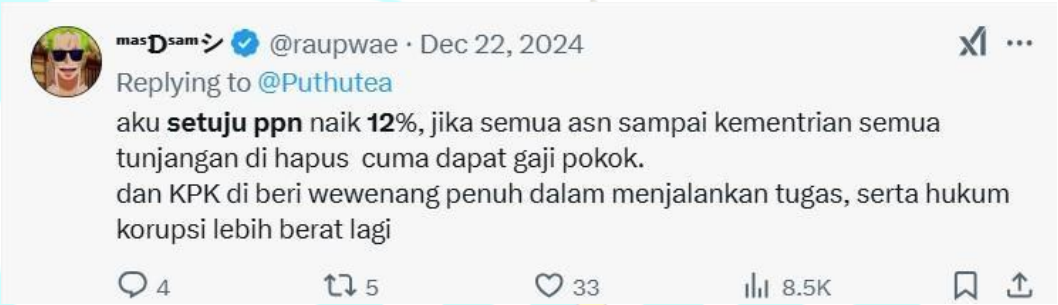
BAB III
METODE PENELITIAN

3.1. Objek Penelitian

Objek Penelitian yang digunakan dalam penelitian ini merupakan opini publik pada sebuah *platform* media sosial *X*. Informasi diambil melalui proses *Crawling* Data menggunakan bahasa pemrograman *python* yang berhasil didapatkan sebanyak 3.332 data dengan rentan waktu dari mulai 1 Januari 2024 sampai 31 Desember 2024 secara bertahap dengan periode satu bulan.

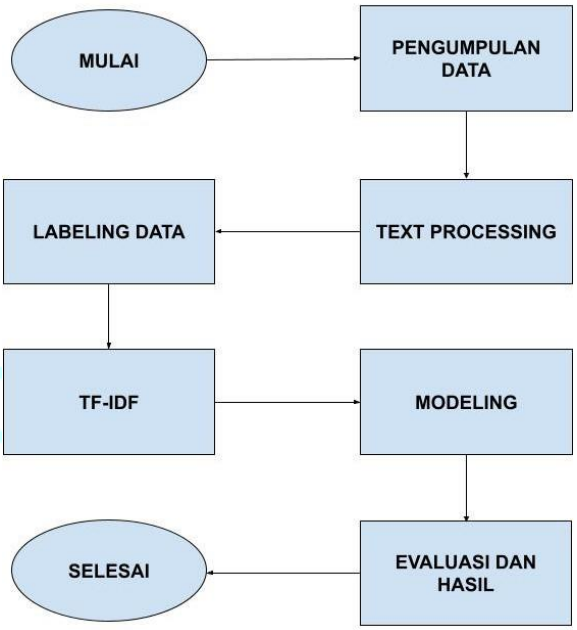


Gambar 3. 1 Contoh *Post* Positif

Gambar 3. 2 Contoh *Post* Negatif

Pada Gambar 3.1 terdapat kalimat “aku setuju PPN naik 12%, jika semua ASN sampai kementrian semua tunjangan dihapus Cuma dapat gaji pokok”. Kata “setuju” artinya tidak menolak kebijakan pemerintah, dengan usulan mempertimbangkan tunjangan Aparatur Sipil Negara sampai tingkat kementrian. Sedangkan pada Gambar 3.2 *post* dengan kalimat “kesannya ‘oh enggak, ini PPN 12% cuman buat barang premium’, padahal isinya *hyper-specific horsesit*”. Dalam *post* tersebut yang menyebutkan adanya upaya memberikan kesan yang kurang baik, serta kata *“horseshit”* yang berarti omong kosong atau tidak masuk akal.

3.2. Prosedur Penelitian



Gambar 3. 3 Alur Prosedur Penelitian

3.3. Pengumpulan Data

Penelitian ini diawali dengan proses pengumpulan data, yang mencakup literatur serta dataset yang diperoleh dari media sosial *X*. Pengumpulan data dalam penelitian ini terbagi menjadi dua tahap utama sebagai berikut:

1. Studi Literatur

   Studi literatur dilakukan untuk memperoleh dasar teori serta referensi dari jurnal dan buku yang relevan. Tujuan dari tahap ini adalah untuk memahami konsep teknik *data mining* yang digunakan dalam tinjauan pustaka, meninjau penelitian sebelumnya, serta menentukan metode yang akan diterapkan dalam penelitian ini.

2. *Crawling Data*

   Proses *crawling data* dilakukan untuk mengambil data opini pada media sosial *X* menggunakan bahasa pemrograman *python* dalam *google colab*, pencarian data menggunakan *hastag* '#KenaikanPPN' dalam rentan waktu 1 Januari 2024 sampai 31 Desember 2024 dengan jumlah data yang didapatkan yaitu 3.332 data.

	conversation_id_str	created_at	favorite_count	full_text	id_str	image_url	in_reply_to_screen_name	lang	location	quote_count	reply_count	retweet_count	
0	1752604375380562373	Wed Jan 31 12:28:45 +0000 2024	0	@tanyakanrl Ppn thn depan 12%	1752670252121587952	NaN	tanyakanrl	in	NaN	0	0	0	https://x.com/sugarla
1	1752277764596789378	Tue Jan 30 14:05:14 +0000 2024	0	@pemanisbuataan @tempodotco Udah mau naikin pp...	1752332143358075350	NaN	pemanisbuataan	in	NaN	0	1	0	https://x.com/V4Ven
2	1751958193700299097	Mon Jan 29 14:13:02 +0000 2024	1	@babyto332 Nama dipanggil HRD trus dikasih amp...	1751971720406401204	NaN	babyto332	in	NaN	0	1	0	https://x.com/IlhamH
3	1751763000321839546	Mon Jan 29 02:12:42 +0000 2024	0	@kikysaputrii Indonesia orangnya hebat.. hebat b...	1751790441249534253	NaN	kikysaputrii	in	Indonesia	0	0	0	https://x.com/gundul_
4	1751502645968412905	Sun Jan 28 14:23:00 +0000 2024	0	@tempodotco Yang bisa batalin kenaikan PPN jad...	1751611838150914312	NaN	tempodotco	in	Jakarta	0	0	0	https://x.com/freedaey
5	1751529095144603945	Sun Jan 28 10:24:15 +0000 2024	1	@Rizlamic @tempodotco @CutSarina5 Tahun dpn pp...	1751551755794710800	NaN	Rizlamic	in	Indonesia	0	0	0	https://x.com/wahidxe
				@geloraco Tahun									

Gambar 3. 4 Hasil Crawling Data

3.4. Text Processing

Sebelum memasuki tahapan pemrosesan data lebih lanjut, dilakukan tahap *text processing* yang meliputi: *Cleaning, Tekonizing, Stopword Removal, Stemming, dan Casefolding.*

1. *Cleaning Data*

   Cleaning data merupakan langkah awal yang sangat penting dalam praproses. Tahap ini bertujuan untuk mengidentifikasi dan mengatasi berbagai masalah kualitas data yang dapat memengaruhi hasil analisis. Proses pembersihan data melibatkan penghapusan atau perbaikan data yang tidak lengkap, tidak akurat, tidak relevan, atau terduplikasi.

   Pembersihan data melibatkan penghapusan karakter yang tidak perlu seperti simbol khusus, tanda baca yang berlebihan, angka yang tidak relevan, dan elemen pemformatan seperti *tag HTML* atau *URL*. Pembersihan data juga melibatkan penanganan penyandian karakter yang tidak cocok, yang sering terjadi saat data berasal dari berbagai sumber dengan standar penyandian yang berbeda.

2. *Case Folding*

   Case folding adalah proses normalisasi teks yang mengubah semua karakter huruf dalam teks menjadi format yang konsisten, biasanya menggunakan huruf kecil. Tahap ini sangat penting dalam pemrosesan teks karena komputer menganggap huruf besar dan huruf kecil sebagai karakter yang berbeda, sehingga kata "Data", "DATA", dan "data" akan dianggap sebagai tiga kata yang berbeda.

Penerapan *case folding* memastikan konsistensi dalam representasi kata, sehingga mengurangi dimensi kosakata dan meningkatkan efisiensi pemrosesan. Proses ini juga membantu menghindari bias yang dapat terjadi karena variasi kapitalisasi yang tidak konsisten dalam data. Namun, perlu dicatat bahwa dalam beberapa kasus, kapitalisasi dapat memiliki makna penting (seperti akronim atau nama diri), sehingga penerapan *case folding* harus disesuaikan dengan konteks dan tujuan analisis.

3. *Slang Word*

   Slang Word merupakan tahap normalisasi bahasa yang mengubah kata-kata non-standar, singkatan, dan slang menjadi bentuk formal atau standar. Tahap ini sangat relevan dalam menganalisis teks yang berasal dari media sosial, *chat*, atau komunikasi informal lainnya yang sering menggunakan bahasa non-standar.

   Proses ini melibatkan penggunaan kamus atau pemetaan kamus yang berisi pasangan kata slang dan kata baku. Misalnya, dalam bahasa Indonesia, kata "gue" akan diubah menjadi "saya", "lo" menjadi "kamu", atau "yg" menjadi "yang". Normalisasi ini penting untuk memastikan bahwa kata-kata dengan makna yang sama tetapi ditulis dalam bentuk yang berbeda dapat dikenali sebagai entitas yang sama oleh algoritma.

4. *Tokenizing*

   Tokenizing adalah proses mendasar dalam pemrosesan teks yang memecah teks menjadi unit-unit yang lebih kecil yang disebut token. Token biasanya berupa kata, frasa, atau bahkan karakter, tergantung pada tingkat ketelitian yang diinginkan untuk analisis. Proses tokenisasi mengubah rangkaian teks yang berkesinambungan menjadi daftar atau larik elemen-elemen terpisah yang dapat diproses secara individual.

   Implementasi tokenisasi dapat menggunakan berbagai strategi, mulai dari tokenisasi spasi sederhana hingga teknik yang lebih canggih yang memperhitungkan tanda baca, konteks, dan aturan linguistik. Dalam bahasa Indonesia, tantangan khusus dalam tokenisasi meliputi penanganan kata majemuk kompleks, reduplikasi, dan afiks.

5. *Stopword Removal*

   Stopword Removal adalah proses penghapusan kata-kata yang sangat umum dan dianggap tidak memberikan informasi penting untuk analisis. Stopword adalah kata-kata yang memiliki frekuensi tinggi dalam suatu bahasa tetapi memiliki nilai informasi yang rendah, seperti konjungsi, kata ganti, artikel, dan preposisi.

   Contoh stopword antara lain "dan", "atau", "dengan", "di", "ke", "dari", "yang", "ini", "itu", "adalah", "untuk", "pada", "di", "saya", "kamu", "dia", dan seterusnya. *Stopword Removal* bertujuan untuk mengurangi noise dalam data, mengurangi dimensionalitas fitur, dan meningkatkan fokus pada kata-kata yang memiliki makna penting untuk analisis.

6. *Stemming*

   Stemming merupakan proses morfologi yang bertujuan untuk mengembalikan kata-kata berafiks ke bentuk dasarnya (kata dasar) dengan menghilangkan afiks seperti prefiks, infiks, dan sufiks. Proses ini sangat penting untuk mengurangi variasi morfologi kata-kata yang memiliki makna dasar yang sama, sehingga meningkatkan efisiensi dan akurasi dalam analisis teks.

   Stemming menjadi lebih kompleks dibandingkan dengan bahasa Inggris karena struktur morfologi bahasa Indonesia yang lebih kaya dan aturan pembentukan kata yang lebih kompleks. Misalnya, kata "berjalan", "berjalan-jalan", "perjalan", dan "perjalan" semuanya memiliki akar kata "jalan". Algoritma *stemming* harus mampu mengidentifikasi dan menghilangkan berbagai kombinasi afiks untuk menghasilkan akar kata yang benar.

3.5. Labeling Data

Setelah seluruh data dibersihkan melalui tahapan *preprocessing* diatas, tahap selanjutnya yaitu melakukan pelabelan data secara manual yang dilakukan oleh seorang ahli Sastra Bahasa Indonesia dengan format *excel*. Berikut hasil proses *labeling data* pada tabel dibawah ini:

Tabel 3. 1 Contoh hasil *Labeling* data positif dan negatif

Labelling Data	
Ulasan	**Keterangan**
omset jualan drop 50 margin keuntungan teru menipi digeru kenaikan harga sbb inflasi pajak pertambahan nilai 11 dpn	**Positif**
pajak pertambahan nilai 12 mudahan makan siang gratis membantu	
bodoh naikkan bbm ga listrik menambah beban masyarakat ditambah kenaikan pajak pertambahan nilai 12 inflasi	**Negatif**
penduduk miskin melonjak ampun kebutuhan pokok terjangkau	

3.6. *Term Frequency – Inverse Document Frequency (TF-IDF)*

Menurut (Tanggraeni dan Sitokdana, 2022) metode *TF-IDF* sebuah proses penghitungan bobot kata yang dihasilkan dari penggabungan dua konsep, yaitu *Term Frequency* dan *Inverse Document Frequency*. Pembobotan *TF-IDF* bertujuan untuk mengetahui bobot setiap kata dalam suatu dokumen.

Term Frequency – Inverse Document Frequency (TF-IDF) dapat dihitung menggunakan rumus sebagai berikut:

$$TF \cdot IDF_{std}(t) = tf_d^t \times \log \frac{n}{df^t} \qquad (9)$$

Keterangan:

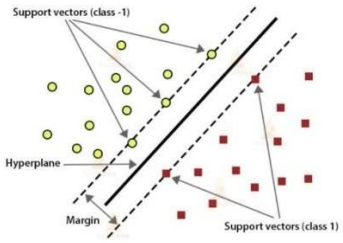
TF = Teks frekuensi

IDF = Teks frekuensi dalam dokumen

n = Jumlah dokumen

3.7. Support Vector Machine

Setelah melalui tahapan-tahapan sebelumnya, maka *dataset* sudah dapat diimplementasikan pada model algoritma. Dalam tahapan ini akan dilakukan suatu proses penentuan parameter terbaik untuk menjalankan algoritma *Support Vector Machine (SVM)*. algoritma *Support Vector Machine (SVM)* bekerja dengan mencari batas keputusan *(hyperplane)* optimal yang dapat memisahkan dua kelas data dengan sejelas mungkin. Dibawah ini adalah ilustrasi cara kerja algoritma *Support Vector Machine* (SVM).



Gambar 3. 5 *Hyperplane Support Vector Machine*

Menurut Rahadi Ramlan et al., (2023) Adapun beberapa langkah algoritma *Support Vector Machine* sebagai berikut:

1. Melakukan inisialisasi terhadap parameter yang diaplikasikan, yaitu λ, α_x, γ, C, dan ∈.
2. Menghitung *Matriks Hessian* (D_{ij}) dengan Persamaan:

$$D_ij = y_i y_j K(x_i \cdot x_j) + \lambda^2 \qquad (10)$$

3. Perhitungan nilai error (E_i) dengan menggunakan persamaan:

$$\backslash E_i = \sum_{j=1}^{n} \alpha_s \cdot D_ij \qquad (11)$$

4. Perhitungan nilai *delta alpha* ($\delta\alpha_i$) dengan menggunakan persamaan:

$$\delta\alpha_i = \min\{\max\{\gamma(1 - E_i), \gamma - \alpha_i\}, C - \alpha_i\} \qquad (12)$$

5. Perhitungan nilai *alpha* baru (α_i) dengan menggunakan persamaan:

$$\alpha_i = \alpha_i + \delta\alpha_i \qquad (13)$$

6. Menghitung nilai w^+(bobot nilai alpha terbesar pada kelas positif) dan w^- (bobot nilai alpha terbesar negatif) menggunakan dua persamaan berikut:

$$w^+ = \sum_{i=1}^{n} \alpha_i y_i (K(x_i, x^+)) \qquad (14)$$

$$w\text{^}+ = \sum\_\{i = 1\}\text{^}n\ \alpha\_i\ y\_i\ (K(x_i, x^-)) \qquad (15)$$

7. Perhitungan nilai bias (b) dengan menggunakan persamaan:

$$b = -\frac{1}{2}(w^+ + w^-) \qquad (16)$$

8. Pengujian terhadap data uji kemudian perhitungan keputusan h(x) menggunakan persamaan:

$$h(x) = \begin{cases} +1, & if\ w.x + b \geq 0 \\ -1, & if\ w.x + b < 0 \end{cases} . \qquad (17)$$

Apabila hasil perhitungan keputusan memiliki nilai 0 atau lebih besar, maka tanda *sign* h(x) adalah +1, menunjukkan kelas positif. Sebaliknya,

jika hasil perhitungan memiliki nilai kurang dari 0, tanda sign h(x) adalah -1. Perhitungan *sign* h(x) menggunakan persamaan:

$$h(x) = w.x + b = \Sigma(i = 1ton)\alpha i.yi.K(xi, xj) + b \qquad (18)$$

3.8. Evaluasi Pengujian

Pada tahap evaluasi ini yaitu melakukan pembagian data menjadi 3 skenario pengujian, yaitu skenario 1 (70% data latih dan 30% data uji), skenario 2 (80% data latih dan 20% data uji), skenario 3 (90% data latih dan 10% data uji). Kemudian dilakukan pengujian model klasifikasi dengan menggunakan *Confusion Matrix* agar dapat melihat analisis kinerja algoritma. Metode evaluasi *Confusion Matrix* dapat diartikan suatu pengukuran performa terhadap permasalahan klasifikasi *machine learning* yang mana berupa hasil lebih dari satu kelas. Adapun persamaan untuk menghitung *Accuracy*, *Precision*, *Recal, dan F1-score* dengan menggunakan *Confusion Matrix* dapat dilihat pada rumus (5), (6), (7), dan (8): (Muhammadi, et al., 2021).