



UNIVERSITAS TRISAKTI

**Analisis Sentimen Mengenai Undang - Undang Tpks pada Media Sosial
Twitter Menggunakan Metode Support Vector Machine dan K-Nearest
Neighbour**

SKRIPSI

Arviandri Naufal Zaki

064001800035

FAKULTAS TEKNOLOGI INDUSTRI

PRODI TEKNIK INFORMATIKA

UNIVERSITAS TRISAKTI

JAKARTA BARAT



UNIVERSITAS TRISAKTI

**Analisis Sentimen Mengenai Undang - Undang Tpks pada Media Sosial
Twitter Menggunakan Metode Support Vector Machine dan K-Nearest
Neighbour**

Diajukan Sebagai Syarat Dalam Memperoleh Gelar Sarjana Strata Satu (S1)
Program Studi Teknik Informatika

SEMINAR HASIL TUGAS AKHIR

Disusun Oleh:

Nama : Arviandri Naufal Zaki

NIM: 064001800035

FAKULTAS TEKNOLOGI INDUSTRI

PRODI TEKNIK INFORMATIKA

UNIVERSITAS TRISAKTI

JAKARTA BARAT

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Arviandri Naufal Zaki

NIM : 064001800035

Tanda Tangan :

Tanggal : 11 Mei 2022

HALAMAN PENGESAHAN

**Analisis Sentimen Mengenai Undang - Undang TPKS pada Media Sosial
Twitter Menggunakan Metode Support Vector Machine dan K-Nearest
Neighbour**

TUGAS AKHIR

Diajukan Sebagai Syarat Dalam Memperoleh Gelar Sarjana Strata
Satu (S1) Program Studi Teknik Informatika Universitas Trisakti

Disusun Oleh :

Arviandri Naufal Zaki

064001800035



Jakarta, 11 Mei 2022

Pembimbing Utama



Dian Pratiwi, ST, MTI

Pembimbing Pendamping



Syandra Sari, S.Kom, M.Kom

KATA PENGANTAR

Puji Syukur saya ucapkan kepada Allah SWT, Tuhan Yang Maha Esa atas berkah, rahmat dan ridho-Nya, saya diberi kesempatan untuk menyelesaikan skripsi ini. Penulisan Skripsi ini dilakukan untuk memenuhi salah satu syarat mencapai gelar Program Sarjana (S1) Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, mulai dari saya masuk ke Universitas ini sampai pada masa penyusunan skripsi ini saya tidak bisa sampai di titik ini, dan selama penyusunan skripsi ini banyak rintangan yang telah dihadapi. Oleh karena itu, pada kesempatan ini saya mengucapkan terima kasih kepada:

1. Allah SWT yang telah memberikan kesempatan untuk saya menyelesaikan skripsi ini.
2. Kedua Orang Tua yang telah memberikan doa dan dukungan selama mengerjakan
3. Dian Pratiwi, ST, MTI sebagai Pembimbing Utama
4. Syandra Sari, S.Kom, M.Kom sebagai Pembimbing Pendamping
5. Ridho Rachmat Giffary, Tasya Aulia, Kino 2017, Farhan 2017, Nabilah 2017 dan teman - teman lain yang membantu dalam proses pembuatan skripsi ini semasa perkuliahan
6. Semua pihak yang tidak dapat disebutkan satu persatu yang telah memberikan Dukungan

Akhir kata, saya mohon maaf atas segala kesalahan jika terdapat kesalahan pada penulisan skripsi ini dan saya juga berharap kepada Allah SWT agar dapat membalas segala kebaikan seluruh pihak yang telah membantu. Semoga skripsi ini dapat membawa manfaat bagi pengembangan ilmu selanjutnya.

Jakarta, 11 Mei 2022

Arviandri Naufal Zaki

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Trisakti, saya yang bertanda tangan di bawah ini:

Nama : Arviandri Naufal Zaki
NIM : 064001800035
Program Studi : Teknik Informatika
Fakultas : Fakultas Teknologi Industri
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Trisakti **Hak Bebas Royalti Noneklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul:

ANALISIS SENTIMEN MENGENAI UNDANG - UNDANG TPKS PADA MEDIA SOSIAL TWITTER MENGGUNAKAN METODE SUPPORT VECTOR MACHINE DAN K-NEAREST NEIGHBOUR

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Trisakti berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Jakarta
Pada Tanggal 11 Mei 2022
Yang Menyatakan

(Arviandri Naufal Zaki)

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
ABSTRAK	xii
BAB I.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II	5
2.1 Kajian Pustaka	5
2.2 Landasan Teori	6
2.2.1 Twitter.....	6
2.2.2 Python	6
2.2.3 Analisis Sentimen	7
2.2.4 <i>Scraping</i> Data	7
2.2.5 Preprocessing.....	7
2.2.6 TF-IDF	7
2.2.7 <i>Lexicon Based Features</i>	11
2.2.8 <i>Valence Aware Dictionary and sEntiment Reasoner (VADER)</i>	11
2.2.9 <i>Support Vector Machine (SVM)</i>	11
2.2.10 <i>K-Nearest Neighbour (KNN)</i>	13
BAB III.....	16
3.1 Metode Penelitian.....	16
3.1.1 Pengumpulan Data.....	17
3.1.2 Pengolahan Data	17
3.1.2.1. Case Folding	17
3.1.2.2. Cleansing.....	17

3.1.2.3. Tokenizing	18
3.1.2.4. Normalization.....	18
3.1.2.5. Stopword Removing	18
3.1.3 Labeling Data	18
3.1.4 Pembobotan kata (TF-IDF).....	19
3.1.5 Mengklasifikasikan Data	19
3.1.6 Visualisasi	19
3.2 Metode Pelabelan Data.....	20
3.3 Metode Klasifikasi	21
BAB IV	22
4.1 Scraping data.....	22
4.2 Pre – Processing.....	24
4.2.1 Case Folding	25
4.2.2 Cleansing	26
4.2.4 Normalize	29
4.2.5 Stopword removal.....	30
4.3 Pelabelan Data.....	31
4.4 Pembobotan Kata.....	36
4.5 Klasifikasi Data Menggunakan SVM.....	38
4.6 Klasifikasi Data Menggunakan KNN.....	43
BAB V	46
5.1 Kesimpulan.....	46
5.2 Saran	46
Daftar Referensi	47

DAFTAR GAMBAR

Gambar III-1 <i>Flowchart</i> dari Penelitian	16
Gambar III-2 <i>Flowchart</i> dari Pelabelan Data	20
Gambar III-3 <i>Flowchart</i> dari Klasifikasi	21
Gambar IV-1 Library yang Digunakan pada Scraping Data	22
Gambar IV-2 <i>Query</i> Pencarian Tweet	23
Gambar IV-3 Kode untuk <i>Scraping</i> Data Twitter.....	23
Gambar IV-4 Kode untuk Drop Data Duplikat.....	23
Gambar IV-5 Library yang Digunakan Pada <i>Pre-Processing</i>	25
Gambar IV-6 Kode untuk <i>Case Folding</i>	25
Gambar IV-7 Kode untuk <i>Cleansing</i> Tahap 1	26
Gambar IV-8 Kode untuk <i>Cleansing</i> Tahap 2	27
Gambar IV-9 Kode untuk <i>Cleansing</i> Tahap 3	27
Gambar IV-10 Kode untuk <i>Tokenizing</i>	28
Gambar IV-11 Kode untuk <i>Normalize</i>	29
Gambar IV-12 Kode untuk <i>Stopword Removal</i>	30
Gambar IV-13 Library yang Digunakan pada Pelabelan Data ke 1	31
Gambar IV-14 Library yang Digunakan pada Pelabelan Data ke 2	31
Gambar IV-15 Kode untuk <i>Detokenize</i>	32
Gambar IV-16 Isi dari Kamus <i>Inset</i>	33
Gambar IV-17 Kode untuk Mengganti <i>Lexicon</i> pada Vader Tahap 1	33
Gambar IV-18 Kode untuk Mengganti <i>Lexicon</i> pada Vader Tahap 2	34
Gambar IV-19 Kode untuk Mendapatkan Skor Sentimen (<i>Polarity Score</i>)	34
Gambar IV-20 Kode untuk Melabelkan Data Berdasarkan Skor Sentimen	35
Gambar IV-21 Hasil dari Proses Pelabelan Data	35
Gambar IV-22 Diagram Pie dari Hasil Pelabelan	36
Gambar IV-23 <i>Wordcloud</i> dari Hasil Pelabelan (Data Positif).....	36
Gambar IV-24 <i>Wordcloud</i> dari Hasil Pelabelan (Semua Data)	36
Gambar IV-25 <i>Wordcloud</i> dari Hasil Pelabelan (Data Negatif)	36
Gambar IV-26 Library yang Digunakan pada Pembobotan Kata	37
Gambar IV-27 Kode untuk Pengubahan Data Netral ke Positif	37
Gambar IV-28 Kode untuk <i>Splitting Data</i>	38
Gambar IV-29 Kode untuk Pembobotan Kata Menggunakan TF-IDF	38

Gambar IV-30 Kode dari Klasifikasi SVM Kernel Linear ke 1	39
Gambar IV-31 Kode dari Klasifikasi SVM Kernel Linear ke 2	39
Gambar IV-32 Kode dari Klasifikasi SVM Kernel RBF ke 1	39
Gambar IV-33 Kode dari Klasifikasi SVM Kernel RBF ke 2	39
Gambar IV-34 Kode dari Visualisasi Akurasi SVM Linear ke 1	40
Gambar IV-35 Kode dari Visualisasi Akurasi SVM Linear ke 2	40
Gambar IV-36 Kode dari Visualisasi Akurasi SVM RBF ke 1	40
Gambar IV-37 Kode dari Visualisasi Akurasi SVM RBF ke 2	41
Gambar IV-38 Hasil dari Visualisasi Akurasi SVM Linear ke 1	41
Gambar IV-39 Hasil dari Visualisasi Akurasi SVM Linear ke 2	42
Gambar IV-40 Hasil dari Visualisasi Akurasi SVM RBF ke 1	42
Gambar IV-41 Hasil dari Visualisasi Akurasi SVM RBF ke 2	43
Gambar IV-42 Kode dari Klasifikasi KNN	43
Gambar IV-43 Kode dari Visualisasi Akurasi KNN ke 1	44
Gambar IV-44 Kode dari Visualisasi Akurasi KNN ke 2	44
Gambar IV-45 Hasil dari Visualisasi Akurasi KNN ke 1	44
Gambar IV-46 Hasil dari Visualisasi Akurasi KNN ke 2	45

DAFTAR TABEL

Tabel IV-1 Hasil dari <i>Case Folding</i>	26
Tabel IV-2 Hasil dari <i>Cleansing</i>	28
Tabel IV-3 Hasil dari <i>Tokenizing</i>	28
Tabel IV-4 Hasil dari <i>Normalize</i>	29
Tabel IV-5 Hasil dari <i>Stopword Removal</i>	30

ABSTRAK

Nama : Arviandri Naufal Zaki

Program Studi : Teknik Informatika

Judul : Analisis Sentimen Mengenai Undang - Undang TPKS pada Media Sosial Twitter Menggunakan Metode Support Vector Machine dan K-Nearest Neighbour

Twitter adalah media sosial yang banyak digunakan oleh masyarakat Indonesia maupun Dunia. Twitter juga dimanfaatkan untuk berbagi kabar dan opini pribadi, memasarkan produk, sampai mengkritik suatu kebijakan atau peraturan. Opini yang diposting sebagai tweet di Twitter juga dapat digunakan sebagai tolak ukur apakah kebijakan yang dikeluarkan banyak yang mendukungnya atau sebaliknya. Untuk memperoleh tolak ukur tersebut maka digunakanlah analisis sentimen untuk memisahkan opini positif dengan opini negatif. Dari pengambilan data untuk diproses maka digunakanlah *scraping* dari website Twitter untuk mendapatkannya. Setelah itu dilakukan proses awal sebelum data diolah yaitu *Preprocessing* untuk menghilangkan bagian yang tidak berguna dalam pengolahan data. Lalu dilakukan teknik Support Vector Machine dan K-Nearest Neighbour untuk mengklasifikasikan opini positif dan negatif guna untuk membandingkan manakah yang lebih banyak dari opini tersebut lalu dijadikanlah tolak ukur terhadap suatu kebijakan.

Kata Kunci : *Analisis Sentimen, Twitter, UU TPKS, Support Vector Machine, K-Nearest Neighbour, Scraping*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Akhir - akhir ini tindak pidana kekerasan seksual banyak terjadi di masyarakat, menurut Komnas Perempuan tercatat bahwa terjadi 14.719 kasus kekerasan seksual terhadap perempuan sepanjang tahun 2020. Sebenarnya pemerintah telah merancang rancangan undang – undang tindak pidana kekerasan seksual sejak tahun 2016, namun baru disahkan pada tahun 2022 setelah desakan dari berbagai pihak, disahkannya UU TPKS menimbulkan banyak pro dan kontra serta menyebabkan masyarakat beropini di media sosial. Salah satunya yang mendukung berpendapat bahwa undang – undang ini akan membuat pelaku kejahatan seksual dihukum sesuai dengan apa yang dilakukannya, sedangkan salah satu yang menentang berpendapat bahwa undang – undang ini akan menciptakan suatu pemikiran bahwa seks bebas itu diperbolehkan. Salah satu media sosial yang sering digunakan untuk beropini oleh masyarakat Indonesia adalah twitter.

Twitter oleh masyarakat Indonesia dimanfaatkan untuk berbagai hal seperti berkomunikasi dengan orang lain secara publik atau personal, berbagi kabar dan opini pribadi, berjualan, sampai mengkritik atau memuji akan suatu hal. Dikarenakan informasi yang berada di Twitter juga dibatasi sekitar 280 karakter biasanya pengguna hanya mengirim suatu hal yang pendek [1].

Pemerintah juga memanfaatkan *platform* ini untuk mengetahui respon masyarakat kepada kebijakan yang baru dikeluarkan seperti pada penelitian ini. Oleh karena itu, pengguna Twitter dapat beropini tentang kebijakan yang dikeluarkan dipengaruhi oleh emosi yang dapat diklasifikasikan untuk menentukan polarisasinya, yaitu positif atau negatif tentang *tweet* mengenai kebijakan pemerintah pada penelitian ini..

Analisis sentimen yaitu kegiatan mengolah kata untuk menghasilkan suatu sentimen (positif atau negatif), Analisis sentimen bertujuan salah satunya yaitu untuk mendapatkan suatu opini dari kebijakan pemerintah yang baru dikeluarkan kemudian opini tersebut diklasifikasikan ke dalam sentimen positif dan negatif. Teknik yang dipakai untuk mengambil data dari Twitter sebelum di analisis yaitu menggunakan teknik *Scraping* yaitu mengambil data langsung dari website Twitter. Lalu teknik yang digunakan untuk memberikan sentimen (label) adalah Valence Aware Dictionary and sEntiment Reasoner (VADER) dikarenakan teknik ini memiliki akurasi yang tinggi serta bisa digunakan untuk sentiment negasi lalu teknik yang digunakan untuk mengklasifikasi data tersebut yaitu *K-Nearest Neighbor (KNN)* dan *Support Vector Machine (SVM)* dikarenakan kedua model klasifikasi tersebut memiliki kelebihan masing – masing pada penelitian sebelumnya tentang perbandingan akurasi keduanya yaitu tingkat akurasinya yang cukup tinggi untuk SVM (89,7 %) sedangkan KNN yaitu dapat memproses data yang besar dalam waktu singkat (1.113 data dalam waktu 0,0160 detik) [2].

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disusun sebelumnya, rumusan masalahnya yaitu :

- a. Bagaimana cara mengambil dan mengolah data tweet yang berasal dari Twitter untuk perhitungan *Support Vector Machine (SVM)* dan *K-Nearest Neighbour (KNN)*.
- b. Bagaimana tingkat keakuratan dari *K-Nearest Neighbour (KNN)* dan *Support Vector Machine (SVM)* pada analisis sentimen di Twitter mengenai Undang – Undang TPKS.
- c. Bagaimana hasil klasifikasi dari tweet menggunakan *Support Vector Machine (SVM)* dan *K-Nearest Neighbour (KNN)* pada analisis sentimen di Twitter mengenai Undang – Undang TPKS.

- d. Bagaimana cara implementasi sentimen analisis secara *hybrid* menggunakan *lexicon based* dan SVM serta KNN.

1.3 Batasan Masalah

Berdasarkan latar belakang yang telah disusun sebelumnya, rumusan masalahnya yaitu :

- a. Data yang digunakan adalah tweet berbahasa Indonesia dengan kata kunci “uu tpks” dari Twitter.
- b. Metode yang digunakan untuk klasifikasi adalah *Support Vector Machine* (SVM) dan *K-Nearest Neighbour* (KNN).
- c. Metode yang digunakan untuk pelabelan data adalah Valence Aware Dictionary and sEntiment Reasoner (VADER).
- d. Data diambil mulai tanggal 12 April 2022 sampai dengan 24 April 2022

1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini yaitu untuk mengklasifikasi tweet berdasarkan positif dan negatifnya untuk mengetahui keakuratan dari kedua metode ini yaitu *Support Vector Machine* (SVM) dan *K-Nearest Neighbour* (KNN) dalam menganalisis sentimen (emosi) pengguna Twitter mengenai Undang – Undang TPKS.

1.5 Manfaat Penelitian

Manfaat penulisan tugas akhir ini adalah:

- a. Memperoleh visualisasi sentimen analisis berupa *feedback* dari pengguna twitter mengenai Undang – Undang TPKS dengan menggunakan metode SVM dan KNN.

- b. Memperoleh perbandingan akurasi dari penggunaan linear dan RBF untuk metode SVM pada penelitian ini.
- c. Memperoleh perbandingan akurasi dari metode SVM dan KNN pada penelitian ini.
- d. Bagi pemerintah dapat mengetahui sentimen yang didapatkan dari pengesahan UU TPKS dan dapat digunakan sebagai rujukan untuk memperbaharui kebijakan lain yang dikeluarkan.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Melakukan penulisan untuk penelitian membutuhkan sebuah panduan dan dukungan dari penelitian yang sudah terlebih dahulu ada sebelumnya yang juga berkaitan dengan penelitian yang sedang berlangsung.

Pada penelitian yang telah dilakukan sebelumnya dapat disimpulkan bahwa penelitiannya memiliki tantangan terbesar salah satunya dalam melakukan pengambilan data dari Twitter masih menggunakan API yang diberikan oleh Twitter. Dengan menggunakan cara tersebut maka data tweet yang didapatkan hanya dalam batas waktu seminggu ke belakang dari hari ini dan pengambilan data setiap hari dibatasi hanya 50.000 tweet per hari [1].

Kemudian dari hasil dari data yang diambil tersebut dilakukan preprosesing pada data tersebut dan dilanjutkan dengan *labeling* untuk menentukan positif dan negatifnya lalu dilakukan ekstraksi data menggunakan TF-IDF untuk pembobotan kata, lalu dilakukan perhitungan menggunakan *Naïve Bayes* sehingga dapat dilakukannya penentuan sentiment (positif atau negatif) pada penelitian tersebut [3] .

Lalu berdasarkan dari penelitian sebelumnya dapat diketahui bahwa akurasi dari metode *Naïve Bayes* 95 % sentimen cenderung negatif tetapi ketika memakai metode *Support Vector Machine* akurasinya 90% sentiment cenderung positif. Dapat disimpulkan bahwa publik memiliki perasaan baik terhadap orang tersebut [4].

Lalu di penelitian sebelumnya yang menggunakan cara analisis yang hampir sama yaitu penggabungan antara metode Vader dengan KNN menghasilkan hasil akurasi yang cukup baik yaitu 75 % [5].

Lalu pada suatu penelitian terdahulu yang bertopik hampir menyerupai yaitu tentang Rancangan Undang – Undang TPKS peneliti menggunakan beberapa klasifikasi yaitu SVM, Bernoulli, dan Logistic Regression yang masing – masing menghasilkan keakuratan hingga 63 %, 65 %, dan 65 % serta peneliti menyimpulkan bahwa kata “kekerasan”, “korban” dan “seksual” pada topik ini mengekspresikan sentimen positif [6].

2.2 Landasan Teori

2.2.1 Twitter

Twitter adalah platform sosial media yang dapat digunakan untuk mengirimkan suatu postingan (*tweet*) dalam bentuk foto maupun teks dengan terbatas yaitu 280 karakter.

Selain daripada itu Twitter juga banyak digunakan oleh masyarakat karena kepraktisannya dalam menyampaikan sesuatu seperti mengungkapkan pendapat, berkomunikasi, dan lain sebagainya [1].

2.2.2 Python

Python adalah bahasa pemrograman dengan kode sumber yang terbuka (*open source*) yang dapat digunakan untuk membuat program secara independent (*standalone*) maupun untuk membuat program *scripting*. Python juga bahasa pemrograman yang yang dianggap paling banyak digunakan di dunia [7].

Bahasa *python* lebih mudah dipahami dikarenakan bahasa pemrograman ini lebih mendekati bahasa manusia dibandingkan bahasa pemrograman lain. Fitur yang terdapat pada *python* juga beragam seperti dapat dijalankan di hampir seluruh sistem operasi (*cross platform*), program atau *script* yang mudah dipindahkan (*portable*), dan masih banyak lagi.

2.2.3 Analisis Sentimen

Analisis sentimen, yang disebut juga dengan penambangan opini (*opinion mining*), merupakan cabang ilmu dari penambangan data yang bertujuan untuk memahami, menganalisis, mengekstrak, dan mengolah data berbentuk teks yang berupa opini terhadap entitas seperti produk, servis, organisasi, individu, dan topik tertentu [8].

2.2.4 Scraping Data

Scraping data adalah tahap pertama yang dilakukan untuk melakukan analisis sentimen dari opini pengguna Twitter. Teknik Scraping menggunakan cara mengambil data dari apa yang ditampilkan oleh website [9] . Pada tahap ini dilakukan penarikan data menggunakan *library* snsrape, karena *library* ini dapat menarik data yang tidak dapat dilakukan oleh API Twitter gratis yaitu lebih dari 7 hari kebelakang dan dapat menarik tweet lebih dari batas API Twitter [10].

2.2.5 Preprocessing

Tujuan dilakukannya preprocessing dokumen adalah untuk menghilangkan suatu hal yang dapat mengganggu jalannya analisis, menyeragamkan bentuk kata dan mengurangi volume kata. Pada tahap preprocessing ini dilakukan proses *Case Folding*, *Cleansing*, *Tokenizing*, *Normalization*, dan *Stopword Removing*.

2.2.6 TF-IDF

TF-IDF merupakan suatu algoritma yang dapat menghasilkan informasi tentang seberapa sering kata tersebut muncul di dalam dataset tersebut dan dimunculkan dalam bentuk berat per kata. Untuk menentukan berat dari per kata tersebut algoritma ini menggunakan beberapa komponen yang sesuai dengan namanya yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) [11].

Term Frequency (TF) adalah seberapa sering kata tersebut muncul dalam dataset sedangkan *Inverse Document Frequency* (IDF) adalah pengurangan dari berat setiap kata yang muncul pada dataset.

Rumus dari *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) adalah sebagai berikut [12]:

$$TF - IDF_{t,d} = TF_{t,d} \times IDF_t$$

Keterangan :

$TF_{t,d}$: Frekuensi kata terhadap kata t di dokumen d

IDF_t : Kejarangan frekuensi kata t pada dokumen

$$IDF_t = \ln \left(\frac{1 + N}{1 + df_t} \right) + 1$$

Keterangan :

N : Jumlah dokumen.

df_t : Jumlah dokumen yang terdapat kata t.

Sedangkan TF-IDF yang terdapat pada *library* scikit-learn dinormalisasikan menggunakan rumus *equlidian*. Rumusnya adalah sebagai berikut [12]:

$$v_{norm} = \frac{v}{\sqrt{v_1^2 + v_2^2 + v_3^2 + v_4^2 + \dots + v_n^2}}$$

Keterangan :

v_{norm} : Vektor TF-IDF setelah normalisasi.

v : Vektor TF-IDF sebelum normalisasi.

$v_1^2 + v_2^2 + v_3^2 + v_4^2 + \dots + v_n^2$: Vektor yang terdapat pada dokumen yang sama.

Contoh :

Terdapat kalimat “Saya sedang belajar hitung - hitung tf idf. Mari belajar hitung hitung bersama tf idf.” tentukan nilai TF dan TF-IDF !

Jawaban ;

Nilai TF (*Term Frequency*) :

Tabel TF (<i>Term Frequency</i>) :		
Term (t)	D1 (Dokumen 1)	D2 (Dokumen 2)
Saya	1	0
sedang	1	0
belajar	1	0
hitung	0,5	0,5
tf	0,5	0,5
idf	0,5	0,5
Mari	0	1
bersama	0	1

Nilai DF (*Document Frequency*) :

Nilai DF (<i>Document Frequency</i>) :	
Term (t)	DF
Saya	1
sedang	1
belajar	1
hitung	4
tf	2
idf	2
Mari	1
bersama	1

Menghitung IDF :

$$IDF_t = \ln \left(\frac{1 + N}{1 + df_t} \right) + 1$$

Term (t)	IDF
Saya	1,40546511
sedang	1,40546511
belajar	1,40546511
hitung	0,48917438
tf	1
idf	1
Mari	1,40546511
bersama	1,40546511

Menghitung TF-IDF :

$$TF - IDF_{t,d} = TF_{t,d} \times IDF_t$$

Term (t)	TF-IDF	
	D1	D2
Saya	1,405465108	0
sedang	1,405465108	0
belajar	1,405465108	0
hitung	0,244587188	0,244587
tf	0,5	0,5
idf	0,5	0,5
Mari	0	1,405465
bersama	0	1,405465

Menghitung TF-IDF yang dinormalisasi :

$$v_{norm} = \frac{v}{\sqrt{v_1^2 + v_2^2 + v_3^2 + v_4^2 + \dots + v_n^2}}$$

TF-IDF dengan Normalisasi		
Term (t)	D1 (Dokumen 1)	D2 (Dokumen 2)
Saya	0,551871088	0
sedang	0,551871088	0
belajar	0,551871088	0
hitung	0,096039807	0,115165388
tf	0,196330412	0,235428088
idf	0,196330412	0,235428088
Mari	0	0,661771926
bersama	0	0,661771926

2.2.7 *Lexicon Based Features*

Lexicon Based Features merupakan fitur kata yang terdapat sentiment positif dan negatif berdasarkan kamus (*lexicon*). *Lexicon* merupakan kumpulan kata pada sentimen yang telah diketahui dan dihimpun dalam bentuk dataset [13].

Untuk melakukan proses pembobotan menggunakan fitur ini, dibutuhkan kamus (*lexicon*) yang mengandung kata yang sudah diberi sentimen.

2.2.8 *Valence Aware Dictionary and sEntiment Reasoner (VADER)*

Vader adalah suatu metode atau alat dalam melakukan sentimen analisis berbasis *lexicon* atau aturan yang sudah dibuat mendekati sentimen pada sosial media.[14]

Metode atau alat ini juga memperhatikan urutan kata maupun negasi yang terdapat pada setiap kalimat.[15]

2.2.9 *Support Vector Machine (SVM)*

Support Vector Machine adalah metode klasifikasi yang menggunakan cara mengklasifikasikan secara linear dengan menemukan *hyperlane* yang terbaik yang berfungsi sebagai pemisah antara 2 kelas. Prinsip dasarnya dilakukan

pengklasifikasian secara linier lalu dikembangkan sampai dapat dipakai pada permasalahan non linier dengan memasukkan konsep kernel trick pada ruang kerja berdimensi tinggi [16].

Alur kerja Support Vector Machine sebagai berikut :

1. Memetakan data
2. Meminimalisir nilai margin

Dengan rumus :

$$\frac{1}{2} \|w\|^2 - \frac{1}{2} (w_1^2 + w_2^2)$$

Dengan syarat :

$$y_i(X_1 \cdot w + b) - 1 \geq 0, i = 1, 2, 3, 4, \dots, n$$

$$y_i(X_1 \cdot W_1 + X_2 \cdot W_2 + b) \geq 1$$

3. Mencari persamaan *hyperlane*
4. Memetakan *hyperlane*
5. Melakukan pengujian terhadap data
6. Melakukan klasifikasi

Contoh :

Terdapat data seperti berikut (4 titik dari 2 kelas yang berbeda) :

X ₁	X ₂	Kelas(y)
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Hitunglah persamaan *hyperplane* data tersebut

Jawaban :

$$y_i(X_1 \cdot w + b) - 1 \geq 0, i = 1, 2, 3, 4, \dots, n$$

$$y_i(X_1.W_1 + X_2.W_2 + b) \geq 1$$

Sehingga : $(W_1 + W_2 + b) \geq 1$ untuk $y_1 = 1, X_1=1, X_2=1$

$$(-W_1 + W_2 - b) \geq 1 \text{ untuk } y_2 = -1, X_1=1, X_2=-1$$

$$(W_1 - W_2 - b) \geq 1 \text{ untuk } y_3 = -1, X_1=-1, X_2=1$$

$$(W_1 + W_2 - b) \geq 1 \text{ untuk } y_1 = -1, X_1=-1, X_2=-1$$

Lalu setelah itu dilakukan menjumlahkan / mengurangi masing persamaan yaitu persamaan 1 dan 2, 2 dan 3, serta 1 dan 3 sehingga menghasilkan nilai sebagai berikut :

$$W_1 = 1$$

$$W_2 = 1$$

$$b = -1$$

Sehingga dapat dicari persamaan dari *hyperplane*-nya

$$W_1.X_1 + W_2.X_2 + b = 0$$

$$1.X_1 + 1.X_2 - 1 = 0$$

$$X_1 + X_2 - 1 = 0$$

$$X_2 = 1 - X_1$$

2.2.10 K-Nearest Neighbour (KNN)

K-Nearest Neighbour adalah sebuah algoritma untuk klasifikasi yang menggunakan cara mengukur tingkat kemiripan antar data yang bertetangga (*cosine similarity*) atau mengukur jarak *euclidean* dari data latih (*training data*) dengan data uji (*test data*) [17].

Alur dari K-Nearest Neighbour sebagai berikut :

1. Menghitung jarak kesemua data *training* menggunakan *cosine similarity* atau *euclidean distance*.

2. Mengurutkan berdasarkan jarak terdekat dan ambil sejumlah K.
3. Mengambil K yang terbaik.
4. Mengambil label K terbaik sebelumnya yang paling banyak.

Contoh soal menggunakan *euclidean distance* :

Diberikan data sebagai berikut :

Tinggi	Berat	Jenis Kelamin
155	50	Perempuan
175	63	Laki - Laki
160	55	Perempuan
177	68	Laki - Laki
163	52	Perempuan
176	78	Laki - Laki

Tentukan jenis kelamin jika tinggi 172 dan berat 58 dengan K=3!

Jawaban :

$$\text{Data 1} = \sqrt{(155 - 172)^2 + (50 - 58)^2} = 18,78829423$$

$$\text{Data 2} = \sqrt{(175 - 172)^2 + (63 - 58)^2} = 5,830951895$$

$$\text{Data 3} = \sqrt{(165 - 172)^2 + (55 - 58)^2} = 12,36931688$$

$$\text{Data 4} = \sqrt{(177 - 172)^2 + (68 - 58)^2} = 11,18033989$$

$$\text{Data 5} = \sqrt{(163 - 172)^2 + (52 - 58)^2} = 10,81665383$$

$$\text{Data 6} = \sqrt{(176 - 172)^2 + (78 - 58)^2} = 20,39607805$$

Jika K=3 maka data yang diambil :

1. Data 6 (Laki - Laki)
2. Data 1 (Perempuan)
3. Data 3 (Laki -Laki)

Dan dapat disimpulkan jika K=3 maka prediksinya adalah Laki-Laki.

Persamaan dari cosine similarity ditunjukkan pada gambar dibawah ini.

$$\text{CosSim}(q, d_j) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2} \cdot \sqrt{\sum_{i=1}^t w_{iq}^2}}$$

Keterangan :

$\text{CosSim}(q, d_j)$: Nilai kemiripan antara dokumen uji (q) dengan dokumen latih ke j (d_j)

t : Jumlah term (kata)

d : Dokumen

q : Kata kunci (*query*)

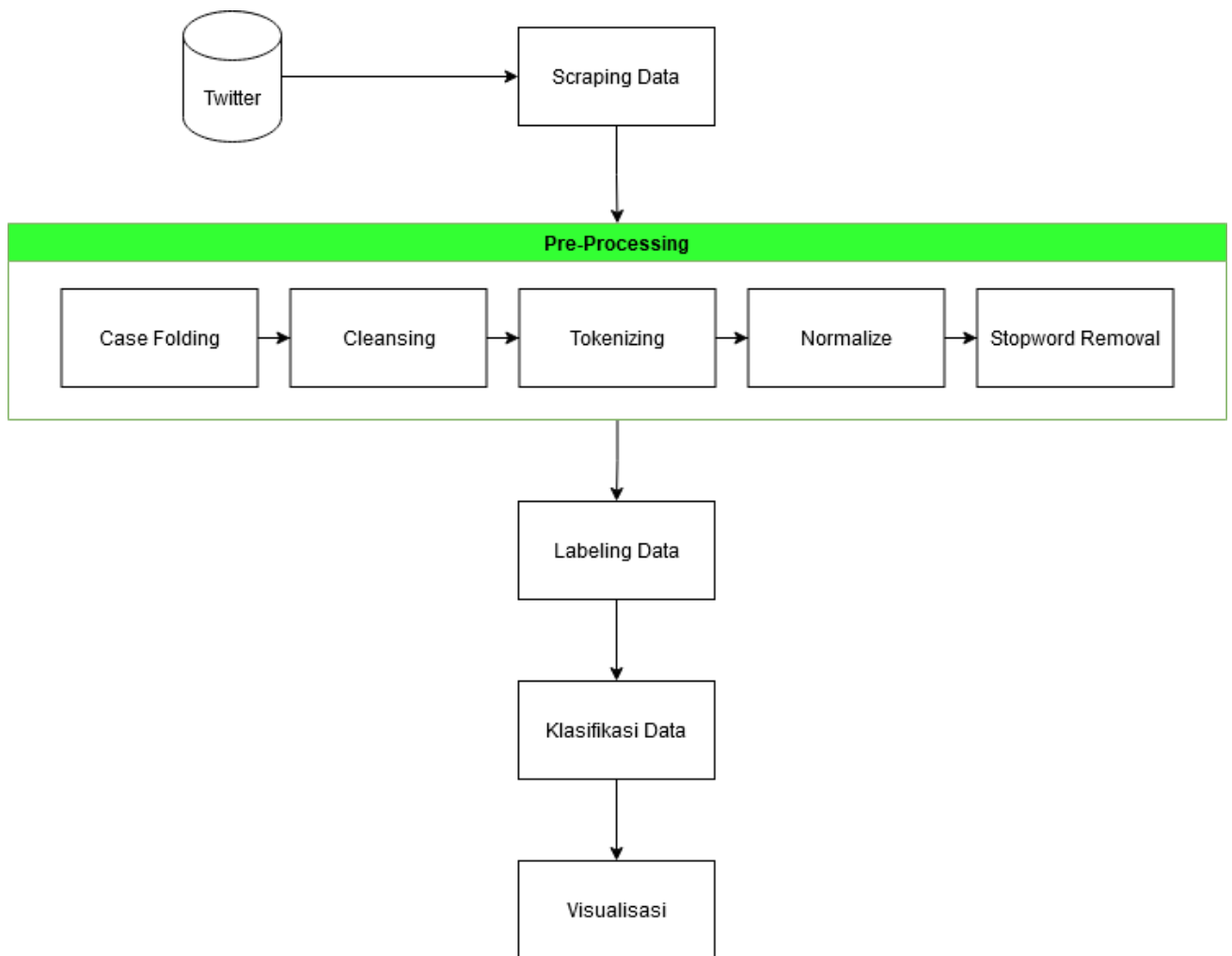
w_{ij} : Bobot term (kata) ke i pada dok. latih j

w_{iq} : Bobot term (kata) ke i pada dok. uji q

BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian

Untuk Menyusun tugas akhir, penulis menggunakan *flowchart* sebagai berikut :



Gambar III-1 *Flowchart* dari Penelitian

Berikut langkah-langkah penyelesaian penelitian ini yaitu:

3.1.1 Pengumpulan Data

Pengumpulan data dari salah satu media social terbesar yaitu Twitter menggunakan teknik *Scraping* data yang menggunakan library. Data yang dikumpulkan berupa tweet berbahasa Indonesia dengan kata kunci “uu tpks” dalam rentang waktu 12 April 2022 hingga 31 April 2022 dan tidak disertakan posting *retweet*.

3.1.2 Pengolahan Data

Setelah melakukan pengumpulan data sebelum dianalisis perlu dilakukan proses awal atau dikenal dengan istilah Preprocessing. Proses ini akan mengolah data awal yang masih tidak beraturan untuk dijadikan data teratur yang dapat diterapkan pada proses selanjutnya. Preprocessing yang dilakukan terdiri dari Case Folding, Cleansing, Tokenizing, Normalization, dan Stopword Removing.

3.1.2.1. Case Folding

Case Folding adalah langkah untuk melakukan perubahan huruf besar atau huruf kapital (*uppercase*) yang terdapat pada teks menjadi huruf kecil (*lowercase*).

3.1.2.2. Cleansing

Cleansing adalah langkah membersihkan data dari hal – hal yang tidak perlu seperti URL, *hashtag*, tanda baca, angka dan lain sebagainya.

3.1.2.3. Tokenizing

Tokenizing adalah melakukan perubahan dari suatu kata pada kalimat yang dipisahkan oleh separator (*space*) menjadi sebuah token.

3.1.2.4. Normalization

Normalization adalah suatu proses dimana kata yang tidak baku atau singkat dirubah menjadi kata baku yang benar.

3.1.2.5. Stopword Removing

Stopword Removing adalah proses dimana kata penghubung seperti yang, di, ke, dari yang tidak diperlukan pada proses analisis dibuang.

3.1.3 Labeling Data

Setelah data dibersihkan lalu dilakukan pelabelan pada data. Labeling pada data dilakukan secara otomatis menggunakan kamus yang sudah berisi bobot sentimen (*lexicon*) dan dihitung total dari sentimen berdasarkan jumlah bobot dari seluruh kata pada setiap data.

3.1.4 Pembobotan kata (TF-IDF)

Setelah di berikan label selanjutnya dilakukan pembobotan kata. Pembobotan kata dilakukan dengan menggunakan *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF).

3.1.5 Mengklasifikasikan Data

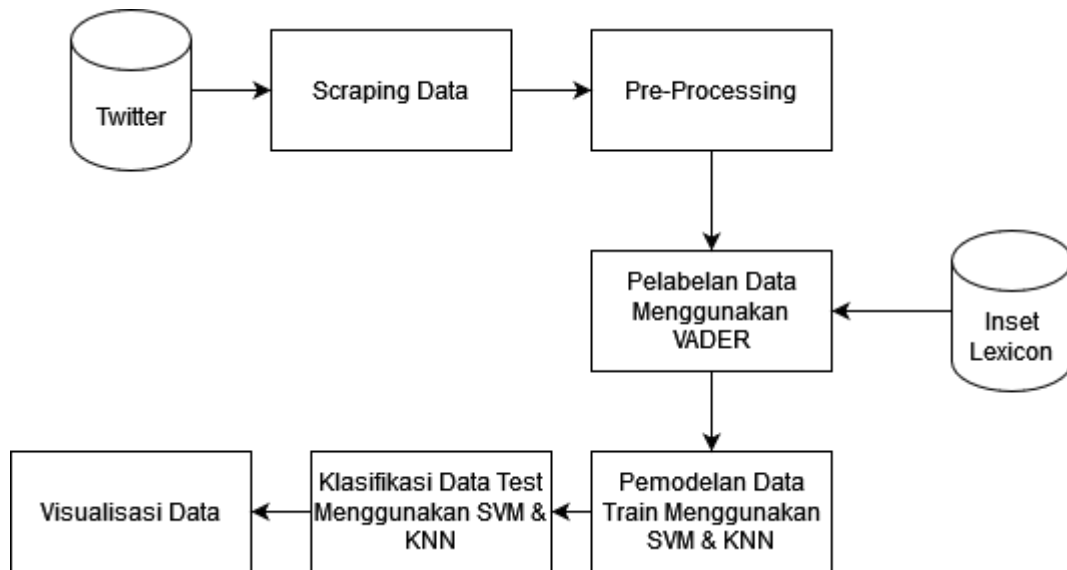
Proses ini bertujuan untuk mengolah data menjadi opini positif dan opini negatif. Ada banyak metode untuk mengklasifikasikan data, salah satunya adalah *Support Vector Machine* dan *K-Nearest Neighbour* . Merupakan salah satu metode untuk mengklasifikasikan data dan regresi. Pada penelitian ini, penulis menggunakan metode *Support Vector Machine* dan *K-Nearest Neighbour* untuk mengklasifikasikan data.

3.1.6 Visualisasi

Pada proses ini akan dilakukan visualisasi terhadap data yang dihasilkan dari proses klasifikasi. Tujuan dari proses ini untuk mempermudah membaca maksud dan informasi dari hasil analisis.

3.2 Metode Pelabelan Data

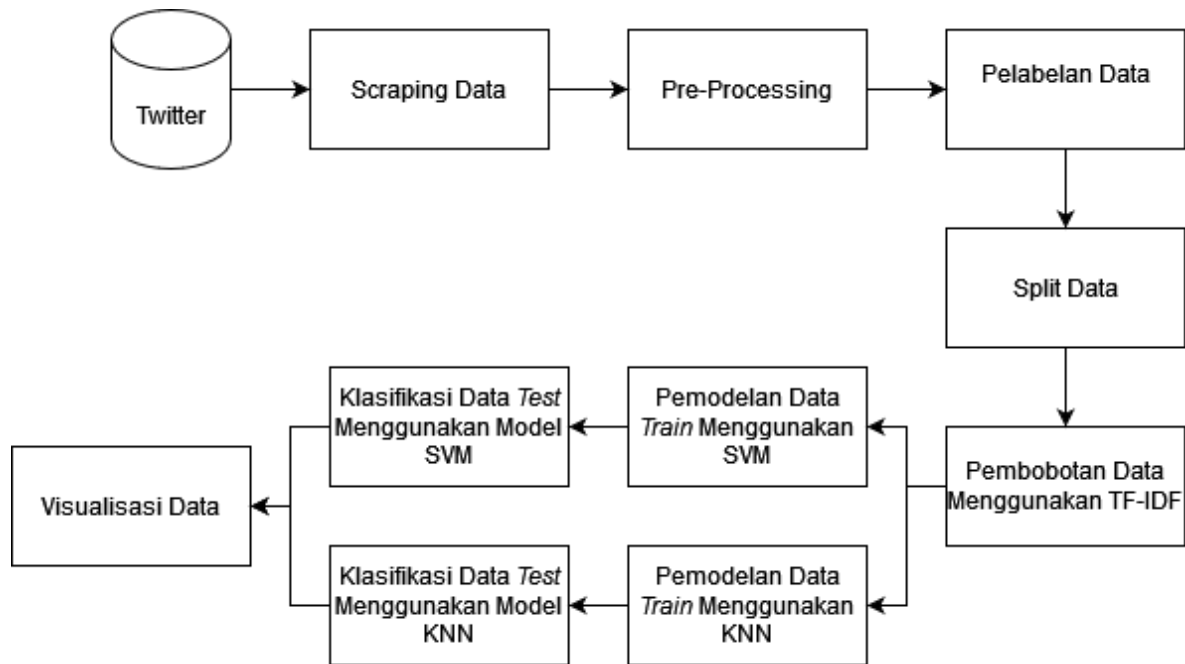
Berikut adalah *flowchart* dari proses Pelabelan Data :



Gambar III-2 *Flowchart* dari Pelabelan Data

Gambar diatas adalah alur dari proses Pelabelan Data yang digunakan pada penelitian ini. Dimulai dengan mengambil data dari Twitter menggunakan cara *scraping* yang dibantu oleh library, lalu dilanjutkan dengan pre-processing data yang terdiri dari *Case Folding*, *Cleansing Data*, *Tokenizing*, *Normalize*, dan *Stopword Removal*. Setelah data bersih lalu dilanjutkan ke tahap pelabelan data dengan *Lexicon – Based* dengan menggunakan Vader yang lexiconnya dirubah terlebih dahulu yang sebelumnya berbahasa inggris dirubah menjadi berbahasa Indonesia dengan menggunakan lexicon pada penelitian InSet (Indonesian Sentiment). Hasil dari pelabelan data yaitu adanya sentiment positif maupun negative pada data tweet. Hasil dari pelabelan ini selanjutnya akan diklasifikasikan menggunakan model yang dibuat dengan SVM & KNN.

3.3 Metode Klasifikasi



Gambar III-3 *Flowchart* dari Klasifikasi

Gambar diatas adalah alur dari proses klasifikasi data yang digunakan pada penelitian ini. Dimulai dengan mengambil data dari Twitter menggunakan cara *scraping* yang dibantu oleh library, lalu dilanjutkan dengan pre-processing data yang terdiri dari *Case Folding*, *Cleansing Data*, *Tokenizing*, *Normalize*, dan *Stopword Removal*. Setelah data bersih lalu dilanjutkan ke tahap pelabelan data dengan *Lexicon – Based* dengan menggunakan Vader. Lalu hasil dari pelabelan ini selanjutnya dilakukan *splitting* data *train* dan *test* setelah data ter-*split* dilakukan pembobotan kata di setiap kalimat dan setelah itu data akan diklasifikasikan menggunakan SVM & KNN dan hasil dari klasifikasinya divisualisasikan dalam bentuk akurasi.

BAB IV

PEMBAHASAN

4.1 Scraping data

Pada tahap ini data tweet yang terdapat pada twitter diambil dengan menggunakan Teknik *Scrapping*. Teknik *Scraping* untuk mendapatkan data dari twitter menggunakan *library Snsrape*. *Snsrape* adalah suatu *library* yang berisi beberapa fungsi yang dapat digunakan untuk menarik data dari sosial media seperti facebook, Instagram, twitter, dan seterusnya.

```
import snsrape.modules.twitter as sntwitter
import pandas as pd
import csv
from tqdm.notebook import tqdm
from pandas_profiling import ProfileReport
```

Gambar IV-1 Library yang Digunakan pada Scraping Data

Kelima *library* tersebut digunakan untuk menarik data dari twitter (*snsrape*), menjadikan data ke *DataFrame* (*pandas*), menghilangkan duplikat dan menambahkan petik (*pandas*), mengekspor data menjadi csv (*pandas*), serta profiling sederhana dari data tersebut(*pandas_profiling*).

Setelah mengimport kelima *library* tersebut, kemudian membuat *query* tweet yang akan di *search* di twitter. *Query* terdiri dari tanggal awal dan akhir, kata kunci, dan Bahasa yang digunakan dalam bentuk *unicode*. Dalam penelitian ini *query* yang digunakan adalah “UU TPKS” untuk kata kunci, 12 – 24 April 2022 untuk tanggal awal dan akhir, serta Indonesia (id) untuk Bahasa seperti gambar berikut :

```

tweets_temp = []
# Creating list to append tweet data
search_words = "'uu tpks'"
search = search_words + " -filter:retweets"
date_since = '2022-04-12'
date_until = '2022-04-24'
lang = 'id'
search = search + ' since:' + date_since + ' until:' + date_until + ' lang:' + lang
tweets_count = 20000

```

Gambar IV-2 Query Pencarian Tweet

Dan untuk cara menjalankan tahap ini dan memasukkannya ke dalam dataframe adalah sebagai berikut :

```

for i, tweet in enumerate(tqdm(sntwitter.TwitterSearchScraper(search).get_items(
    , desc="Proses Scrapping Data", total=tweets_count)):
    if i>tweets_count:
        break
    tweets_temp.append([tweet.date, tweet.id, tweet.content, tweet.user.username])

# Creating a dataframe from the tweets list above
df_tweets = pd.DataFrame(tweets_temp, columns=['Datetime', 'Tweet Id', 'Tweet', 'Username'])

```

Gambar IV-3 Kode untuk Scrapping Data Twitter

Setelah data terkumpul, dilakukan filtrasi awal yaitu menghapus data yang duplikat. Tahap ini dilakukan agar data yang terhimpun tidak ada yang berulang – ulang, cara menjalankan tahap ini adalah sebagai berikut :

```

bruto = int(len(df_tweets))
df_tweets.drop_duplicates(subset=['Tweet'])
print("Dataset dibuang (Karena duplikat) : "+(str(bruto-int(len(df_tweets))))+" data")
print("Dataset masuk : "+str(len(df_tweets))+" data")
df_tweets.to_csv('data/tweets.csv',index=False,quoting=csv.QUOTE_ALL)

```

Gambar IV-4 Kode untuk Drop Data Duplikat

Serta data sebelum dan sesudah dilakukannya filtrasi awal yang berupa penghapusan data duplikat adalah sebagai berikut :

Sebelum Penghapusan Data Duplikat	Setelah Penghapusan Data Duplikat
Dewan Perwakilan Rakyat akhirnya mengesahkan Undang-Undang Tindak Pidana Kekerasan Seksual (UU TPKS) pada Selasa, 12 April lalu	Dewan Perwakilan Rakyat akhirnya mengesahkan Undang-Undang Tindak Pidana Kekerasan Seksual (UU TPKS) pada Selasa, 12 April lalu
Dewan Perwakilan Rakyat akhirnya mengesahkan Undang-Undang Tindak Pidana Kekerasan Seksual (UU TPKS) pada Selasa, 12 April lalu	

Penarikan ini menghasilkan 15.632 data tweet yang berjarak di tanggal 12 sampai dengan 24 April 2022. Tanggal tersebut diambil dikarenakan pada tanggal tersebut saat disahkannya RUU TPKS menjadi UU TPKS.

4.2 *Pre – Processing*

Pada tahap ini dilakukannya pembersihan dan pengubahan terhadap data yang telah dihimpun sebelumnya agar tidak terdapat data yang dapat mengganggu jalannya analisis dan untuk menjadikan data dapat diproses ke tahap selanjutnya. Library yang digunakan pada tahapan ini adalah sebagai berikut :

```
import pandas as pd
import numpy as np
import string, re, nltk
from nltk.tokenize import word_tokenize
from nltk.tokenize.treebank import TreebankWordDetokenizer
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from tqdm.notebook import tqdm
import matplotlib.pyplot as plt
```

Gambar IV-5 Library yang Digunakan Pada *Pre-Processing*

Library tersebut digunakan untuk mengimport data dari tahap sebelumnya (pandas), melakukan cleansing (string, re), melakukan *tokenizing* dan *normalize* (nltk) serta stopwords removal (nltk).

Pada tahap pre-processing terdapat 5 tahap yang digunakan untuk menjadikan data bersih dan siap untuk digunakan untuk tahap selanjutnya, tahap tersebut yaitu :

4.2.1 *Case Folding*

Tahap ini bertujuan untuk merubah huruf kapital menjadi huruf kecil agar datanya sama rata. Cara menjalankan tahap ini adalah sebagai berikut :

```
df['tweet'] = df['tweet'].str.lower()
print('Hasil Case Folding : \n')
print(df['tweet'].head(10))
```

Gambar IV-6 Kode untuk *Case Folding*

Serta data sebelum dan sesudah dilakukannya *case folding* adalah sebagai berikut :

Sebelum "Case Folding"	Setelah "Case Folding"
Lihat tanggal chatnya, kalau setelah April udah bisa dijerat UU TPKS	lihat tanggal chatnya, kalau setelah april udah bisa dijerat uu tpks

Tabel IV-1 Hasil dari Case Folding

4.2.2 Cleansing

Tahap ini bertujuan untuk menghilangkan data *hashtag*, *mention*, tanda baca, angka, url, *space* yang tidak berguna, serta data yang bukan ASCII seperti emotikon, data berbahasa china, dan seterusnya. Cara menjalankan tahap ini adalah sebagai berikut :

```
def remove_tweet_special(text):
    # remove tab, new line, and back slice
    text = text.replace('\t', " ").replace('\n', " ").replace('\u', " ").replace('\\', "")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9+])|(\w+:\/\/\S+)", " ", text).split())
    # remove url (menghapus link)
    text = re.sub(r'\w+:\/\/{2}[\d\w-]+(\.[\d\w-]+)?(?:\/[\^\/])?', '', text)
    # remove incomplete URL
    return text.replace("http://", " ").replace("https://", " ")

df['tweet'] = df['tweet'].apply(remove_tweet_special)
```

Gambar IV-7 Kode untuk Cleansing Tahap 1

```

#remove number (menghapus angka)
def remove_number(text):
    return re.sub(r"\d+", "", text)

df['tweet'] = df['tweet'].apply(remove_number)

#remove punctuation (menghapus tanda baca)
def remove_punctuation(text):
    return text.translate(str.maketrans("", "", string.punctuation))

df['tweet'] = df['tweet'].apply(remove_punctuation)

# remove single char (menghapus 1 karakter)
def remove_singl_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)

df['tweet'] = df['tweet'].apply(remove_singl_char)

```

Gambar IV-8 Kode untuk *Cleansing* Tahap 2

```

#remove whitespace leading & trailing (menghapus spasi awal dan akhir)
def remove_whitespace_LT(text):
    return text.strip()

df['tweet'] = df['tweet'].apply(remove_whitespace_LT)

#remove multiple whitespace into single whitespace
def remove_whitespace_multiple(text):
    return re.sub('\s+', ' ', text)

df['tweet'] = df['tweet'].apply(remove_whitespace_multiple)

```

Gambar IV-9 Kode untuk *Cleansing* Tahap 3

Serta data sebelum dan sesudah dilakukannya *Cleansing* adalah sebagai berikut :

Sebelum ” <i>Cleansing</i> ”	Setelah ” <i>Cleansing</i> ”
lihat tanggal chatnya, kalau setelah april udah bisa dijerat uu tpks 😊 https://t.co/c89NcYzjQv	lihat tanggal chatnya, kalau setelah april udah bisa dijerat uu tpks

UUTPKS diterapkan keras mulai dari pemrentah dan @DPR_RI, setuju? @KemensetnegRI https://t.co/isAu9nBZpx	UUTPKS diterapkan keras mulai dari pemrentah dan setuju
---	--

Tabel IV-2 Hasil dari *Cleansing*

4.2.3 Tokenizing

Di tahap ini data akan dipisahkan berdasarkan separator (space) menjadi token – token (kata di setiap kalimat pada data). Langkah ini berfungsi untuk menjadikan data kompartibel dengan tahap selanjutnya seperti *normalize*, *stopword* removal, dan sebagainya. Cara menjalankan tahap ini adalah sebagai berikut :

```

nltk.download('punkt')
def word_tokenize_wrapper(text):
    return word_tokenize(text)

df['tweet'] = df['tweet'].apply(word_tokenize_wrapper)
df['tweet']

```

Gambar IV-10 Kode untuk *Tokenizing*

Serta data sebelum dan sesudah dilakukannya *tokenizing* adalah sebagai berikut :

Sebelum " <i>Tokenizing</i> "	Setelah " <i>Tokenizing</i> "
lihat tanggal chatnya kalau setelah april udah bisa dijerat uu tpks	[lihat, tanggal, chatnya, kalau, setelah, april, udah, bisa, dijerat, uu, tpks]

Tabel IV-3 Hasil dari *Tokenizing*

4.2.4 Normalize

Pada tahap ini data yang berbentuk tidak baku diubah menjadi kata baku. Langkah ini berfungsi untuk mencegah terjadinya terdapat kata – kata yang diluar *lexicon* (*out of vocabulary*) dikarenakan sebagian besar *lexicon* adalah kata baku. Cara menjalankan tahap ini adalah sebagai berikut :

```
! pip install openpyxl
normalizad_word = pd.read_excel("data/normalisasi.xlsx",
                                engine='openpyxl')

normalizad_word_dict = {}

for index, row in normalizad_word.iterrows():
    if row[0] not in normalizad_word_dict:
        normalizad_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalizad_word_dict[term] if term
            in normalizad_word_dict else term for term in document]

df['tweet'] = df['tweet'].apply(normalized_term)

df.head(10)
```

Gambar IV-11 Kode untuk *Normalize*

Serta data sebelum dan sesudah dilakukannya *normalize* adalah sebagai berikut :

Sebelum " <i>Normalize</i> "	Setelah " <i>Normalize</i> "
kagak ada yg ribut nyuruh seragaman baju nasional kebaya uu tpks sah vaksin serviks bakal jd vaksin wajib	tidak ada yang ribut nyuruh seragaman baju nasional kebaya uu tpks sah vaksin serviks bakal jadi vaksin wajib

Tabel IV-4 Hasil dari *Normalize*

4.2.5 Stopword removal

Tahapan ini bertujuan untuk menghilangkan kata – kata yang tidak diperlukan pada data yang dapat mengganggu jalannya analisis. File daftar stopwords yang dipakai didapatkan dari data library spaCy yang digabung dengan suatu daftar stopwords di github, linknya yaitu : <https://raw.githubusercontent.com/Muhammad-Yunus/Neural-Network/master/3.%20Convolutional%20Neural%20Network/Text%20Preprocessing/stopwords.txt>. Cara menjalankan tahap ini adalah sebagai berikut :

```
nlTK.download('stopwords')
list_stopwords = stopwords.words('indonesian')
list_stopwords.extend(["yg", "dg", "rt", "dgn", "ny", "d", 'klo',
                        'kalo', 'amp', 'biar', 'bikin', 'bilang',
                        'gak', 'ga', 'krn', 'nya', 'nih', 'sih',
                        'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',
                        'jd', 'jgn', 'sdh', 'aja', 'n', 't',
                        'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',
                        '&amp;', 'yah', 'uu', 'tpks', 'uu tpks', 'ruu', 'uutpks' ])
txt_stopword = pd.read_csv("data/stopwords.txt", names= ["stopwords"], header = None)
list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))
list_stopwords = set(list_stopwords)
def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]
df['tweet'] = df['tweet'].apply(stopwords_removal)
df
```

Gambar IV-12 Kode untuk *Stopword Removal*

Serta data sebelum dan sesudah dilakukannya *Stopword removal* adalah sebagai berikut :

Sebelum ” <i>Stopword Removal</i> ”	Setelah ” <i>Stopword Removal</i> ”
lihat tanggal chatnya kalau setelah april sudah bisa dijerat uu tpks	tanggal chatnya april dijerat uu tpks

Tabel IV-5 Hasil dari *Stopword Removal*

4.3 Pelabelan Data

Setelah data dibersihkan melalui tahap *pre-processing*, selanjutnya masuk ke tahap analisis sentiment. Tahap ini bertujuan untuk memberikan label terhadap data menjadi sebuah sentiment (positif & negatif). Pelabelan yang digunakan menggunakan metode berbasis *Lexicon – based* dengan library *VADER* (*Valence Aware Dictionary and sEntiment Reasoner*).

Langkah – Langkah yang terdapat pada tahap ini yaitu data dari tahap sebelumnya (*Pre-Processing*) yang masih berbentuk token terlebih dahulu di un-tokenize agar dapat diproses oleh library setelah itu kamus bawaan dari library *VADER* dibersihkan terlebih dahulu dan dimasukkan kamus *Inset* lalu dilakukan pelabelan terhadap setiap baris berdasarkan nilai polaritas yang dihasilkan oleh library *VADER* setelah selesai dilakukan pelabelan lalu divisualisasikan menggunakan Wordcloud dan Pie Chart.

Daftar library yang digunakan pada tahap ini adalah sebagai berikut :

```
import pandas as pd
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.tokenize.treebank import TreebankWordDetokenizer
from nltk.tokenize import word_tokenize
import json
import re
from wordcloud import WordCloud
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
```

Gambar IV-13 Library yang Digunakan pada Pelabelan Data ke 1

```
import nltk
nltk.download('vader_lexicon')
```

Gambar IV-14 Library yang Digunakan pada Pelabelan Data ke 2

Library tersebut digunakan untuk mengimport data dari tahap sebelumnya (pandas), melakukan *detokenize* (nlTK), mengubah *lexicon* (nlTK, json, replib) menentukan skor polaritas (nlTK), menentukan sentimen, serta visualisasi sementara dari data yang telah diberikan sentimen (wordcloud, PIL, matplotlib).

Sebelum masuk ke tahap selanjutnya data yang sebelumnya di ekspor ke file csv di tahap *Pre – Processing* masih berbentuk token oleh karena itu harus data terlebih dahulu harus dirubah ke format yang sesuai. Cara mengubah file sesuai format yang akan dijalankan di tahap selanjutnya adalah sebagai berikut ini :

```
temp_detokenize = []

def detokenize(text):
    text1 = text.replace(' ', '').replace('[', '')
    arr = text1.replace('"', '').replace("\'", "").split(",")
    return(TreebankWordDetokenizer().detokenize(arr))

df['tweet'] = df['tweet'].astype('U').apply(detokenize)
```

Gambar IV-15 Kode untuk *Detokenize*

Pada tahap ini library *Vader* akan dirubah kamus bawaannya (Bahasa Inggris) menjadi kamus Bahasa Indonesia yang dibuat oleh para peneliti dari penelitian *Inset* (Indonesian Sentiment). Kamus ini berisi kata – kata Bahasa Indonesia yang setiap kata – katanya diberikan nilai dari -5 sampai dengan +5.

Isi dari kamus *Inset* (Indonesian Sentiment) adalah sebagai berikut (Sebelah Kiri Kamus Positif dan Sebelah Kanan Kamus Negatif) :

<code>{ "hai": 3,</code>	<code>{ "putus tali gantung": -2,</code>
<code>"merekam": 2,</code>	<code>"gelebah": -2,</code>
<code>"ekstensif": 3,</code>	<code>"gobar hati": -2,</code>
<code>"paripurna": 1,</code>	<code>"tersentuh (perasaan)": -1,</code>
<code>"detail": 2,</code>	<code>"isak": -5,</code>
<code>"pernik": 3,</code>	<code>"larat hati": -3,</code>
<code>"belas": 2,</code>	<code>"nelangsa": -3,</code>
<code>"welas": 4,</code>	<code>"remuk redam": -5,</code>
<code>"kabung": 1,</code>	<code>"tidak segan": -2,</code>
<code>"rahayu": 4,</code>	<code>"gemar": -1,</code>
<code>"maaf": 2,</code>	<code>"tak segan": -1,</code>
<code>"hello": 2,</code>	<code>"sesal": -4,</code>
<code>"promo": 3,</code>	<code>"pengen": -2,</code>
<code>"terimakasih": 5,</code>	<code>"penghayatan": -2,</code>

Gambar IV-16 Isi dari Kamus *Inset*

Cara menjalankan tahap ini adalah sebagai berikut :

```
# Memanfaatkan nltk VADER untuk menggunakan leksikon kustom
sia1A = SentimentIntensityAnalyzer()
sia1B = SentimentIntensityAnalyzer()
sia2 = SentimentIntensityAnalyzer()

# membersihkan leksikon VADER default
sia1A.lexicon.clear()
sia1B.lexicon.clear()
sia2.lexicon.clear()

# Membaca leksikon InSet
# Leksikon InSet lexicon dibagi menjadi dua, yakni polaritas negatif dan polaritas positif;
# kita akan menggunakan nilai compound saja untuk memberi label pada suatu kalimat
with open('data/lexicon/InSet/positive.json') as f:
    data1A = f.read()
with open('data/lexicon/InSet/negative.json') as f:
    data1B = f.read()
```

Gambar IV-17 Kode untuk Mengganti *Lexicon* pada Vader Tahap 1

```

# Membaca leksikon kata2 kasar
with open('data/lexicon/swear-words.json') as f:
    data2 = f.read()

# Mengubah leksikon sebagai dictionary
insetNeg = json.loads(data1A)
insetPos = json.loads(data1B)
senti = json.loads(data2)

# Update leksikon VADER yang sudah 'dimodifikasi'
sia1A.lexicon.update(insetNeg)
sia1B.lexicon.update(insetPos)
sia2.lexicon.update(senti)

print(reprlib.repr(sia1A.lexicon))
print(reprlib.repr(sia1B.lexicon))
print(reprlib.repr(sia2.lexicon))

```

Gambar IV-18 Kode untuk Mengganti *Lexicon* pada Vader Tahap 2

Lalu di tahap ini dilakukannya pemberian sentiment (*Sentiment Analysis*) pada setiap tweet dengan menggunakan kamus yang telah dimasukkan sebelumnya. Cara menjalankan tahap ini adalah sebagai berikut :

```

def is_positive_inset(tweet):
    """True if tweet has positive compound sentiment, False otherwise."""
    sia1a_pol = sia1A.polarity_scores(tweet)["compound"]
    sia1b_pol = sia1B.polarity_scores(tweet)["compound"]
    sia2_pol = sia2.polarity_scores(tweet)["compound"]
    return sia1a_pol + sia1b_pol + sia2_pol

```

Gambar IV-19 Kode untuk Mendapatkan Skor Sentimen (*Polarity Score*)

```

df2 = pd.DataFrame()
temp_df2 = []

df2['tweet'] = df['tweet'].copy()

for tweet in df2['tweet']:
    if is_positive_inset(tweet) > 0:
        label = "Positif"
    elif is_positive_inset(tweet) == 0:
        label = "Netral"
    else:
        label = "Negatif"
    temp_df2.append([label])

temp_df2 = pd.DataFrame(temp_df2, columns=['sentimen'])
df2['sentimen'] = temp_df2['sentimen'].copy()
df2.reset_index(drop=True, inplace=True)
df2

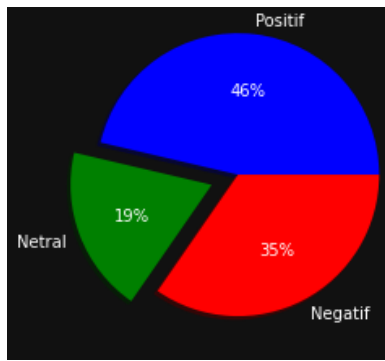
```

Gambar IV-20 Kode untuk Melabelkan Data Berdasarkan Skor Sentimen

Dan untuk hasil dari tahap ini adalah sebagai berikut :

	tweet	sentimen
1		
2	@convomf Kelakuan kayak gini bisa masuk ranah UU TPKS gak sih?	Netral
3	Puan mengatakan, pengesahan UU TPKS menjadi undang-undang merupakan bentuk hadiah bagi para perempuan di Indonesia menjelang Hari Kartini. https://t.co/0LvzlLLqh	Positif
4	Netizen mengapresiasi sikap Puan yang dinilai serius memperjuangkan UU TPKS disahkan menjadi undang – undang. https://t.co/9t7RVozPEK	Positif
5	@Kalpanax14 @tacenda_____ 1 "persetujuan untuk melakukan hubungan seksual" yang berada pada Bab V Pasal 16 UU TPKS. disitu jls bhw silma kedua belah pihak setuju, boleh tdk mengindahkan pernikahan diantaranya, ataupun karena asal dasar suka sama suka, itu sm sj melegalkn seks bebas!	Negatif
6	@ndagels Pergaulan tolol. Saatnya gunakan UU TPKS untuk mengadili perbuatan anak setan satu ini	Negatif
7	kenapa PKS ngotot nolak UU TPKS ini selain singkatan UU ini mengarah ke nama partainya, mungkin di PKS semuanya adalah tokoh agama jadi kalo terjerat UU ini hukumannya lebih berat. juga relasi kuasa disana kan cukup kuat karena kaderisasi yang ketat. mungkin. belum tentu bener	Negatif
8	"Sekarang saatnya UU TPKS diterjemahkan menjadi aturan-aturan pelaksanaan teknis agar semangat penyusunannya dapat segera dirasakan wujud nyatanya." --Puan Maharani https://t.co/z3l6yKKhHp	Positif
9	Bersyukur banget UU TPKS ini akhirnya disahkan. Merasa negara hadir untuk perempuan Indonesia 🙏🏻 https://t.co/hERt92yrLB	Positif

Gambar IV-21 Hasil dari Proses Pelabelan Data



Gambar IV-22 Diagram Pie dari Hasil Pelabelan



Gambar IV-24 Wordcloud dari Hasil Pelabelan (Semua Data)



Gambar IV-23 Wordcloud dari Hasil Pelabelan (Data Positif)



Gambar IV-25 Wordcloud dari Hasil Pelabelan (Data Negatif)

Dapat dilihat dari Gambar IV-21 bahwa setimen masyarakat mengenai UU TPKS sebagian besar adalah positif sebesar 46 %. Dan dari Gambar IV-22 dapat disimpulkan bahwa kata yang paling banyak muncul yaitu “Kekerasan” dan “Seksual” dan terdapat juga kata *collocation* seperti “disahkan puan” dan “ketua dpr”.

4.4 Pembobotan Kata

Di tahap ini dilakukan proses untuk mengubah kata – kata yang terkumpul menjadi *vektor* agar data dapat digunakan di proses selanjutnya

yaitu proses klasifikasi menggunakan SVM dan KNN. Library yang digunakan pada tahap ini adalah sebagai berikut :

```
from sklearn import model_selection
from sklearn.feature_extraction.text import TfidfVectorizer
```

Gambar IV-26 Library yang Digunakan pada Pembobotan Kata

Library ini berfungsi untuk melakukan *splitting* data (sklearn) dan pembobotan kata (sklearn).

Namun sebelum dilakukannya pembobotan kata dilakukan perubahan pada label netral menjadi positif dikarenakan pada tahap selanjutnya data yang disupport adalah berbentuk *binary*. Cara menjalankan tahap ini adalah sebagai berikut :

```
def merge_neutral(text):
    if text == "Netral":
        return "Positif"
    else:
        return "Negatif"

df['sentimen'] = df['sentimen'].apply(merge_neutral)
```

Gambar IV-27 Kode untuk Pengubahan Data Netral ke Positif

Lalu setelah itu dilakukan *splitting* pada data, untuk memisahkan data yang digunakan untuk *test* maupun untuk *train* dengan perbandingan 1:9. Cara menjalankan tahap ini adalah sebagai berikut :

```

train_X_df, test_X_df, train_Y_df, test_Y_df = model_selection.train_test_split(df['tweet']
                                                                              , df['sentimen']
                                                                              , test_size = 0.1
                                                                              , random_state = 42)

data_train = pd.DataFrame()
data_train['tweet'] = train_X_df
data_train['sentimen'] = train_Y_df

data_test = pd.DataFrame()
data_test['tweet'] = test_X_df
data_test['sentimen'] = test_Y_df

```

Gambar IV-28 Kode untuk *Splitting Data*

Setelah dilakukan *Splitting* terhadap data menjadi data *train* dan *test* dengan perbandingan 1:9 selanjutnya dilakukan pembobotan terhadap setiap kata. Tahap ini menggunakan metode TF-IDF. Cara menjalankan tahap ini adalah sebagai berikut :

```

datatfidf = TfidfVectorizer()
datatfidf.fit(df['tweet'].values.astype('U'))
train_X_datatfidf = datatfidf.transform(data_train['tweet'].values.astype('U'))
test_X_datatfidf = datatfidf.transform(data_test['tweet'].values.astype('U'))

```

Gambar IV-29 Kode untuk Pembobotan Kata Menggunakan TF-IDF

4.5 Klasifikasi Data Menggunakan SVM

Pada tahap ini data yang telah melalui proses pembobotan data, data selanjutnya masuk di tahap klasifikasi. Tahapan ini bertujuan untuk mengklasifikasikan data berdasarkan sentimen yang telah didapatkan sebelumnya. Pada tahap ini klasifikasi dilakukan menggunakan metode *Support Vector Machine* (SVM) Kernel Linear dan RBF dikarenakan kedua kernel tersebut adalah kernel yang memiliki akurasi yang cukup akurat. Cara yang digunakan pada tahap ini adalah sebagai berikut :

```
from sklearn.svm import SVC

model = SVC(kernel='linear')
model.fit(train_X_datatfidf, train_Y_df)
```

Gambar IV-30 Kode dari Klasifikasi SVM Kernel Linear ke 1

```
from sklearn.metrics import accuracy_score

predictionsSVM = model.predict(test_X_datatfidf)

test_prediction = pd.DataFrame()
test_prediction['tweet'] = test_X_df
test_prediction['sentimen'] = predictionsSVM

SVMaccuracy = accuracy_score(predictionsSVM, test_Y_df)*100
SVMaccuracy = round(SVMaccuracy,1)
```

Gambar IV-31 Kode dari Klasifikasi SVM Kernel Linear ke 2

```
from sklearn.svm import SVC

model_rbf = SVC(random_state=42, kernel='rbf')
model_rbf.fit(train_X_datatfidf, train_Y_df)
```

Gambar IV-32 Kode dari Klasifikasi SVM Kernel RBF ke 1

```
from sklearn.metrics import accuracy_score

predictionsSVM_rbf = model_rbf.predict(test_X_datatfidf)

test_prediction_rbf = pd.DataFrame()
test_prediction_rbf['tweet'] = test_X_df
test_prediction_rbf['sentimen'] = predictionsSVM_rbf

SVMaccuracy_rbf = accuracy_score(predictionsSVM_rbf, test_Y_df)*100
SVMaccuracy_rbf = round(SVMaccuracy_rbf,1)
```

Gambar IV-33 Kode dari Klasifikasi SVM Kernel RBF ke 2

Setelah dilakukan klasifikasi pada data *train* selanjutnya ditampilkan visualisasi berupa akurasi dari klasifikasi yang dilakukan pada proses sebelumnya. Cara yang digunakan untuk menampilkan visualisasi akurasi adalah sebagai berikut :

```
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

print("Support Vector Machine Acuracy:", accuracy_score(test_Y_df, predictionsSVM)*100, "%")
print("Support Vector Machine Precision:", precision_score(test_Y_df, predictionsSVM, average="binary",
                                                         pos_label="Negatif")*100, "%")
print("Support Vector Machine Recall:", recall_score(test_Y_df, predictionsSVM, average="binary",
                                                    pos_label="Negatif")*100, "%")
print("Support Vector Machine f1_score:", f1_score(test_Y_df, predictionsSVM, average="binary",
                                                  pos_label="Negatif")*100, "%")
print('=====\\n')
print (classification_report(test_Y_df, predictionsSVM))
```

Gambar IV-34 Kode dari Visualisasi Akurasi SVM Linear ke 1

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

print("Support Vector Machine Acuracy:", SVMaccuracy, "%")

conf_mat = confusion_matrix(test_Y_df, predictionsSVM)
class_label = ["Negative", "Positive"]
test = pd.DataFrame(conf_mat, index = class_label, columns = class_label)
sns.heatmap(test, annot = True, fmt = "d")
plt.title("Confusion Matrix for test data Support Vector Machine")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```

Gambar IV-35 Kode dari Visualisasi Akurasi SVM Linear ke 2

```
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

print("SVM RBF DF Acuracy:", accuracy_score(test_Y_df, predictionsSVM_rbf)*100, "%")
print("SVM RBF DF Precision:", precision_score(test_Y_df, predictionsSVM_rbf, average="binary",
                                              pos_label="Positif")*100, "%")
print("SVM RBF DF Recall:", recall_score(test_Y_df, predictionsSVM_rbf, average="binary",
                                         pos_label="Positif")*100, "%")
print("SVM RBF DF f1_score:", f1_score(test_Y_df, predictionsSVM_rbf, average="binary",
                                       pos_label="Positif")*100, "%")
print('=====\\n')
print (classification_report(test_Y_df, predictionsSVM_rbf))
```

Gambar IV-36 Kode dari Visualisasi Akurasi SVM RBF ke 1

```

from sklearn.metrics import confusion_matrix
import seaborn as sns

print("Support Vector Machine Acuracy:", SVMaccuracy_rbf, "%")

conf_mat = confusion_matrix(test_Y_df, predictionsSVM_rbf)
class_label = ["Negative", "Positive"]
test = pd.DataFrame(conf_mat, index = class_label, columns = class_label)
sns.heatmap(test, annot = True, fmt = "d")
plt.title("Confusion Matrix for test data Support Vector Machine")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()

```

Gambar IV-37 Kode dari Visualisasi Akurasi SVM RBF ke 2

Serta hasil dari visualisasi akurasi klasifikasi *support vector machine* (SVM) adalah sebagai berikut :

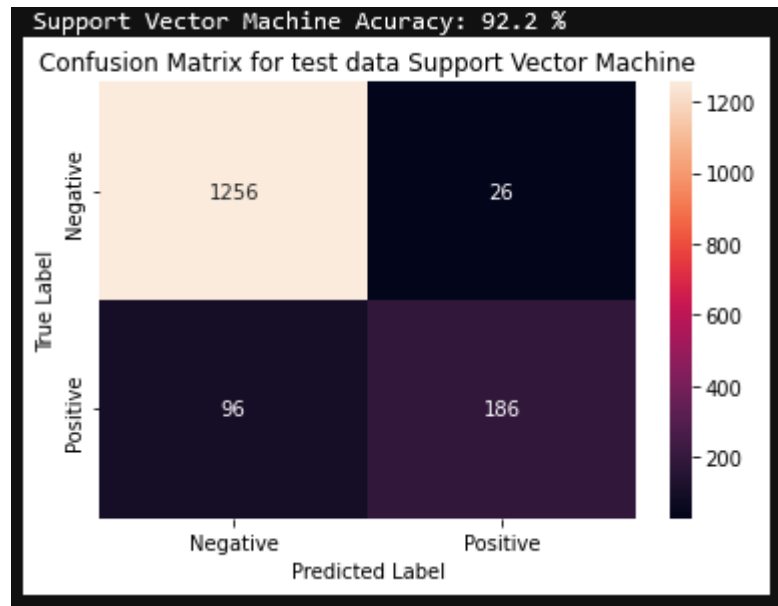
```

Support Vector Machine Acuracy: 92.1994884910486 %
Support Vector Machine Precision: 92.89940828402366 %
Support Vector Machine Recall: 97.97191887675507 %
Support Vector Machine f1_score: 95.36826119969628 %
=====

```

	precision	recall	f1-score	support
Negatif	0.93	0.98	0.95	1282
Positif	0.88	0.66	0.75	282
accuracy			0.92	1564
macro avg	0.90	0.82	0.85	1564
weighted avg	0.92	0.92	0.92	1564

Gambar IV-38 Hasil dari Visualisasi Akurasi SVM Linear ke 1

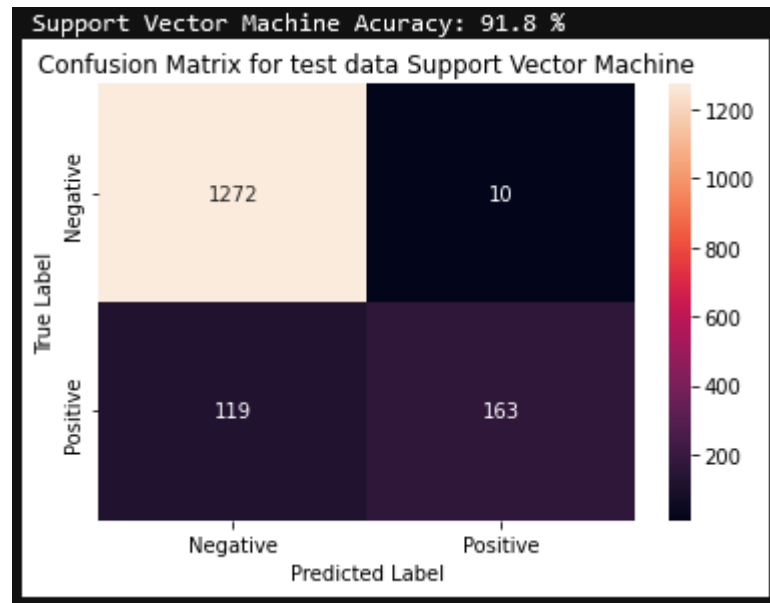


Gambar IV-39 Hasil dari Visualisasi Akurasi SVM Linear ke 2

SVM RBF Acuracy: 91.75191815856778 %
SVM RBF Precision: 94.21965317919076 %
SVM RBF Recall: 57.801418439716315 %
SVM RBF f1_score: 71.64835164835164 %
=====

	precision	recall	f1-score	support
Negatif	0.91	0.99	0.95	1282
Positif	0.94	0.58	0.72	282
accuracy			0.92	1564
macro avg	0.93	0.79	0.83	1564
weighted avg	0.92	0.92	0.91	1564

Gambar IV-40 Hasil dari Visualisasi Akurasi SVM RBF ke 1



Gambar IV-41 Hasil dari Visualisasi Akurasi SVM RBF ke 2

4.6 Klasifikasi Data Menggunakan KNN

Pada tahap ini sama seperti tahap sebelumnya data yang telah melalui proses pembobotan data, data selanjutnya masuk di tahap klasifikasi. Tahapan ini bertujuan untuk mengklasifikasikan data berdasarkan sentimen yang telah didapatkan sebelumnya. Pada tahap ini klasifikasi dilakukan menggunakan metode *K-Nearest Neighbour* (KNN). Cara yang digunakan pada tahap ini adalah sebagai berikut :

```
text_clf = Pipeline([('vect', CountVectorizer()),
                    ('tfidf', TfidfTransformer()),
                    ('clf', KNeighborsClassifier(n_neighbors=10)),
                    ])

text_clf.fit(train_X_df, train_Y_df)
predicted = text_clf.predict(test_X_df)
```

Gambar IV-42 Kode dari Klasifikasi KNN

Setelah dilakukan pemodelan pada data *train* dan klasifikasi data *test* selanjutnya ditampilkan visualisasi berupa akurasi dari klasifikasi yang dilakukan pada proses sebelumnya. Cara yang digunakan untuk menampilkan visualisasi akurasi adalah sebagai berikut :

```
print("K-Nearest Neighbors Accuracy:", accuracy_score(test_Y_df, predicted)*100, "%")
print("K-Nearest Neighbors Precision:", precision_score(test_Y_df, predicted, average="binary",
pos_label="Negatif")*100, "%")
print("K-Nearest Neighbors Recall:", recall_score(test_Y_df, predicted, average="binary",
pos_label="Negatif")*100, "%")
print("K-Nearest Neighbors f1_score:", f1_score(test_Y_df, predicted, average="binary",
pos_label="Negatif")*100, "%")

print(f'Confusion Matrix:\n {confusion_matrix(test_Y_df, predicted)}')
print('=====\n')
print(classification_report(test_Y_df, predicted, zero_division=0))
```

Gambar IV-43 Kode dari Visualisasi Akurasi KNN ke 1

```
print("K-Nearest Neighbors Accuracy:", accuracy_score(test_Y_df, predicted)*100, "%")

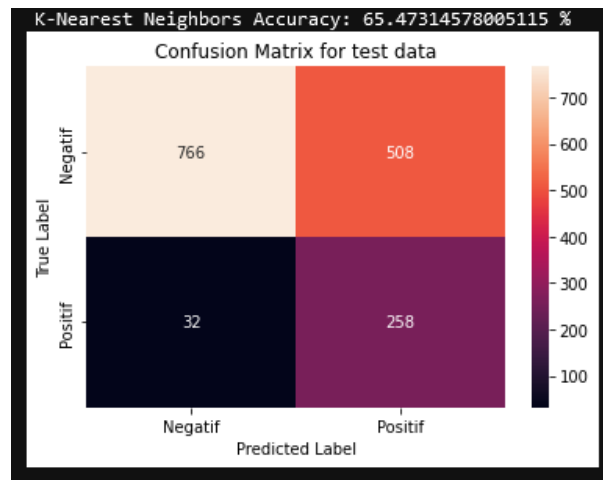
conf_mat = confusion_matrix(test_Y_df, predicted)
class_label = ["Negatif", "Positif"]
test = pd.DataFrame(conf_mat, index = class_label, columns = class_label)
sns.heatmap(test, annot = True, fmt = "d")
plt.title("Confusion Matrix for test data")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```

Gambar IV-44 Kode dari Visualisasi Akurasi KNN ke 2

```
K-Nearest Neighbors Accuracy: 65.47314578005115 %
K-Nearest Neighbors Precision: 95.98997493734336 %
K-Nearest Neighbors Recall: 60.12558869701727 %
K-Nearest Neighbors f1_score: 73.93822393822393 %
Confusion Matrix:
[[766 508]
 [ 32 258]]
=====
```

	precision	recall	f1-score	support
Negatif	0.96	0.60	0.74	1274
Positif	0.34	0.89	0.49	290
accuracy			0.65	1564
macro avg	0.65	0.75	0.61	1564
weighted avg	0.84	0.65	0.69	1564

Gambar IV-45 Hasil dari Visualisasi Akurasi KNN ke 1



Gambar IV-46 Hasil dari Visualisasi Akurasi KNN ke 2

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Data yang didapatkan dalam *scrapping* data sebesar 15.632 data. Hasil dari sentimen yang diperoleh menggunakan *Vader* dalam penelitian ini yaitu positive sebesar 46 %, negative sebesar 35 % dan netral sebesar 19 %. Dari hasil ini dapat disimpulkan tersebut menunjukkan kepuasan masyarakat terhadap kebijakan yang dikeluarkan tersebut.

Di dalam proses klasifikasi, ditemukan untuk klasifikasi KNN iterasi terbaik untuk penelitian ini adalah sejumlah 4 kali untuk menghasilkan akurasi yang cukup akurat. Serta kesimpulan dari klasifikasi tersebut adalah metode SVM lebih akurat dibandingkan metode KNN dengan akurasi sebesar 92 % untuk SVM dan 75 % untuk KNN.

5.2 Saran

Penelitian ini hanya membandingkan keakuratan dari kedua metode klasifikasi (KNN & SVM). Pada penelitian selanjutnya diharapkan peneliti untuk menambah atau merubah metode klasifikasi pada penelitian agar dapat mendapatkan lebih banyak faktor untuk menentukan sentimen dari data yang digunakan.

Daftar Referensi

- [1] K. Makice, *Twitter API: Up and Running*. 2009.
- [2] N. P. Aprilia, D. Pratiwi, and A. Barlianto, “Sentiment Visualization Of Covid-19 Vaccine Based On Naïve Bayes Analysis,” vol. 6, no. 2, pp. 195–208, 2021.
- [3] G. A. Buntoro, “Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter,” *INTEGER J. Inf. Technol.*, vol. 1, no. 1, pp. 32–41, 2017, [Online]. Available: https://www.researchgate.net/profile/Ghulam_Buntoro/publication/316617194_Analisis_Sentimen_Calon_Gubernur_DKI_Jakarta_2017_Di_Twitter/links/5907eee44585152d2e9ff992/Analisis-Sentimen-Calon-Gubernur-DKI-Jakarta-2017-Di-Twitter.pdf
- [4] T. Mustaqim, K. Umam, and M. A. Muslim, “Twitter text mining for sentiment analysis on government’s response to forest fires with vader lexicon polarity detection and k-nearest neighbor algorithm,” *J. Phys. Conf. Ser.*, vol. 1567, no. 3, 2020, doi: 10.1088/1742-6596/1567/3/032024.
- [5] G. D. Hamidi, F. A. Bestari, A. Situmorang, and N. A. Rakhmawati, “Sentiment Analysis on the Ratification of Penghapusan Kekerasan Seksual Bill on Twitter,” *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 3, pp. 655–665, 2021, doi: 10.28932/jutisi.v7i3.4051.
- [6] M. Lutz, *Python pocket ref*. 2014.
- [7] A. Novantirani, M. K. Sabariah, and V. Effendy, “Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine,” *e-Proceeding Eng.*, vol. 2, no. 1, pp. 1–7, 2015.
- [8] B. Zhao, “Encyclopedia of Big Data,” *Encycl. Big Data*, no. May 2017, 2020, doi: 10.1007/978-3-319-32001-4.
- [9] JustAnotherArchivist, “snsrape: A social networking service scraper in

- Python,” 2021. <https://github.com/JustAnotherArchivist/snscrape> (accessed Oct. 22, 2021).
- [10] J. Patterson and A. Gibson, *Deep learning: A Practionar Approach*, vol. 521, no. 7553. 2017. [Online]. Available: <http://www.nature.com/doifinder/10.1038/nature14539>
 - [11] “6.2. Feature extraction — scikit-learn 1.0.2 documentation.” https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction (accessed May 07, 2022).
 - [12] M. Desai and M. A. Mehta, “Techniques for sentiment analysis of Twitter data: A comprehensive survey,” *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2016*, pp. 149–154, 2017, doi: 10.1109/CCAA.2016.7813707.
 - [13] S. Elbagir and J. Yang, “Language Toolkit and VADER Sentiment,” *Proc. Int. MultiConference Eng. Comput. Sci.*, vol. 0958, pp. 12–16, 2019.
 - [14] E. Hutto, C.J. and Gilbert, “VADER: A Parsimonious Rule-based Model for,” *Eighth Int. AAI Conf. Weblogs Soc. Media*, p. 18, 2014, [Online]. Available: <https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/viewPaper/8109>
 - [15] R. Feldman and J. Sanger, *The Text Mining Handbook*. 2007.
 - [16] F. Gorunescu, *Data Mining Concepts, Models and Techniques*. 2011.