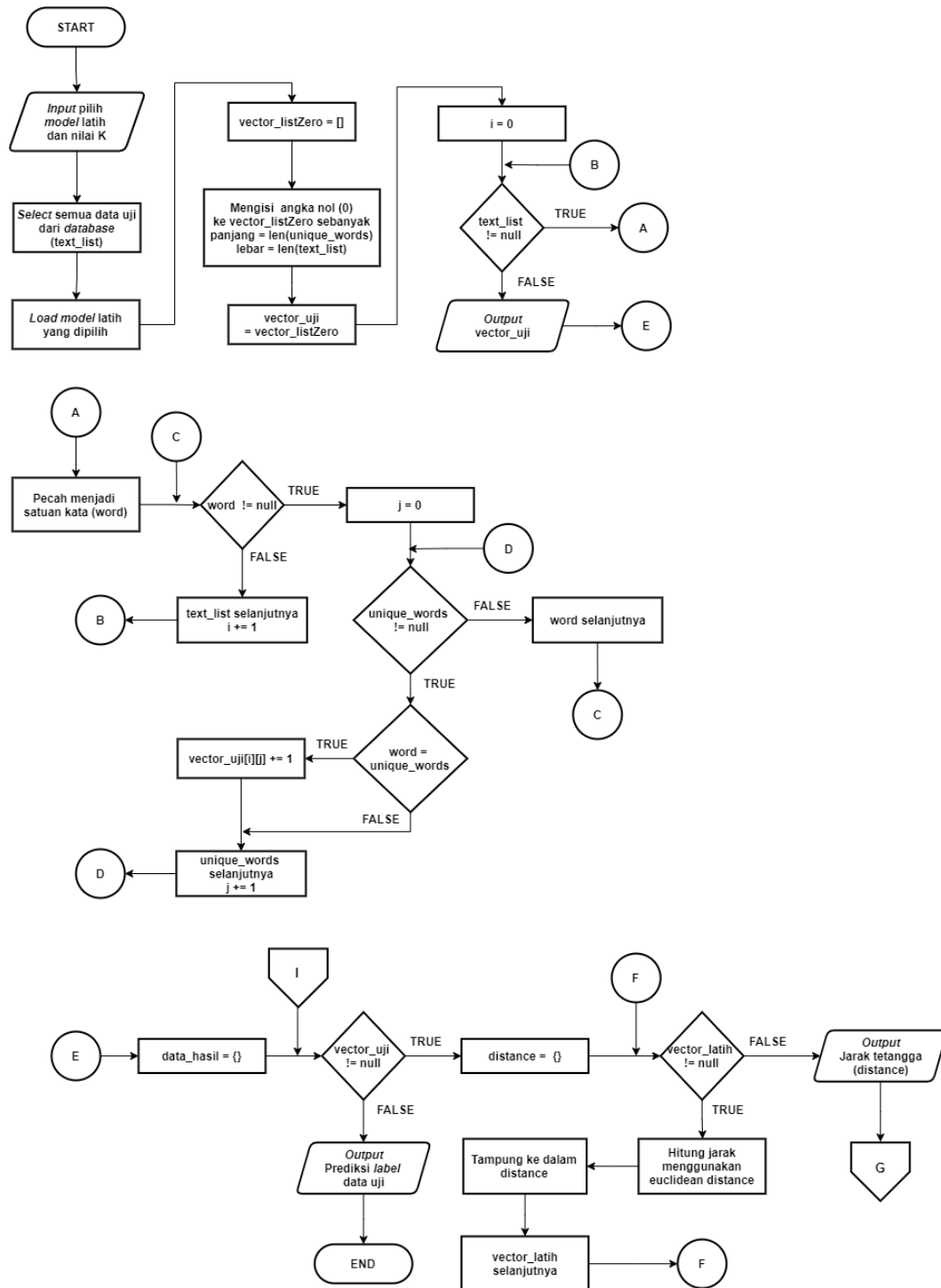


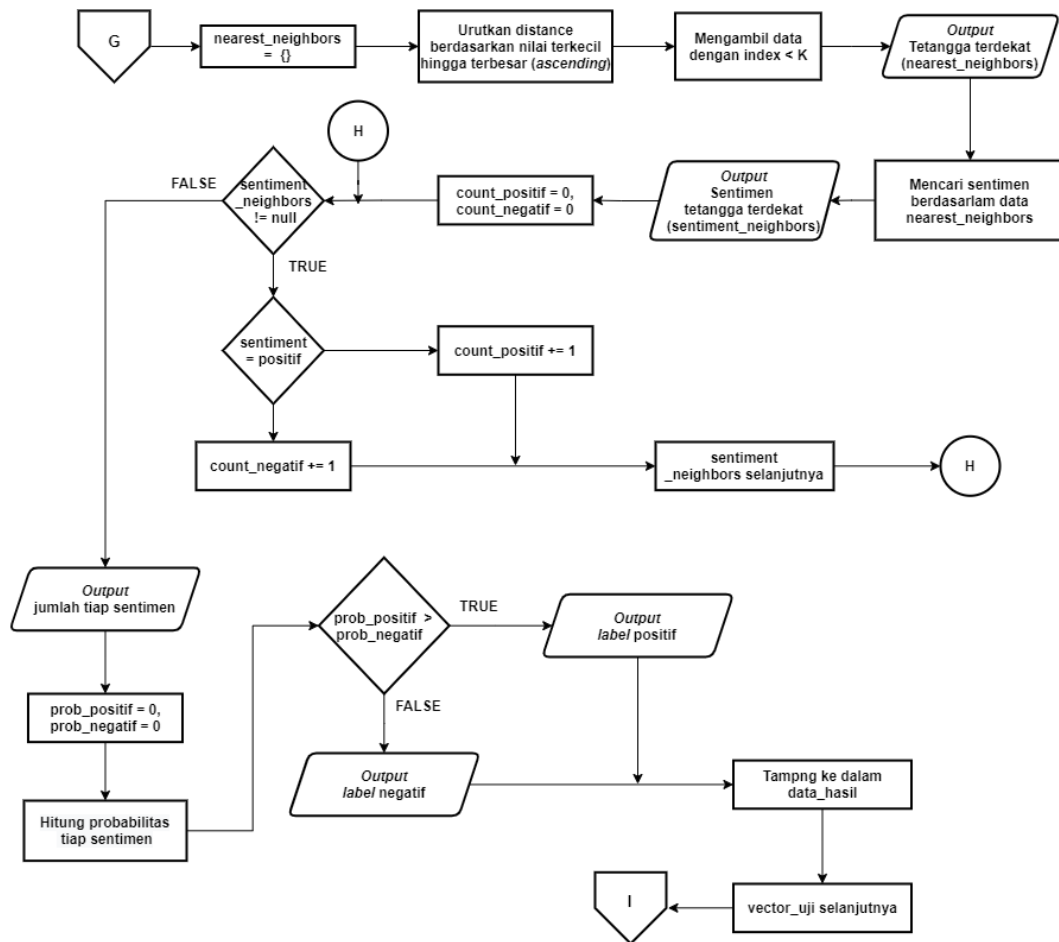


Gambar 4.20 Flowchart proses modeling

#### 4.3.7. *Flowchart* proses klasifikasi

Pada *flowchart* ini, menjelaskan proses klasifikasi dengan *K-Nearest Neighbor*. Dimulai dari proses pembuatan vektor uji, menghitung jarak antar data, mencari tetangga terdekat, dan menghitung nilai probabilitas. *Flowchart* proses klasifikasi dapat dilihat pada Gambar 4.21 berikut:

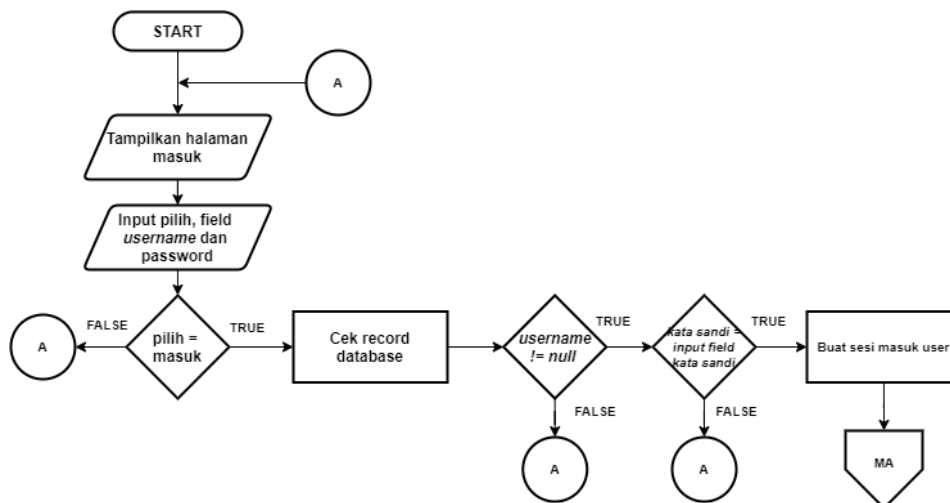




**Gambar 4.21 Flowchart proses klasifikasi**

#### 4. 3. 8. Flowchart menu masuk aplikasi

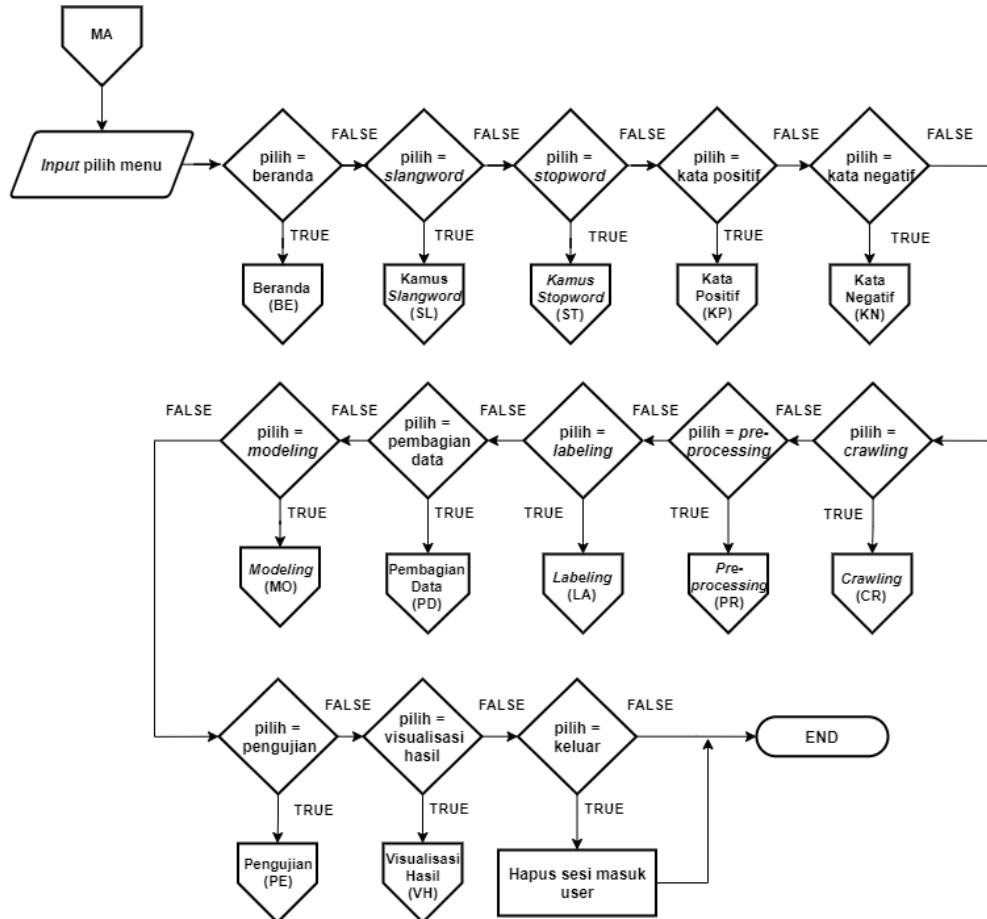
Pada *flowchart* ini, menjelaskan mengenai proses menu masuk dalam aplikasi yang dibuat. *Flowchart* menu masuk dapat dilihat pada Gambar 4.22 berikut:



**Gambar 14.22 Flowchart menu masuk aplikasi**

#### 4. 3. 9. *Flowchart* menu aplikasi

Pada *flowchart* ini, menjelaskan mengenai menu-menu dalam aplikasi yang dibuat. *Flowchart* keseluruhan menu aplikasi dapat dilihat pada Gambar 4.23 berikut:



**Gambar 4.23** *Flowchart* menu aplikasi

#### 4. 3. 10. *Flowchart* menu beranda

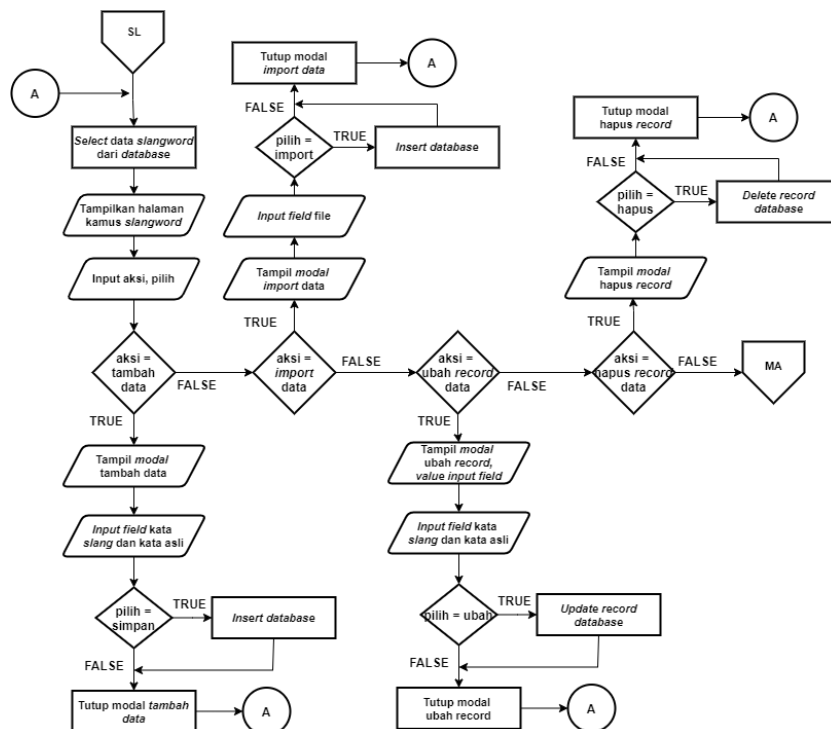
Pada *flowchart* ini, menjelaskan mengenai proses menu beranda dalam aplikasi yang dibuat. *Flowchart* menu beranda dapat dilihat pada Gambar 4.24 berikut:



Gambar 4.24 Flowchart menu beranda

#### 4. 3. 11. Flowchart menu slang word

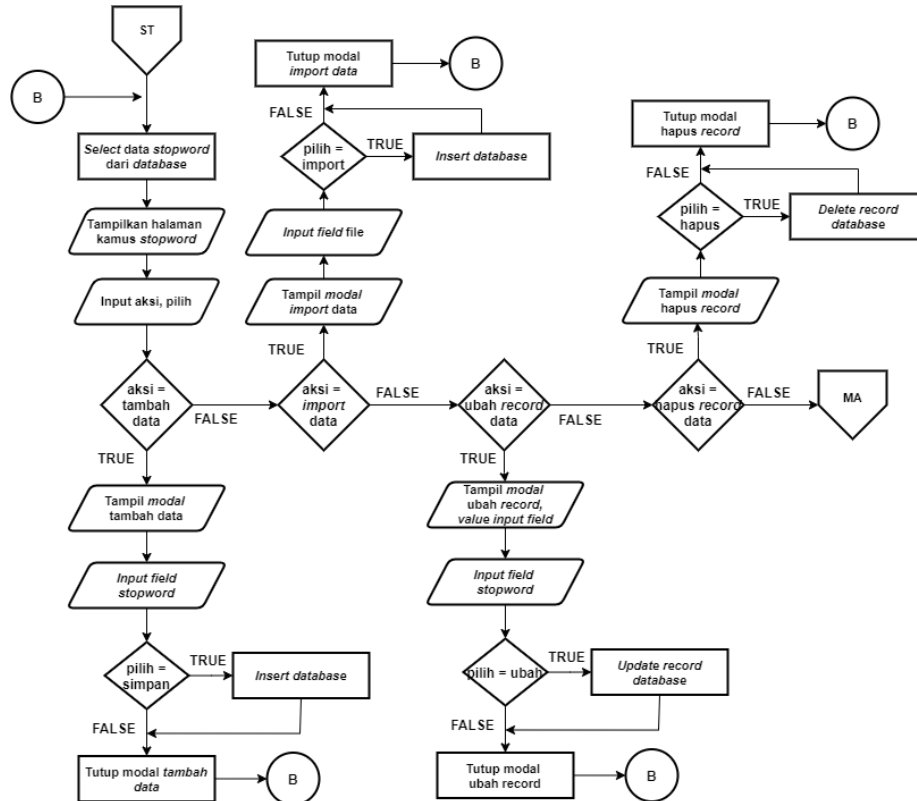
Pada *flowchart* ini, menjelaskan mengenai proses menu *slang word* dalam aplikasi yang dibuat. *Flowchart* menu *slang word* dapat dilihat pada Gambar 4.25 berikut:



Gambar 4.25 Flowchart menu slang word

#### 4. 3. 12. Flowchart menu stop word

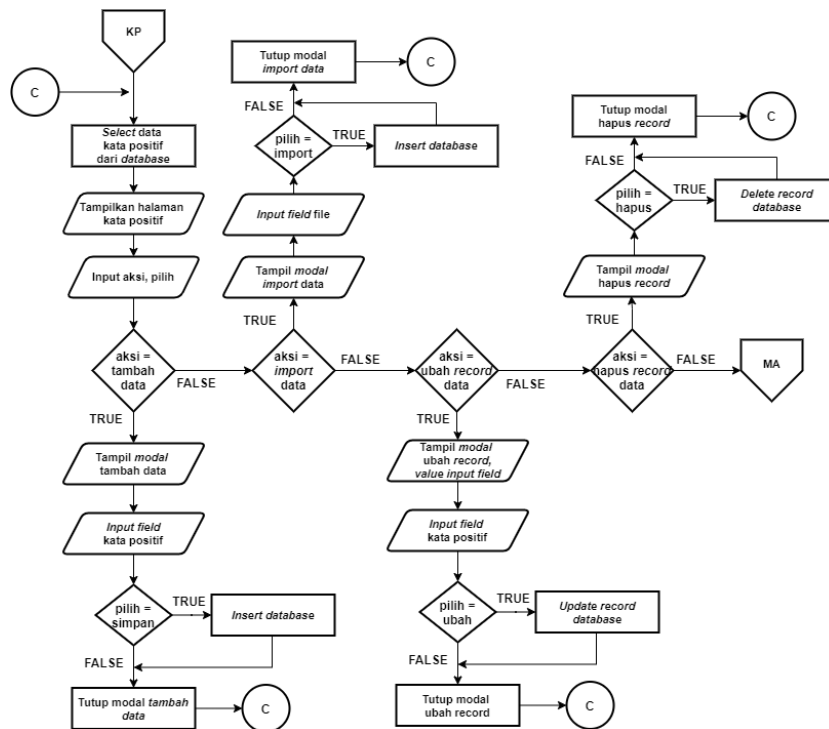
Pada *flowchart* ini, menjelaskan mengenai proses menu *stop word* dalam aplikasi yang dibuat. *Flowchart* menu *stop word* dapat dilihat pada Gambar 4.26 berikut:



**Gambar 4.26 Flowchart menu stop word**

#### 4. 3. 13. Flowchart menu kata positif

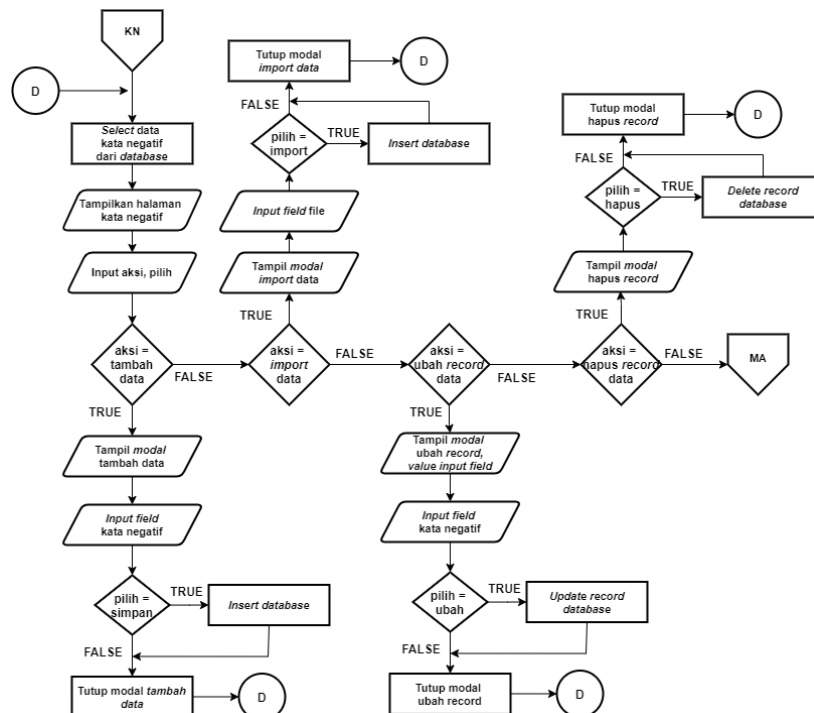
Pada *flowchart* ini, menjelaskan mengenai proses menu kata positif dalam aplikasi yang dibuat. *Flowchart* menu kata positif dapat dilihat pada Gambar 4.27 berikut:



**Gambar 4.27 Flowchart menu kata positif**

#### 4. 3. 14. Flowchart menu kata negatif

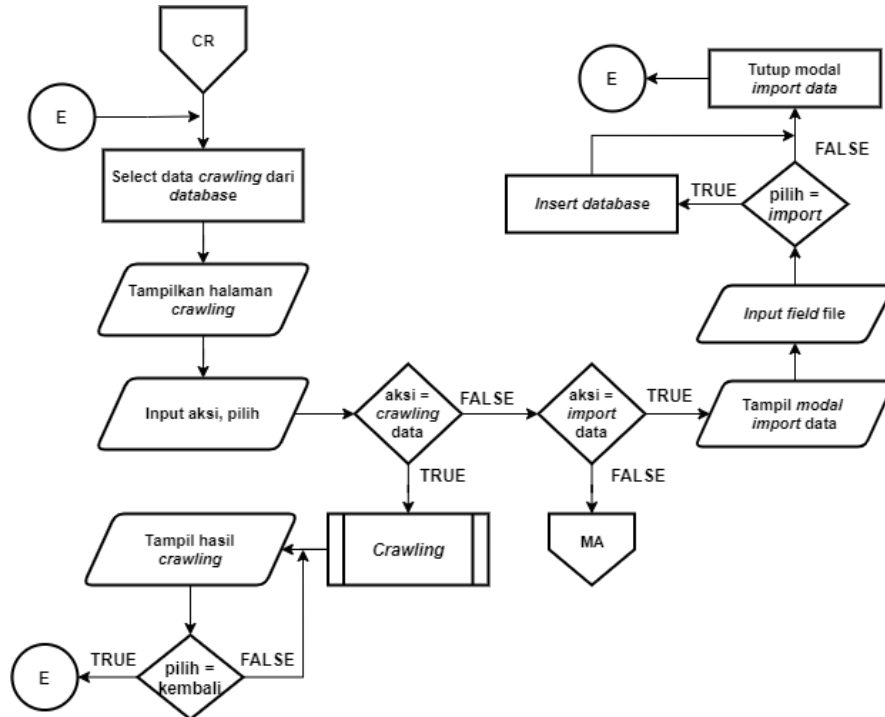
Pada *flowchart* ini, menjelaskan mengenai proses menu kata negatif dalam aplikasi yang dibuat. *Flowchart* menu kata negatif dapat dilihat pada Gambar 4.28 berikut:



**Gambar 4.28 Flowchart menu kata negatif**

#### 4. 3. 15. Flowchart menu crawling

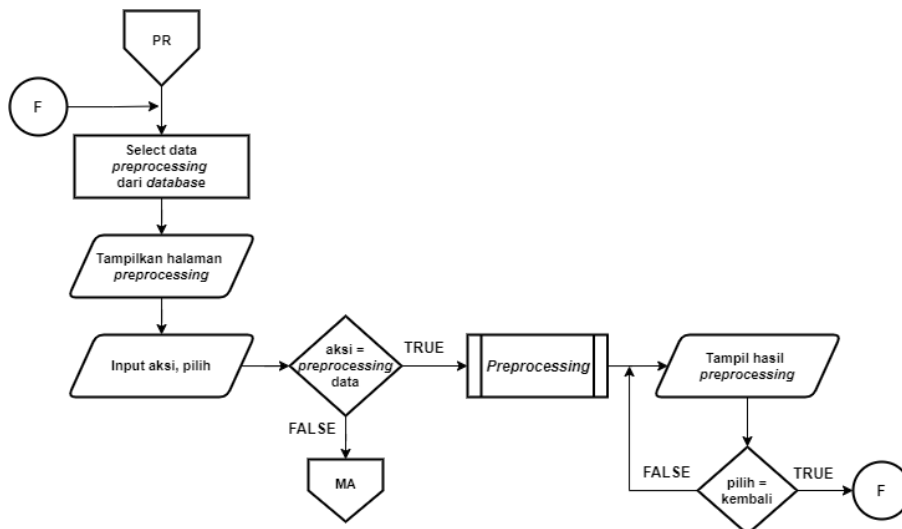
Pada *flowchart* ini, menjelaskan mengenai proses menu *crawling* dalam aplikasi yang dibuat. *Flowchart* menu *crawling* dapat dilihat pada Gambar 4.29 berikut:



Gambar 4.29 Flowchart menu crawling

#### 4. 3. 16. Flowchart menu preprocessing

Pada *flowchart* ini, menjelaskan mengenai proses menu *preprocessing* dalam aplikasi yang dibuat. *Flowchart* menu *preprocessing* dapat dilihat pada Gambar 4.30 berikut:

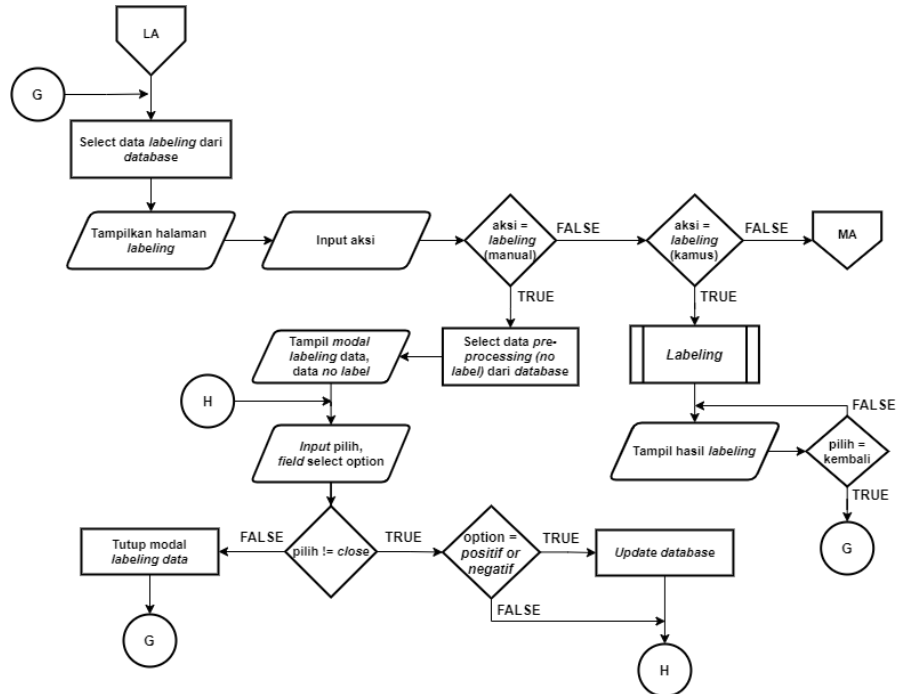


Gambar 4.30 Flowchart menu preprocessing



#### 4. 3. 17. Flowchart menu labeling

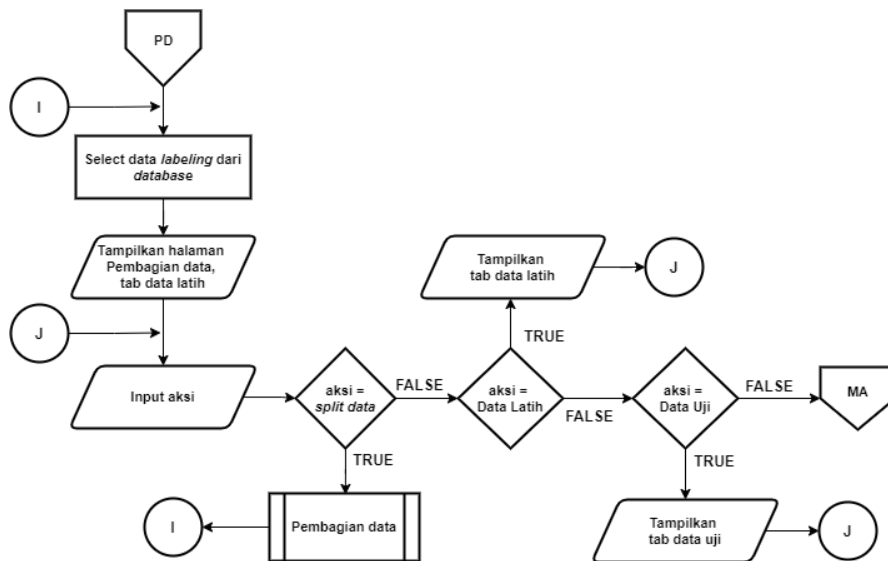
Pada flowchart ini, menjelaskan mengenai proses menu labeling dalam aplikasi yang dibuat. Flowchart menu labeling dapat dilihat pada Gambar 4.31 berikut:



Gambar 4.31 Flowchart menu labeling

#### 4. 3. 18. Flowchart menu pembagian data

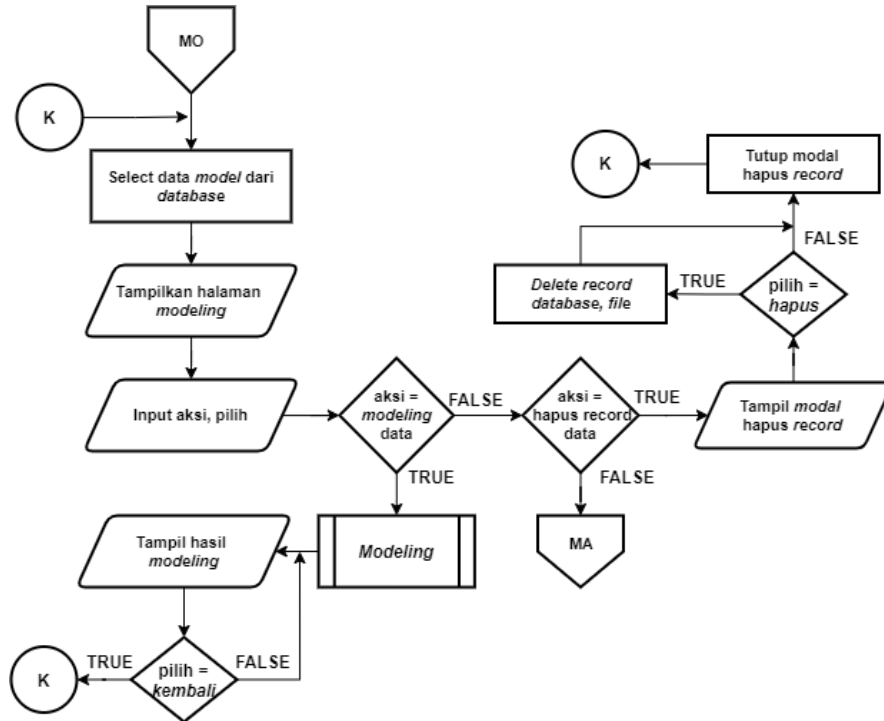
Pada flowchart ini, menjelaskan mengenai proses menu pembagian data dalam aplikasi yang dibuat. Flowchart menu pembagian data dapat dilihat pada Gambar 4.32 berikut:



Gambar 4.32 Flowchart menu pembagian data

#### 4. 3. 19. Flowchart menu *modeling*

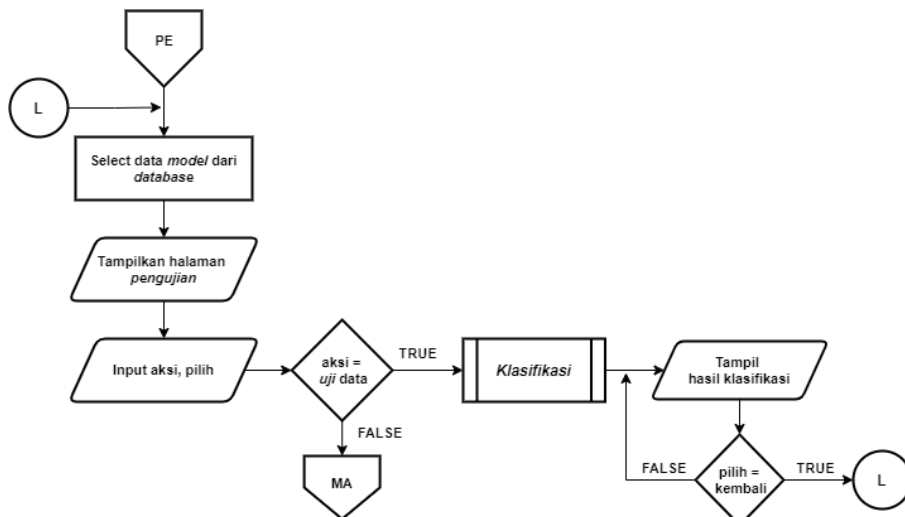
Pada *flowchart* ini, menjelaskan mengenai proses menu *modeling* dalam aplikasi yang dibuat. *Flowchart* menu *modeling* dapat dilihat pada Gambar 4.33 berikut:



**Gambar 4.33 Flowchart menu *modeling***

#### 4. 3. 20. Flowchart menu pengujian

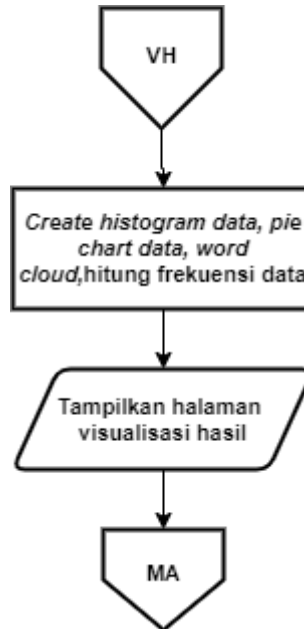
Pada *flowchart* ini, menjelaskan mengenai proses menu pengujian dalam aplikasi yang dibuat. *Flowchart* menu pengujian dapat dilihat pada Gambar 4.34 berikut:



**Gambar 4.34 Flowchart menu pengujian**

#### 4. 3. 21. *Flowchart* menu visualisasi hasil

Pada *flowchart* ini, menjelaskan mengenai proses menu visualisasi hasil dalam aplikasi yang dibuat. *Flowchart* menu visualisasi hasil dapat dilihat pada Gambar 4.35 berikut:



**Gambar 4.35 *Flowchart* menu visualisasi hasil**

#### 4. 4. **Algoritme Tahapan Metode**

Algoritme merupakan suatu urutan atau tahapan proses yang dijabarkan dalam bentuk tulisan, algoritme juga merupakan representasi pengaplikasian dari suatu *flowchart*. Berikut adalah penjabaran algoritme berdasarkan pada *flowchart* yang telah dibuat sebelumnya.

##### 4. 4. 1. Algoritme keseluruhan proses sistem

Pada algoritme ini dijelaskan tentang proses keseluruhan sistem yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme keseluruhan proses sistem dapat dilihat pada Algoritme 4.1 berikut:

**Algoritme 4.1 Algoritme keseluruhan proses sistem**

1	start
2	Lakukan proses Crawling
3	Baca Dataset Text (Tweet)
4	Lakukan proses Preprocessing
5	Baca Dataset Clean Text (Tweet)
6	Lakukan proses Labeling
7	Baca Dataset
8	Lakukan proses Pembagian Dataset
9	Baca Data Latih & Data Uji
10	Lakukan proses Modeling menggunakan data latih
11	Simpan Model ke file JSON
12	Lakukan proses Klasifikasi untuk data uji

```

13 Output prediksi label data uji
14 if (pengujian ulang == true)
15     Kembali ke nomor 12
16 endif
17 end

```

#### 4. 4. 2. Algoritme proses *crawling*

Pada algoritme ini dijelaskan tentang proses pengumpulan data dengan cara *crawling* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *crawling* dapat dilihat pada Algoritme 4.2 berikut:

##### **Algoritme 4.2 Algoritme proses *crawling***

```

1 start
2 Input field kata kunci
3 Input field tanggal awal
4 Input field tanggal akhir
5 Crawling menggunakan pustaka Tweepy
6 Baca dokumen Excel (dataframe)
7 Simpan dataframe ke dalam database
8 end

```

#### 4. 4. 3. Algoritme proses *preprocessing*

Pada algoritme ini dijelaskan tentang proses *preprocessing* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *preprocessing* dapat dilihat pada Algoritme 4.3 berikut:

##### **Algoritme 4.3 Algoritme proses *preprocessing***

```

1 start
2 Select Dateset (Tweet)
3 data_simpan = []
4 if (Tweet != null)
5     Lakukan sub proses Casefolding
6     (result_text)
7     Lakukan sub proses Cleansing (result_text)
8     Lakukan sub proses Slangword (result_text)
9     Lakukan sub proses Stopword (result_text)
10    Lakukan sub proses Stemming (result_text)
11    Simpan result_text ke dalam list
12    data_simpan
13    Tweet selanjutnya
14    Kembali ke nomor 4
15 else
16    Simpan data_simpan ke dalam database
17 endif
18 end

```

##### a. Algoritme proses case folding

Pada algoritme ini dijelaskan tentang proses *case folding* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *case folding* dapat dilihat pada Algoritme 4.4 berikut:

##### **Algoritme 4.4 Algoritme proses *case folding***

```

1 start

```

```

2  Baca Datasets (Tweet)
3  Ubah Tweet ke huruf kecil menggunakan pre-
   made function lower()
4  Output result_text
5  Simpan result_text
6  end

```

b. Algoritme proses cleansing

Pada algoritme ini dijelaskan tentang proses *cleansing* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *cleansing* dapat dilihat pada Algoritme 4.5 berikut:

**Algoritme 4.5 Algoritme proses cleansing**

```

1  start
2  Select result_text
3  Hapus URL
4  Hapus Mention
5  Hapus Hastag
6  Output result_text
7  Hapus selain huruf
8  Output result_text
9  Hapus spasi berlebih
10 Output result_text
11 end

```

c. Algoritme proses slang word

Pada algoritme ini dijelaskan tentang proses pengubahan *slang word* berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *slang word* dapat dilihat pada Algoritme 4.6 berikut:

**Algoritme 4.6 Algoritme proses slang word**

```

1  start
2  Select kamus slangword dari database
3  if (slangword != null)
4      if (teks == slangword)
5          Ubah result_text (slangword) dengan
           kata asli
6          Output result_text
7          Simpan result_text
8      else
9          slangword selanjutnya
10         Kembali ke nomor 3
11     endif
12 endif
13 end

```

d. Algoritme proses stop word

Pada algoritme ini dijelaskan tentang proses penghilangan *stop word* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *stop word* dapat dilihat pada Algoritme 4.7 berikut:

**Algoritme 4.7 Algoritme proses stop word**

```

1  start

```

```

2   Select kamus stopwords dari database
3   if (stopword != null)
4       if (result_text in stopwords)
5           Hapus result_text (stopword)
6           Output result_text
7           Simpan result_text
8       endif
9       stopwords selanjutnya
10      Kembali ke nomor 3
11  endif
12  end

```

e. Algoritme proses stemming

Pada algoritme ini dijelaskan tentang proses *stemming* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *stemming* dapat dilihat pada Algoritme 4.8 berikut:

**Algoritme 4.8 Algoritme proses stemming**

```

1   start
2   Load pustaka Sastrawi
3   Inisialisasi paket StemmerFactory
4   Input result_text sebagai parameter fungsi
   stem()
5   Jalankan fungsi stem() melalui paket
   StemmerFactory
6   Output result_text
7   Simpan result_text
8   end

```

4. 4. 4. Algoritme proses *labeling*

Pada algoritme ini dijelaskan tentang proses *labeling* kelas atau *labeling* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *labeling* dapat dilihat pada Algoritme 4.9 berikut:

**Algoritme 4.9 Algoritme proses labeling**

```

1   start
2   Select dataset tanpa label dari database
3   Select kamus sentimen positif (Kata_positif)
   dan negatif (Kata_negatif) dari database
4   data_ubah = []
5   if (Tweet != null)
6       skor = 0
7       Split Tweet menjadi satuan kata (Kata)
8       if (Kata != null)
9           if (Kata_positif != null)
10              if (Kata == Kata_positif)
11                  skor = skor + 1
12                  break (Langsung menuju ke nomor
17)
13              else
14                  Kata_positif selanjutnya
15                  Kembali ke nomor 9
16              endif
17          endif

```

```

18         if (Kata_negatif != null)
19             if (Kata == Kata_negatif)
20                 skor = skor - 1
21                 break (Langsung menuju ke nomor
26)
22             else
23                 Kata_negatif selanjutnya
24                 Kembali ke nomor 18
25             endif
26         endif
27         Kata selanjutnya
28         Kembali ke nomor 8
29     else
30         if (skor > 0)
31             Output label positif
32             Simpan ke dalam list data_ubah
33         else if (skor < 0)
34             Output label negatif
35             Simpan ke dalam list data_ubah
36         else
37             continue (Langsung menuju ke nomor
38)
39         endif
40         Tweet selanjutnya
41         Kembali ke nomor 5
42     endif
43     Update database berdasarkan list data_ubah
44 endif
45 en

```

#### 4. 4. 5. Algoritme proses pembagian data

Pada algoritme ini dijelaskan tentang proses pembagian data yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses pembagian data dapat dilihat pada Algoritme 4.10 berikut:

##### **Algoritme 4.10 Algoritme proses pembagian data**

```

1  start
2  Input pilih rasio pembagian data
3  Select dataset berlabel
4  Buat list dengan isian satu (1) sebanyak 90%
   dari total dataset berlabel
5  Buat list dengan isian nol (0) sebanyak 10%
   dari total dataset berlabel
6  data_simpan_latih = []
7  data_simpan_tes = []
8  i = 0
9  if (Tweet != null)
10     if(list[i] == 0)
11         Simpan Tweet dalam data_simpan_tes
12     else
13         Simpan Tweet le dalam data_simpan_latih
14     endif
15     list = list + 1
16     Kembali ke nomor 9

```

```

17 else
18     Simpan data_simpan_latih ke dalam database
19     Simpan data_simpan_tes ke dalam database
20 endif
21 end

```

#### 4. 4. 6. Algoritme proses *modeling*

Pada algoritme ini dijelaskan tentang proses pembuatan *model* latih atau *modeling* yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses *modeling* dapat dilihat pada Algoritme 4.11 berikut:

**Algoritme 4.11 Algoritme proses *modeling***

```

1  start
2  Input jumlah sampel positif dan negatif
3  Select data latih sebanyak jumlah input sampel
   (text_list)
4  text_word = []
5  if (text_list != null)
6      Split text_list menjadi satuan kata (kata)
7      Simpan kata ke dalam list text_word
8      text_list selanjutnya
9      Kembali ke nomor 5
10 else
11     Output text_word
12     unique_words = []
13     if (text_word != null)
14         if (text_word in unique_words)
15             text_word selanjutnya
16         else
17             Simpan text_word ke dalam list
               unique_words
18             text_word selanjutnya
19         endif
20         Kembali ke nomor 13
21     else
22         Output unique words
23         vector_listZero = []
24         Isi dengan angka nol (0) ke dalam list
               vector_listZero dengan panjang =
               len(unique_words dan lebar =
               len(text_list)
25         vector_latih = vector_listZero
26         i = 0
27         if (text_list[i] != null)
28             Split text_list[i] menjadi satuan
               kata (word)
29             if (word != null)
30                 j = 0
31                 if (unique_words[j] != null)
32                     if (word ==
                           unique_words[j])
33                         vector_latih[i][j] =
                           vector_latih[i][j] + 1
34                     endif
35                     j = j + 1

```



```

36             Kembali ke nomor 31
37         else
38             word selanjutnya
39             Kembali ke nomor 29
40         endif
41     else
42         i = i + 1
43         Kembali ke nomor 27
44     endif
45 else
46     Output vector_latih
47     Simpan isi vector_latih ke dalam
48     file JSON (.json) pada direktori
49     Simpan nama vector_latih ke dalam
50     database
51 endif
52 endif
53 endif

```

#### 4. 4. 7. Algoritme proses klasifikasi

Pada algoritme ini dijelaskan tentang proses klasifikasi yang dilakukan berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme proses klasifikasi dapat dilihat pada Algoritme 4.12 berikut:

**Algoritme 4.12 Algoritme proses klasifikasi**

```

1  start
2  Input pilih model latih dan nilai K
3  Select data uji dari database (text_list)
4  Load model latih yang dipilih dari direktori
5  vector_listZero = []
6  Isi dengan angka nol (0) ke dalam list
   vector_listZero dengan panjang =
   len(unique_words) dan lebar = len(text_list)
7  vector_uji = vector_listZero
8  i = 0
9  if (text_list[i] != null)
10     Split text_list menjadi satuan kata (word)
11     if (word != null)
12         j = 0
13         if (unique_words[j] != null)
14             if (word == unique_words[j])
15                 vector_uji[i][j] =
16                 vector_uji[i][j] + 1
17             endif
18             j = j + 1
19             Kembali ke nomor 13
20         else
21             word selanjutnya
22             Kembali ke nomor 11
23         endif
24     else
25         i = i + 1
26         Kembali ke nomor 9
27     endif
28 else

```

```

28     Output vector_uji
29     data_hasil = {}
30     if (vector_uji != null)
31         distance = {}
32         if(vector_latih != null)
33             Hitung jarak menggunakan euclidean
              distance
34             Simpan hasil jarak ke dalam dict
              distance
35             vector_latih selanjutnya
36             Kembali ke nomor 32
37         else
38             Output dict distance
39             nearest_neighbors = {}
40             Urut data secara ascending
              berdasarkan jarak (distance)
41             Select data dengan index < K
42             Output nearest_neighbors
43             Mencari jenis label berdasarkan
              data nearest_neighbors
44             Output sentiment_neighbors
45             count_positif = 0
46             count_negatif = 0
47             if (sentiment_neighbors != null)
48                 if (sentimen == 'positif')
49                     count_positif += 1
50                 else
51                     count_negatif += 1
52                 endif
53             sentiment_neighbors selanjutnya
54             Kembali ke nomor 47
55         else
56             Output count_positif &
              count_negatif
57             prob_positif = 0
58             prob_negatif = 0
59             Hitung probabilitas positif dan
              negatif
60             if (prob_positif >
              prob_negatif)
61                 Output label positif
62             else
63                 Output label negatif
64             endif
65             Simpan label ke dalam list
              data_hasil
66             vector_uji selanjutnya
67             Kembali ke nomor 30
68         endif
69     endif
70     else
71         Output Prediksi label data uji
              (data_hasil)
72     endif
73 endif
74 end

```

#### 4. 4. 8. Algoritme menu masuk aplikasi

Pada algoritme ini dijelaskan tentang menu masuk (*login*) dalam aplikasi yang dibuat, berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme keseluruhan menu aplikasi dapat dilihat pada Algoritme 4.13 berikut:

**Algoritme 4.13 Algoritme menu masuk aplikasi**

```
1  start
2  Tampilkan halaman masuk aplikasi
3  Input field username
4  Input field password
5  pilih = ''
6  if(pilih == 'masuk')
7      Cek record database berdasarkan inputan
8      if(record username != null)
9          if(record kata sandi == inputan kata
              sandi)
10             Buat sesi masuk untuk user
11             Akses menu awal
12         else
13             Kembali ke nomor 2
14         endif
15     else
16         Kembali ke nomor 2
17     endif
18 else
19     Kembali ke nomor 2
20 endif
```

#### 4. 4. 9. Algoritme menu aplikasi

Pada algoritme ini dijelaskan tentang menu-menu yang ada dalam aplikasi yang dibuat, berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme keseluruhan menu aplikasi dapat dilihat pada Algoritme 4.14 berikut:

**Algoritme 4.14 Algoritme menu aplikasi**

```
1  Input pilih = ''
2  if (pilih == 'beranda')
3      Akses menu beranda
4  else (if pilih == 'slangword')
5      Akses menu kamus slangword
6  else (if pilih == 'stopword')
7      Akses menu kamus stopword
8  else (if pilih == 'kata positif')
9      Akses menu kata positif
10 else (if pilih == 'kata negatif')
11     Akses menu kata negatif
12 else (if pilih == 'crawling')
13     Akses menu crawling
14 else (if pilih == 'preprocessing')
15     Akses menu preprocessing
16 else (if pilih == 'labeling')
17     Akses menu labeling
18 else (if pilih == 'pembagian data')
19     Akses menu pembagian data
```

```

20 else (if pilih == 'modeling')
21     Akses menu modeling
22 else (if pilih == 'pengujian')
23     Akses menu pengujian
24 else (if pilih == 'visualisai hasil')
25     Akses menu visualisai hasil
26 else (if pilih == 'keluar')
27     Hapus sesi masuk user
28 endif
29 end

```

#### 4. 4. 10. Algoritme menu beranda

Pada algoritme ini dijelaskan tentang menu beranda berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu beranda dapat dilihat pada Algoritme 4.15 berikut:

##### **Algoritme 4.15 Algoritme menu beranda**

```

1  Hitung jumlah data crawling
2  Hitung jumlah data preprocessing
3  Hitung jumlah data labeling
4  Hitung jumlah data latih
5  Hitung jumlah data uji
6  Tampilkan halaman beranda
7  Akses menu awal

```

#### 4. 4. 11. Algoritme menu *slang word*

Pada algoritme ini dijelaskan tentang menu *slang word* berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu *slang word* dapat dilihat pada Algoritme 4.16 berikut:

##### **Algoritme 4.16 Algoritme menu *slang word***

```

1  Select data slangword
2  Tampilkan halaman kamus slangword
3  aksi = ''
4  pilih = ''
5  if (aksi == 'tambah data')
6      Tampil modal tambah data
7      Input field kata slang dan kata asli
8      if (pilih == 'simpan')
9          Insert database
10         Kembali ke nomor 1
11     else
12         Tutup modal tambah data
13         Kembali ke nomor 1
14     endif
15 else if (aksi == 'import data')
16     Tampil modal import data
17     Input field file
18     if (pilih == 'import')
19         Insert database
20         Kembali ke nomor 1
21     else
22         Tutup modal import data
23         Kembali ke nomor 1
24     endif
25 else if (aksi == 'ubah record data')

```

```

26     Tampil modal ubah record data dan value
      input field
27     Input field kata slang dan kata asli
28     if (pilih == 'ubah')
29         Update record database
30         Kembali ke nomor 1
31     else
32         Tutup modal ubah record data
33         Kembali ke nomor 1
34     endif
35 else if (aksi == 'hapus record data')
36     Tampil modal hapus record data
37     if (pilih == 'hapus')
38         Delete record database
39         Kembali ke nomor 1
40     else
41         Tutup modal hapus record data
42         Kembali ke nomor 1
43     endif
44 else
45     Akses menu awal
46 endif

```

#### 4. 4. 12. Algoritme menu *stop word*

Pada algoritme ini dijelaskan tentang menu *stop word* berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu *stop word* dapat dilihat pada Algoritme 4.17 berikut:

##### **Algoritme 4.17 Algoritme menu *stop word***

```

1  Select data stopword
2  Tampilkan halaman kamus stopword
3  aksi = ''
4  pilih = ''
5  if (aksi == 'tambah data')
6      Tampil modal tambah data
7      Input field stopword
8      if (pilih == 'simpan')
9          Insert database
10         Kembali ke nomor 1
11     else
12         Tutup modal tambah data
13         Kembali ke nomor 1
14     endif
15 else if (aksi == 'import data')
16     Tampil modal import data
17     Input field file
18     if (pilih == 'import')
19         Insert database
20         Kembali ke nomor 1
21     else
22         Tutup modal import data
23         Kembali ke nomor 1
24     endif
25 else if (aksi == 'ubah record data')
26     Tampil modal ubah record data dan value
      input field

```

```

27     Input field stopword
28     if (pilih == 'ubah')
29         Update record database
30         Kembali ke nomor 1
31     else
32         Tutup modal ubah record data
33         Kembali ke nomor 1
34     endif
35 else if (aksi == 'hapus record data')
36     Tampil modal hapus record data
37     if (pilih == 'hapus')
38         Delete record database
39         Kembali ke nomor 1
40     else
41         Tutup modal hapus record data
42         Kembali ke nomor 1
43     endif
44 else
45     Akses menu awal
46 endif

```

#### 4. 4. 13. Algoritme menu kata positif

Pada algoritme ini dijelaskan tentang menu kata positif berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu kata positif dapat dilihat pada Algoritme 4.18 berikut:

##### **Algoritme 4.18 Algoritme menu kata positif**

```

1  Select data kata positif
2  Tampilkan halaman kata positif
3  aksi = ''
4  pilih = ''
5  if (aksi == 'tambah data')
6      Tampil modal tambah data
7      Input field kata positif
8      if (pilih == 'simpan')
9          Insert database
10         Kembali ke nomor 1
11     else
12         Tutup modal tambah data
13         Kembali ke nomor 1
14     endif
15 else if (aksi == 'import data')
16     Tampil modal import data
17     Input field file
18     if (pilih == 'import')
19         Insert database
20         Kembali ke nomor 1
21     else
22         Tutup modal import data
23         Kembali ke nomor 1
24     endif
25 else if (aksi == 'ubah record data')
26     Tampil modal ubah record data dan value
27     input field
28     Input field kata positif
29     if (pilih == 'ubah')

```

```

29         Update record database
30         Kembali ke nomor 1
31     else
32         Tutup modal ubah record data
33         Kembali ke nomor 1
34     endif
35 else if (aksi == 'hapus record data')
36     Tampil modal hapus record data
37     if (pilih == 'hapus')
38         Delete record database
39         Kembali ke nomor 1
40     else
41         Tutup modal hapus record data
42         Kembali ke nomor 1
43     endif
44 else
45     Akses menu awal
46 endif

```

#### 4. 4. 14. Algoritme menu kata negatif

Pada algoritme ini dijelaskan tentang menu kata negatif berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu kata negatif dapat dilihat pada Algoritme 4.19 berikut:

##### **Algoritme 4.19 Algoritme menu kata negatif**

```

1  Select data kata negatif
2  Tampilkan halaman kata negatif
3  aksi = ''
4  pilih = ''
5  if (aksi == 'tambah data')
6      Tampil modal tambah data
7      Input field kata negatif
8      if (pilih == 'simpan')
9          Insert database
10         Kembali ke nomor 1
11     else
12         Tutup modal tambah data
13         Kembali ke nomor 1
14     endif
15 else if (aksi == 'import data')
16     Tampil modal import data
17     Input field file
18     if (pilih == 'import')
19         Insert database
20         Kembali ke nomor 1
21     else
22         Tutup modal import data
23         Kembali ke nomor 1
24     endif
25 else if (aksi == 'ubah record data')
26     Tampil modal ubah record data dan value
27     input field
28     Input field kata negatif
29     if (pilih == 'ubah')
30         Update record database
31         Kembali ke nomor 1

```

```

31     else
32         Tutup modal ubah record data
33         Kembali ke nomor 1
34     endif
35 else if (aksi == 'hapus record data')
36     Tampil modal hapus record data
37     if (pilih == 'hapus')
38         Delete record database
39         Kembali ke nomor 1
40     else
41         Tutup modal hapus record data
42         Kembali ke nomor 1
43     endif
44 else
45     Akses menu awal
46 endif

```

#### 4. 4. 15. Algoritme menu *crawling*

Pada algoritme ini dijelaskan tentang menu *crawling* berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu *crawling* dapat dilihat pada Algoritme 4.20 berikut:

**Algoritme 4.20 Algoritme menu *crawling***

```

1  Select data crawling
2  Tampilkan halaman crawling
3  aksi = ''
4  pilih = ''
5  if (aksi == 'crawling data')
6      Akses proses crawling
7      Tampil hasil proses crawling
8      if (pilih == 'kembali')
9          Kembali ke nomor 1
10     endif
11 else if (aksi == 'import data')
12     Tampil modal import data
13     Input field file
14     if (pilih == 'import')
15         Insert database
16         Kembali ke nomor 1
17     else
18         Tutup modal import data
19         Kembali ke nomor 1
20     endif
21 else
22     Akses menu awal
23 endif

```

#### 4. 4. 16. Algoritme menu *preprocessing*

Pada algoritme ini dijelaskan tentang menu *preprocessing* berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu *preprocessing* dapat dilihat pada Algoritme 4.21 berikut:

**Algoritme 4.21 Algoritme menu *preprocessing***

```

1  Select data preprocessing
2  Tampilkan halaman preprocessing
3  aksi = ''

```



```

4  pilih = ''
5  if (aksi == 'preprocessing data')
6      Akses proses preprocessing
7      Tampil hasil proses preprocessing
8      if (pilih == 'kembali')
9          Kembali ke nomor 1
10     endif
11 else
12     Akses menu awal
13 endif

```

#### 4. 4. 17. Algoritme menu *labeling*

Pada algoritme ini dijelaskan tentang menu *labeling* berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu *labeling* dapat dilihat pada Algoritme 4.22 berikut:

##### **Algoritme 4.22 Algoritme menu *labeling***

```

1  Select data labeling
2  Tampilkan halaman labeling
3  aksi = ''
4  pilih = ''
5  if (aksi == 'labeling (manual)')
6      Select data tidak berlabel (hasil preprocessing)
7      Tampil modal labeling data, data tidak berlabel
8      Input select option
9      if (pilih != 'close')
10         if(option == 'positif' || option == 'negatif')
11             Update database
12             Kembali ke nomor 8
13         endif
14     else
15         Tutup modal labeling data
16         Kembali ke nomor 1
17     endif
18 else if (aksi == 'labeling (kamus)')
19     Akses proses labeling
20     Tampil hasil proses labeling
21     if (pilih == 'kembali')
22         Kembali ke nomor 1
23     endif
24 else
25     Akses menu awal
26 endif

```

#### 4. 4. 18. Algoritme menu pembagian data

Pada algoritme ini dijelaskan tentang menu pembagian data berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu pembagian data dapat dilihat pada Algoritme 4.23 berikut:

##### **Algoritme 4.23 Algoritme menu pembagian data**

```

1  Select data hasil labeling
2  Tampilkan halaman pembagian data, tab data latih

```

```

3 aksi = ''
4 if (aksi == 'split data')
5     Akses proses pembagian data
6     Kembali ke nomor 1
7 else if (aksi == 'data latih')
8     Tampilkan tab data latih
9     Kembali ke nomor 3
10 else if (aksi == 'data uji')
11     Tampilkan tab data uji
12     Kembali ke nomor 3
13 else
14     Akses menu awal
15 endif

```

#### 4. 4. 19. Algoritme menu *modeling*

Pada algoritme ini dijelaskan tentang menu *modeling* berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu *modeling* dapat dilihat pada Algoritme 4.24 berikut:

**Algoritme 4.24 Algoritme menu *modeling***

```

1 Select data model
2 Tampilkan halaman modeling
3 aksi = ''
4 pilih = ''
5 if (aksi == 'modeling data')
6     Akses proses modeling
7     Tampil hasil proses modeling
8     if (pilih == 'kembali')
9         Kembali ke nomor 1
10    endif
11 else if (aksi == 'hapus record data')
12     Tampil modal hapus record data
13     if (pilih == 'hapus')
14         Delete record database, file
15         Kembali ke nomor 1
16     else
17         Tutup modal hapus record data
18         Kembali ke nomor 1
19     endif
20 else
21     Akses menu awal
22 endif

```

#### 4. 4. 20. Algoritme menu pengujian

Pada algoritme ini dijelaskan tentang menu pengujian berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu pengujian dapat dilihat pada Algoritme 4.25 berikut:

**Algoritme 4.25 Algoritme menu pengujian**

```

1 Select data model
2 Tampilkan halaman pengujian
3 aksi = ''
4 pilih = ''
5 if (aksi == 'uji data')
6     Akses proses klasifikasi
7     Tampil hasil proses klasifikasi

```

```

8      if (pilih == 'kembali')
9          Kembali ke nomor 1
10     endif
11  else
12      Akses menu awal
13  endif

```

#### 4. 4. 21. Algoritme menu visualisasi hasil

Pada algoritme ini dijelaskan tentang menu visualisasi hasil berdasarkan *flowchart* yang dibuat sebelumnya. Algoritme menu visualisasi hasil dapat dilihat pada Algoritme 4.26 berikut:

**Algoritme 4.26 Algoritme menu visualisasi hasil**

```

1  Create histogram data
2  Create pie chart data
3  Create word cloud data
4  Hitung frekuensi data
5  Tampilkan halaman visualisasi hasil
6  Akses menu awal

```

### 4. 5. Pengujian

Pengujian merupakan salah satu hal yang perlu dilakukan dalam setiap pengembangan sistem untuk mengevaluasi, menganalisa dan mengetahui tingkat akurasi atau kesamaan hasil yang telah dicapai oleh sistem yang telah dirancang. Pada penelitian ini, dilakukan pengujian dari sisi akurasi, presisi dan *recall* pada implementasi algoritme *K-nearest neighbor* (KNN) dalam memprediksikan *label* untuk data uji. Selain pada sisi akurasi, presisi dan *recall* pengujian pada penelitian ini juga menguji nilai K berdasarkan variasi yang telah ditetapkan, yaitu K=3, K=5, K=7, K=9, dan K=11. Hasil sampel prediksi oleh algoritme KNN dengan nilai K=3 dapat dilihat pada Tabel 4.15 berikut:

**Tabel 4.15 Sampel data hasil prediksi**

No	<i>Tweet</i>	<i>Label</i> aktual	<i>Label</i> prediksi
1	Selain untuk memperkuat kompetensi literasi dan numerasi, tujuan lain program BDR adalah untuk membangun ikatan emosional dalam keluarga. <a href="https://t.co/Hks8kSVhl3">https://t.co/Hks8kSVhl3</a>	positif	positif
2	@schfess Kontra, udah capek batin belajar dari rumah, materi gak ada yang masuk dan karna gua SMK di mana harus banyak praktek.	negatif	negatif
3	Pandemi juga memaksa adik-adik kita belajar online dari rumah. Sayangnya, tidak semua dari mereka memiliki fasilitas internet dan gawai yang mendukung. Untungnya, banyak orang baik di luar sana yang tergerak	positif	negatif

No	<i>Tweet</i>	<i>Label</i> aktual	<i>Label</i> prediksi
	membantu adik-adik agar bisa tetap bisa belajar		
...	...	...	...
395	Seri Kepribadian Muslim 3 : Bersih dan Sehat - Diskon 20% menjadi Rp.20000 Gratis ongkos kirim hingga 40rb keseluruh Indonesia. #mainanbukuanak #promo #dirumahaja #belajardirumah #buku #membaca #booklover #bookstorm SINOPSIS:Aku suka bersih. Aku suka	positif	positif

Pada Tabel 4.15 Sampel data hasil prediksi, kolom *label* aktual merupakan data *label* yang diperoleh melalui proses *labeling*, sementara *label* prediksi merupakan data *label* hasil dari proses klasifikasi menggunakan KNN. Keseluruhan hasil prediksi data uji (395 data *tweet*) kemudian direpresentasikan ke dalam *confusion matrix*. Representasi *confusion matrix* untuk K=3 yang terbentuk dapat terlihat pada Tabel 4.16 berikut:

**Tabel 4.16 *Confusion matrix* pengujian K=3**

		Nilai Aktual	
		positif	negatif
Nilai Prediksi	positif	211	70
	negatif	50	64

Berdasarkan Tabel 4.16 *Confusion matrix* pengujian K=3, maka perolehan nilai akurasi, presisi dan *recall* menggunakan rumus yang telah dijabarkan dalam persamaan (3. 1), persamaan (3. 2), dan persamaan (3. 3) dapat dilihat pada Tabel 4.17 berikut:

**Tabel 4.17 Nilai pengujian K=3**

Pengujian		
Akurasi	$= \frac{211+64}{211+64+70+50}$	0.7 (70 %)
Presisi	$= \frac{211}{211+70}$	0.75 (75 %)

Pengujian		
<i>Recall</i>	$= \frac{211}{211+50}$	0.81 (81 %)

Pengujian di atas dilakukan secara berulang dengan variasi nilai K yang berbeda-beda. Sehingga dapat diketahui hasil pengujian secara keseluruhan adalah seperti Tabel 4.18 berikut:

**Tabel 4.18 Hasil pengujian**

	K=3	K=5	K=7	K=9	K=11
Akurasi	0.7	0.7	0.71	0.68	0.67
Presisi	0.75	0.75	0.74	0.72	0.72
<i>Recall</i>	0.81	0.84	0.86	0.84	0.81

Berdasarkan Tabel 4.18 Hasil pengujian, dapat diketahui bahwa hasil pengujian menunjukkan bahwa algoritme KNN mampu memperoleh nilai pengujian tertinggi menggunakan nilai K=7, dengan akurasi 71%, presisi 74%, dan *recall* 86%.

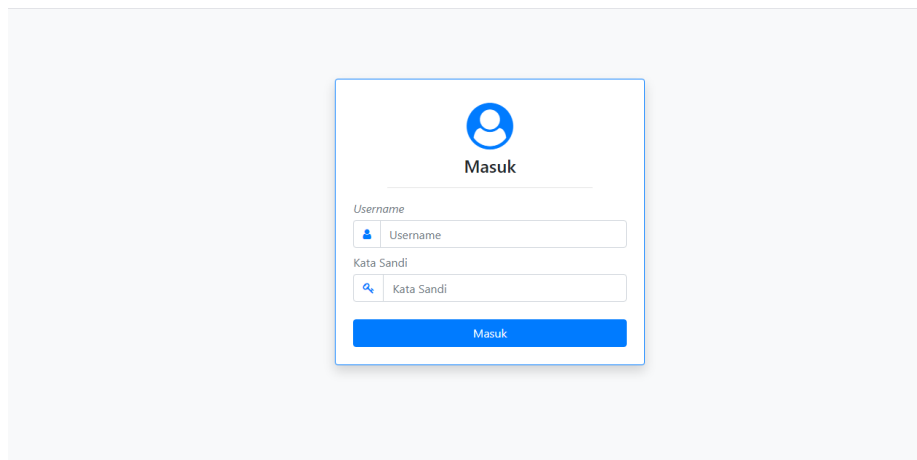
Sementara itu hasil analisis sentimen terhadap 3.954 *tweet* menunjukkan arah pandangan (sentimen) masyarakat Indonesia cenderung positif sebesar 76.56%, sementara sentimen negatif sebesar 23.44% pada periode Desember 2020.

#### 4. 6. Tampilan Layar Aplikasi

Dalam penerapannya, penelitian ini dituangkan ke dalam bentuk program aplikasi, berikut beberapa tampilan layar dari aplikasi yang dibuat.

##### 4. 6. 1. Tampilan layar masuk aplikasi

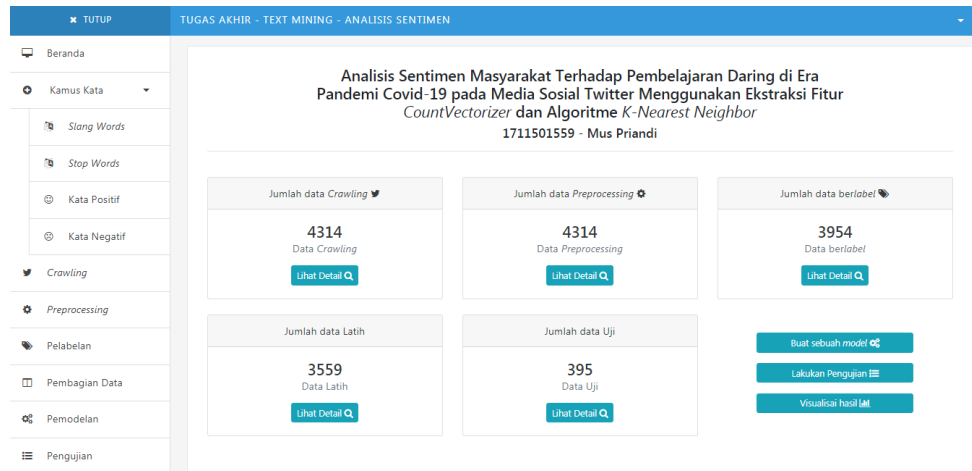
Tampilan layar masuk aplikasi dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.36 berikut:



**Gambar 4.36 Tampilan layar masuk aplikasi**

#### 4. 6. 2. Tampilan layar beranda

Tampilan layar beranda dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.37 berikut:

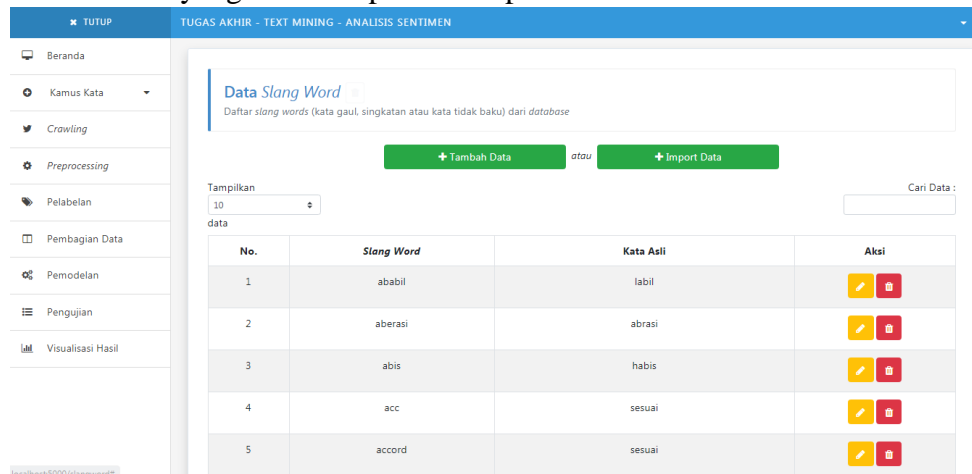


Gambar 4.37 Tampilan layar beranda

#### 4. 6. 3. Tampilan layar kamus kata

##### a. Tampilan layar kamus slang word

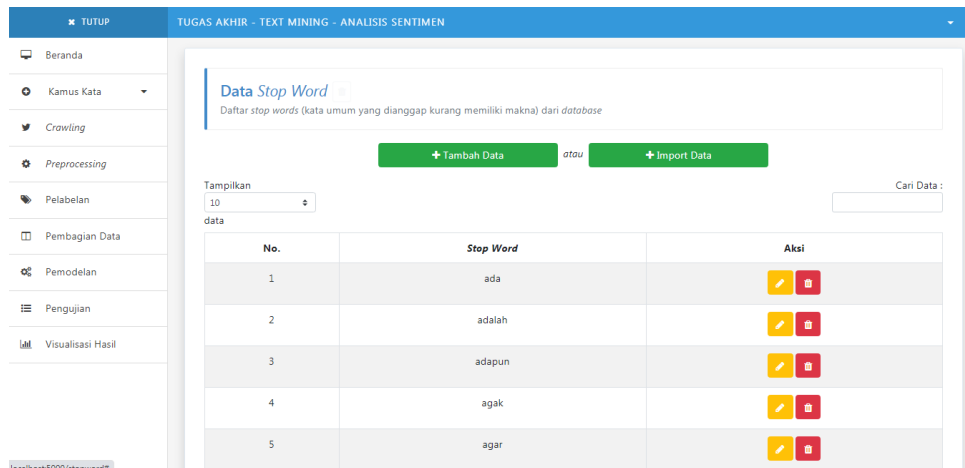
Tampilan layar kamus *slang word* dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.38 berikut:



Gambar 4.38 Tampilan layar kamus *slang word*

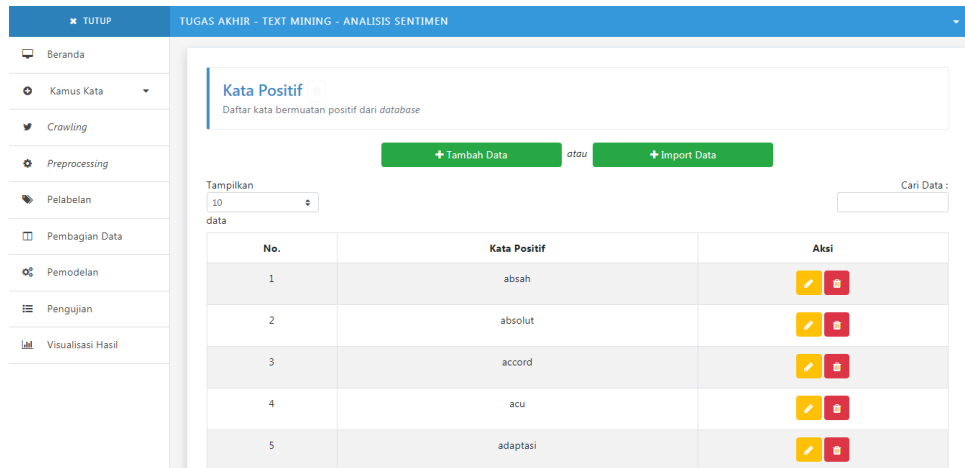
##### b. Tampilan layar kamus stop word

Tampilan layar kamus *stop word* dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.39 berikut:



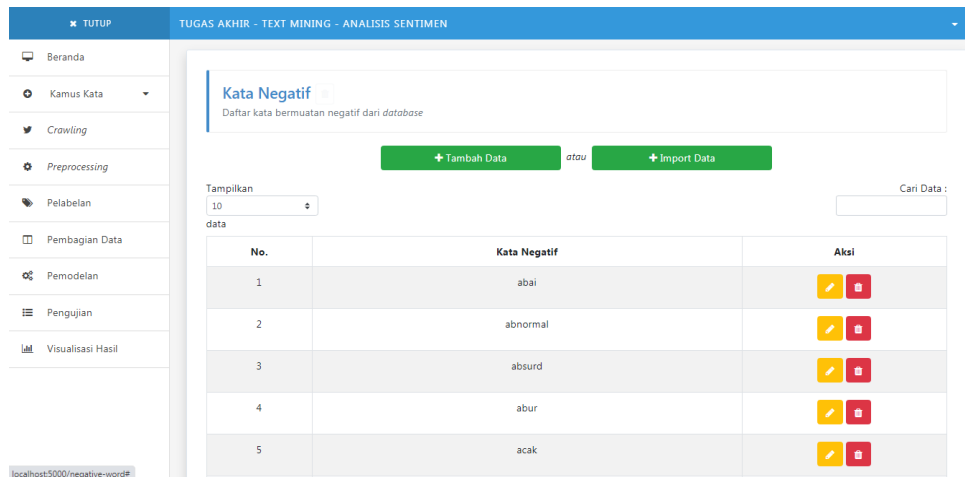
**Gambar 4.39 Tampilan layar kamus *stop word***

- c. Tampilan layar kamus kata positif  
Tampilan layar kamus kata positif dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.40 berikut:



**Gambar 4.40 Tampilan layar kamus kata positif**

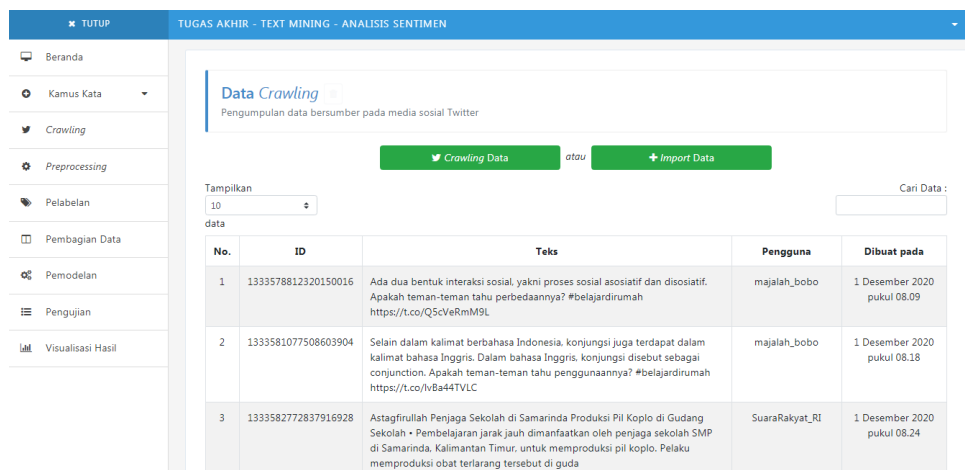
- d. Tampilan layar kamus kata negatif  
Tampilan layar kamus kata negatif dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.41 berikut:



**Gambar 4.41 Tampilan layar kamus kata negatif**

#### 4. 6. 4. Tampilan layar *crawling*

Tampilan layar *crawling* dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.42 berikut:

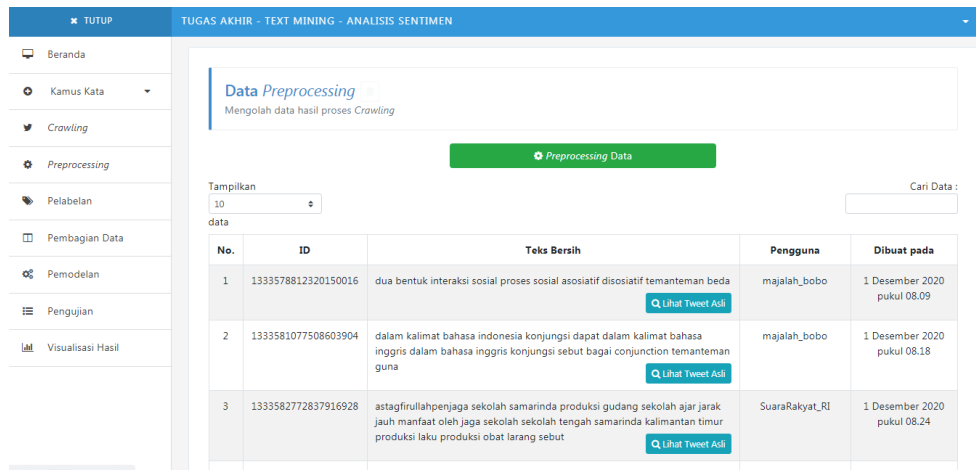


**Gambar 4.42 Tampilan layar *crawling***

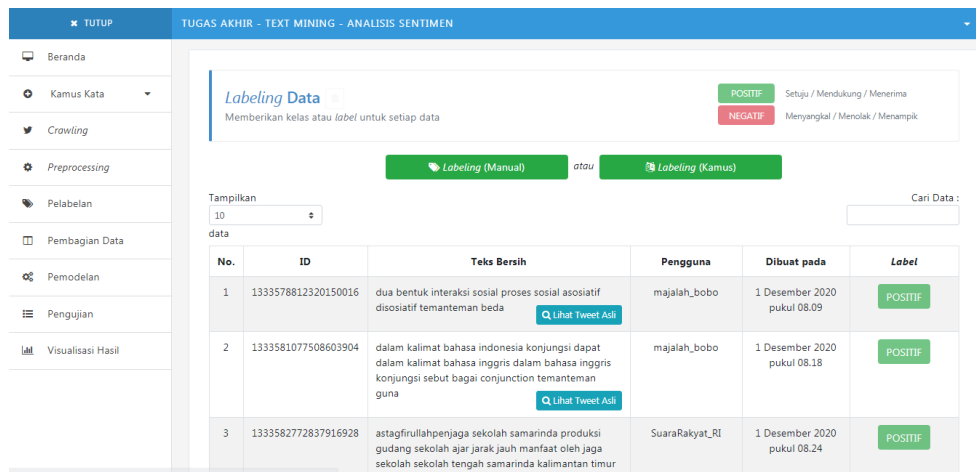
#### 4. 6. 5. Tampilan layar *preprocessing*

Tampilan layar *preprocessing* dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.43 berikut:





#### 4. 6. 6. Tampilan layar *labeling*



#### 4. 6. 7. Tampilan layar pembagian data

TUTUP

TUGAS AKHIR - TEXT MINING - ANALISIS SENTIMEN

Beranda

Kamus Kata

Crawling

Preprocessing

Pelabelan

Pembagian Data

Pemodelan

Pengujian

Visualisasi Hasil

Pembagian Data

Membagi data berlabel menjadi Data Uji dan Data Latih

Split Data

Data Latih

Data Uji

Tampilkan10data

Cari Data :

No.	ID	Teks Bersih	Pengguna	Dibuat pada	Label
1	1333578812320150016	dua bentuk interaksi sosial proses sosial asosiatif disosiatif temanteman beda <div>Lihat Tweet Asli</div>	majalah_bobo	1 Desember 2020 pukul 08.09	POSITIF
2	1333581077508603904	dalam kalimat bahasa indonesia konjungsi dapat dalam kalimat bahasa inggris dalam bahasa inggris konjungsi sebut sebagai conjunction temanteman guna <div>Lihat Tweet Asli</div>	majalah_bobo	1 Desember 2020 pukul 08.18	POSITIF
3	1333582772837916928	astagirullahpenjaga sekolah samarinda produksi	SuaraRakyat_RI	1 Desember 2020	POSITIF

Gambar 4.45 Tampilan layar pembagian data

#### 4. 6. 8. Tampilan layar *modeling*

Tampilan layar *modeling* dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.46 berikut:

TUTUP

TUGAS AKHIR - TEXT MINING - ANALISIS SENTIMEN

Beranda

Kamus Kata

Crawling

Preprocessing

Pelabelan

Pembagian Data

Pemodelan

Pengujian

Visualisasi Hasil

Pemodelan Data

Membuat pengetahuan (Model latih) dengan Data Latih

Modeling Data

Tampilkan10data

Cari Data :

No.	Nama Model	Sentimen Positif	Sentimen Negatif	Total Sentimen	Pilihan
1	sentiment_model[01-02-2021 170438].json	750	750	1500	Hapus
2	sentiment_model[01-02-2021 174854].json	200	200	400	Hapus
3	sentiment_model[01-02-2021 204343].json	500	500	1000	Hapus

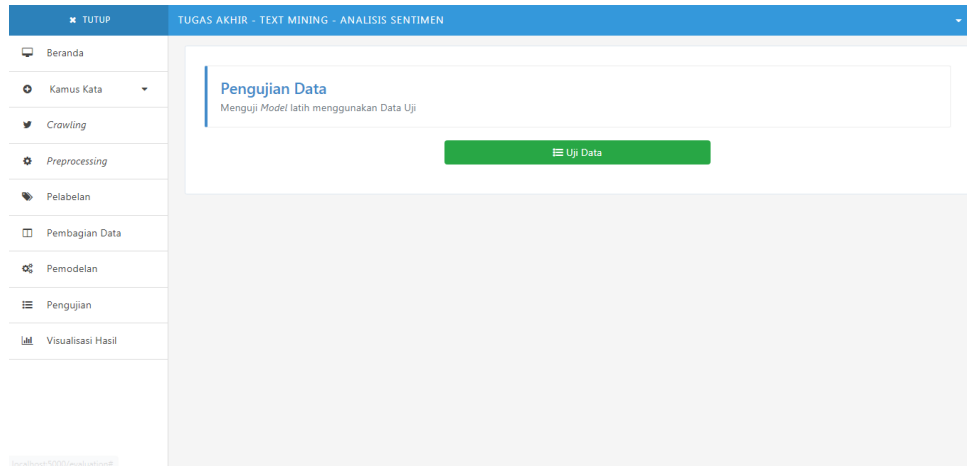
Tampil 1 hingga 3 dari 3 data

Sebelumnya1Selanjutnya

Gambar 4.46 Tampilan layar *modeling*

#### 4. 6. 9. Tampilan layar pengujian

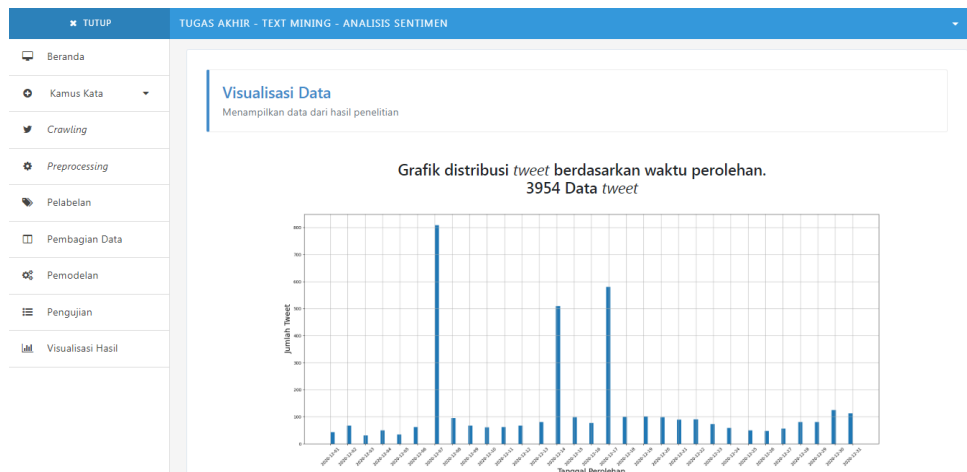
Tampilan layar pengujian dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.47 berikut:



**Gambar 4.47 Tampilan layar penguujian**

#### 4. 6. 10. Tampilan layar visualisasi hasil

Tampilan layar visualisasi hasil dari program aplikasi yang dibuat dapat dilihat pada Gambar 4.48 berikut:



**Gambar 4.48 Tampilan layar visualisasi hasil**

## **BAB V**

### **PENUTUP**

#### **5. 1. Kesimpulan**

Berdasarkan hasil pengujian dan evaluasi dari aplikasi yang dibuat menggunakan *dataset* dan algoritme yang diusulkan, maka dapat disimpulkan bahwa:

- a. Berdasarkan 3.954 *tweet*, arah pandangan (sentimen) masyarakat Indonesia terhadap pembelajaran daring cenderung ke arah sentimen positif sebesar 76.56% pada periode Desember 2020.
- b. Tahap utama yang terdapat dalam penelitian ini antara lain: *crawling*, *preprocessing*, *labeling*, *modeling*, klasifikasi *K-Nearest Neighbor* (KNN). Tahap *preprocessing* yang baik menjadi penentu dalam terbentuknya hasil yang optimal untuk tahap selanjutnya. Penggunaan kamus sentimen dapat membantu proses pemberian kelas atau *label* juga meminimalisir waktu dan usaha dalam melakukan proses *labeling*.
- c. Penggunaan ekstraksi fitur *CountVectorizer* dan algoritme *K-Nearest Neighbor* (KNN) dalam melakukan analisis sentimen dapat berjalan dengan baik, dengan nilai pengujian dan evaluasi tertinggi yang diperoleh sebesar: akurasi 71%, presisi 74% dan *recall* 86% menggunakan nilai  $K=7$ .

#### **5. 2. Saran**

Adapun saran yang dapat peneliti berikan sebagai pengembangan lebih lanjut untuk aplikasi ini agar dapat berjalan lebih baik lagi adalah sebagai berikut:

- a. Menambahkan kata kunci pencarian *tweet* sehingga dapat menghasilkan pandangan (sentimen) yang lebih beragam.
- b. Menambahkan kamus kata (*stop word*, *slang word*, kata positif dan kata negatif) seiring dengan keberagaman bahasa pada *tweet* yang akan diproses.
- c. Melakukan proses pelabelan dengan cara manual dengan bantuan ahli atau pakar dalam bidang bahasa.
- d. Merubah proses pelabelan menggunakan kamus sentimen, semula berdasarkan frekuensi kata positif dan negatif menjadi menggunakan skor untuk tiap kata positif dan negatif.
- e. Melakukan pembagian data dengan rasio pembagian yang lebih beragam untuk mendapatkan data yang optimal.
- f. Menambah kemungkinan nilai  $K$  yang dalam proses klasifikasi data uji untuk mencari nilai pengujian yang lebih optimal.
- g. Menggunakan *pustaka* atau *plugin* pemrograman yang dapat meringkas waktu pemrosesan data.

## DAFTAR PUSTAKA

- Afrizal, S. *et al.* (2019) 'Implementasi Metode Naïve Bayes untuk Analisis Sentimen Warga Jakarta Terhadap Kehadiran Mass Rapid Transit', *Jurnal Informatik*, 4221, pp. 157–168.
- Antinasari, P., Perdana, R. S. and Fauzi, M. A. (2017) 'Analisis Sentimen Tentang Opini Film Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dengan Perbaikan Kata Tidak Baku', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1(12), pp. 1733–1741.
- Aribowo, A. S. (2018) 'Analisis Sentimen Publik pada Program Kesehatan Masyarakat menggunakan Twitter Opinion Mining', *Seminar Nasional Informatika Medis (Snimed)*, pp. 17–23.
- Buntoro, G. A. (2017) 'Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter', *Integer Journal*, 2(1), pp. 32–41.
- Daeli, N. O. F. and Adiwijaya (2020) 'Sentiment Analysis on Movie Reviews Using Information Gain and K-Nearest Neighbor', *Journal of Data Science And Its Applications*, 3(1), pp. 1–7. doi: 10.34818/JDSA.2020.3.22.
- Ferdiana, R. *et al.* (2019) 'Dataset Indonesia untuk Analisis Sentimen', *JNTETI*, 8(4), pp. 334–339.
- Fitriyyah, S. N. J., Safriadi, N. and Pratama, E. E. (2019) 'Analisis Sentimen Calon Presiden Indonesia 2019 dari Media Sosial Twitter Menggunakan Metode Naive Bayes', *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 5(3), pp. 279–285.
- Liu, B., Hu, M. and Cheng, J. (2005) 'Opinion Observer: Analyzing and Comparing Opinions on the Web', *Proceedings of the 14th International World Wide Web Conference (WWW-2005)*.
- Medhat, W., Hassan, A. and Korashy, H. (2014) 'Sentiment analysis algorithms and applications: A survey', *Ain Shams Engineering Journal*, 5(4), pp. 1093–1113. doi: 10.1016/j.asej.2014.04.011.
- Munawar (2019) 'Sistem Pendeteksi Berita Palsu (Fake News) Di Media Sosial Dengan Teknik Data Mining Scikit Learn'.
- Nurulbaiti, F. and Retno Subekti, M. S. (2020) 'Analisis Sentimen Terhadap Data Tweet Untuk Badan Penyelenggara Jaminan Sosial (BPJS) Menggunakan Program R', *Jurnal Pendidikan Matematika dan Sains*, pp. 1–9.
- Oktasari, L., Chrisnanto, Y. H. and Yuniarti, R. (2016) 'Text Mining Dalam Analisis Sentimen Asuransi Menggunakan Metode Naïve Bayes Classifier', *Prosiding SNST ke-7*, pp. 37–42.
- Ristyawati, A. (2020) 'Efektifitas Kebijakan Pembatasan Sosial Berskala Besar Dalam Masa Pandemi Corona Virus 2019 oleh Pemerintah Sesuai Amanat

- UUD NRI Tahun 1945', *Administrative Law & Governance Journal*, 3(2), pp. 240–249.
- Romadloni, N. T., Santoso, I. and Budilaksono, S. (2019) 'Perbandingan Metode Naive Bayes, KNN Dan Decision Tree Terhadap Analisis Sentimen Transportasi KRL Commuter Line', *Jurnal IKRA-ITH Informatika*, 3(2), pp. 1–9.
- Sadikin, A. and Hamidah, A. (2020) 'Pembelajaran Daring di Tengah Wabah Covid-19 (Online Learning in the Middle of the Covid-19 Pandemic)', *BIODIK: Jurnal Ilmiah Pendidikan Biologi*, 6(1), pp. 214–224. doi: <https://doi.org/10.22437/bio.v6i2.9759>.
- Santoso, E. B. and Nugroho, A. (2019) 'Analisis Sentimen Calon Presiden Indonesia 2019 Berdasarkan Komentar Publik di Facebook', *Jurnal Eksplora Informatika*, 9(1), pp. 60–69. doi: 10.30864/eksplora.v9i1.254.
- Sari, F. V. and Wibowo, A. (2019) 'Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi', *Jurnal SIMETRIS*, 10(2), pp. 681–686.
- Septian, J. A., Fahrudin, T. M. and Nugroho, A. (2019) 'Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor', *Journal of Intelligent Systems And Computation*, 1(1), pp. 43–49.
- Statista.com, (2020). Leading countries based on number of Twitter users as of October 2020. [online] Available at: <https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/> [Accessed 05 Jan. 2021].
- Sudiantoro, A. V. and Zuliarso, E. (2018) 'Analisis Sentimen Twitter Menggunakan Text Mining Dengan Algoritma Naïve Bayes Classifier', *Prosiding SINTAK*, pp. 398–401.
- Wahid, D. H. and SN, A. (2017) 'Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity', *Jurnal IJCCS*, 10(2), pp. 207–218.
- Watrianthos, R. (2020) 'Analisis Pembelajaran Daring di Era Pandemic Covid-19', *Merdeka Kreatif di Era Pandemi Covid-19*, pp. 55–64.
- Wijoyo, H. (2020) 'Guru Milenial dan Covid-19', *Merdeka Kreatif di Era Pandemi Covid-19*, pp. 27–41.