BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen merupakan salah satu cabang *Natural Language Processing* (NLP) yang berfokus pada proses identifikasi, ekstraksi, dan klasifikasi opini maupun sikap subjektif yang terkandung dalam sebuah teks [13]. Pada ranah media sosial, analisis sentimen memegang peranan penting karena memungkinkan respons publik terhadap suatu isu, kebijakan, produk, atau peristiwa dipahami secara otomatis dalam skala besar. Melalui pendekatan ini, berbagai pemangku kepentingan seperti pemerintah, pelaku industri, dan peneliti dapat memperoleh gambaran opini masyarakat yang menyeluruh tanpa perlu menelaah setiap komentar satu per satu secara manual [14].

Secara umum, sentimen diklasifikasikan ke dalam tiga kategori, yakni positif, negatif, dan netral [14]. Ketika diterapkan pada ranah kebijakan publik, analisis sentimen dapat memperlihatkan bagaimana persepsi masyarakat terdistribusi atas suatu keputusan pemerintah. Sejumlah penelitian terdahulu memperlihatkan bahwa YouTube menjadi salah satu sumber data komentar yang kaya untuk analisis sentimen bertema kebijakan, mengingat tingginya tingkat *engagement* pengguna pada konten berita maupun diskusi publik [15, 16].

2.2 YouTube

YouTube merupakan salah satu platform media sosial berbasis video yang banyak digunakan masyarakat untuk berbagi konten sekaligus menyampaikan opini melalui kolom komentar. Tingginya tingkat *engagement* pengguna pada konten berita maupun diskusi publik menjadikan komentar YouTube sebagai sumber data yang kaya untuk analisis sentimen [16]. Sifat komentar yang terbuka dan spontan memungkinkan peneliti memperoleh gambaran opini publik terhadap suatu isu atau kebijakan secara langsung [15].

2.3 Kendaraan Listrik

Kendaraan listrik atau Kendaraan Bermotor Listrik Berbasis Baterai (KBLBB) merupakan kendaraan yang digerakkan oleh motor listrik dengan

sumber energi dari baterai sebagai pengganti mesin pembakaran berbahan bakar minyak. Kendaraan listrik menjadi salah satu instrumen penting dalam transisi energi nasional karena berperan mengurangi emisi gas rumah kaca sekaligus menekan ketergantungan terhadap bahan bakar minyak [2]. Pengembangan ekosistem kendaraan listrik di Indonesia terus didorong melalui berbagai kebijakan pemerintah.

2.4 Insentif Pajak

Insentif pajak merupakan kebijakan fiskal berupa keringanan atau pembebasan kewajiban perpajakan yang diberikan pemerintah untuk mendorong perilaku tertentu, dalam hal ini percepatan adopsi kendaraan listrik. Bentuk insentif pajak yang umum diterapkan antara lain Pajak Pertambahan Nilai Ditanggung Pemerintah (PPN DTP), pembebasan Pajak Penjualan atas Barang Mewah (PPnBM), serta keringanan bea masuk [17]. Insentif ini bertujuan menurunkan harga jual sekaligus menarik investasi pada industri kendaraan listrik dalam negeri.

2.5 Insentif Impor Mobil Listrik

Insentif impor mobil listrik merupakan salah satu bentuk insentif fiskal yang ditujukan untuk mempercepat masuknya kendaraan listrik ke pasar nasional. Melalui Peraturan Presiden Nomor 79 Tahun 2023, pemerintah memberikan pembebasan bea masuk dan PPnBM sebesar 0% bagi impor kendaraan listrik dalam keadaan utuh (*Completely Built-Up*/CBU), dengan syarat produsen berkomitmen membangun produksi di dalam negeri [18]. Insentif ini berlaku hingga akhir tahun 2025 sebagai instrumen transisional sebelum arah kebijakan beralih pada penguatan produksi lokal.

2.6 Natural Language Processing

Natural Language Processing (NLP) merupakan bidang dalam kecerdasan buatan yang berfokus pada pemodelan komputasional bahasa manusia agar komputer mampu menganalisis, menalar, serta menghasilkan teks secara otomatis. Perkembangan NLP didorong oleh ketersediaan data tekstual dalam jumlah besar serta kemajuan teknik representasi bahasa yang memungkinkan teks diubah menjadi bentuk numerik yang dapat diolah komputer. Pada tugas klasifikasi teks seperti

analisis sentimen, proses ini umumnya dilakukan dengan merepresentasikan teks ke dalam fitur numerik yang selanjutnya diklasifikasikan menggunakan algoritma *machine learning* [13].

Dalam konteks analisis sentimen berbahasa Indonesia, NLP menghadapi sejumlah tantangan linguistik yang khas. Bahasa Indonesia memiliki struktur morfologi yang aglutinatif dengan ragam imbuhan yang kompleks, serta dipengaruhi oleh banyak bahasa daerah dan bahasa asing sehingga memunculkan fenomena *code-mixing*. Pada teks media sosial, tantangan tersebut semakin kompleks akibat penggunaan bahasa informal, *slang*, singkatan, serta ekspresi yang tidak baku. Kondisi ini menjadikan tahap *preprocessing* sebagai bagian yang krusial untuk menormalkan teks sebelum diolah lebih lanjut. Penelitian-penelitian terdahulu telah menangani tantangan tersebut melalui pendekatan yang disesuaikan dengan karakteristik bahasa Indonesia [19].

2.7 Text Preprocessing

Sebelum memasuki tahap pemodelan, data mentah perlu diproses terlebih dahulu melalui *text preprocessing* untuk membersihkan sekaligus menyeragamkan format teks supaya lebih mudah diolah oleh algoritma NLP [20]. *Dataset* pada penelitian ini terdiri atas sekitar 2.892 komentar berbahasa Indonesia yang bersumber dari platform YouTube. Oleh karena teks pada media sosial umumnya bersifat tidak formal, sejumlah tahapan prapemrosesan berikut dijalankan secara berurutan:

- **Case Folding**: Penyeragaman format *string* dengan mengubah seluruh karakter huruf ke dalam bentuk huruf kecil (*lowercase*). Tahap ini penting untuk menghilangkan kerangkapan kosakata yang timbul akibat perbedaan penggunaan huruf kapital.
- **Text Cleaning**: Penghapusan elemen non-tekstual yang tidak mengandung muatan sentimen. Elemen yang disingkirkan antara lain tautan URL, angka tanpa konteks, simbol numerik, tanda baca, serta karakter *noise* lainnya.
- **Text Normalization**: Pengubahan kata-kata tidak baku, istilah gaul (*slang*), maupun singkatan khas media sosial menjadi bentuk baku bahasa Indonesia dengan memanfaatkan pemetaan leksikon.
- **Stopword Removal**: Penyaringan untuk menghilangkan kata-kata fungsional

yang sering muncul tetapi memiliki makna semantik rendah (misalnya kata hubung ”dan”, ”yang”, ”di”, ”ke”, ”itu”) demi menekan dimensi data.

- **Tokenization**: Pemecahan rangkaian kalimat utuh menjadi satuan-satuan kata tunggal (*token*) sehingga siap diekstraksi nilainya oleh model klasifikasi.
- **Lemmatization**: Pengubahan kata berimbuhan menjadi bentuk dasarnya (*lemma*) dengan tetap mempertimbangkan konteks sehingga kata dasar yang dihasilkan tetap baku. Berbeda dengan *stemming* yang hanya memotong imbuhan, *lemmatization* menghasilkan kata dasar yang lebih akurat secara linguistik sekaligus mereduksi dimensi fitur dengan menyatukan varian morfologi dari kata yang sama.

2.8 Manual Labeling

Pelabelan manual merupakan proses pemberian label kelas sentimen pada setiap data teks yang dilakukan secara langsung oleh anotator manusia berdasarkan pemahaman terhadap makna dan konteks komentar. Hasil pelabelan ini berperan sebagai data acuan kebenaran (*ground truth*) yang menjadi dasar bagi model untuk mempelajari pola hubungan antara fitur teks dan kelas sentimennya [13]. Berbeda dengan pelabelan otomatis berbasis leksikon yang menentukan polaritas dari akumulasi skor kata, pelabelan manual memungkinkan anotator mempertimbangkan konteks kalimat, makna tersirat, serta nuansa bahasa informal yang sulit ditangkap secara otomatis, sehingga label yang dihasilkan cenderung lebih representatif terhadap maksud sebenarnya dari penulis komentar.

Agar proses pelabelan manual bersifat konsisten dan dapat dipertanggungjawabkan, diperlukan indikator atau kriteria yang menjadi acuan dalam menentukan kelas dari setiap komentar. Penetapan indikator ini bertumpu pada pembedaan antara kalimat subjektif yang mengandung opini dan kalimat objektif yang bersifat faktual, serta arah polaritas opini yang terkandung di dalamnya [13]. Pada penelitian ini, komentar diklasifikasikan ke dalam tiga kelas, yaitu positif, negatif, dan netral [14], dengan indikator sebagaimana dirangkum pada Tabel 2.1.

Tabel 2.1. Indikator penentuan kelas sentimen

Kelas Sentimen	**Indikator**	**Contoh Data**
Positif	Komentar mengandung dukungan, persetujuan, pujian, optimisme, dan harapan terhadap pencabutan insentif impor mobil listrik maupun rasa ketidakadilan mengenai mobil listrik yang dibebaskan pajak.	"Okay lah subsidi kendaraan listrik di cabut. Biar adil semua rakyat diberikan gratis listrik di rumah masing-masing, jadi semua bisa nge cas di rumah."
Negatif	Komentar mengandung penolakan, kritik, ketidaksetujuan, kekecewaan, keluhan, sindiran, kekhawatiran, atau pesimisme terhadap kebijakan pemerintah, kenaikan harga, dan ketidakkonsistenan pemerintah.	"Sok jago pemerintah, target penjualan 2025 aja harus dikoreksi turun. Mobil nasional itu resiko banyak korupsi, urus aja pabrik biar bisa export."
Netral	Komentar bersifat objektif atau faktual, berupa pertanyaan maupun pernyataan informatif tanpa muatan opini yang jelas, termasuk komentar yang mengandung sentimen positif dan negatif secara berimbang sehingga tidak condong pada salah satu kutub.	"Pertanyaannya, jika harga EV naik sekitar 10% apakah akan tetap laku? Atau konsumen akan balik ke mobil ICE? Karena harga EV tidak kompetitif lagi."

2.9 Pseudo-Labeling

Pseudo-labeling adalah salah satu metode dalam *semi-supervised learning* yang pertama kali diperkenalkan oleh Lee [21]. Melalui teknik ini, data yang belum berlabel akan memperoleh *pseudo-label* dari hasil prediksi model, yang kemudian dimanfaatkan bersama-sama dengan data berlabel asli pada tahap pelatihan model. Pendekatan semacam ini berguna untuk memaksimalkan penggunaan data tak berlabel sekaligus berpeluang memperkuat daya generalisasi model, khususnya ketika *pseudo-label* yang dihasilkan disertai tingkat keyakinan (*confidence*) yang tinggi [21]. Secara garis besar, langkah-langkah *pseudo-labeling* yang diterapkan dalam penelitian ini diuraikan sebagai berikut:

1. Menganotasi secara manual sebanyak 750 sampel data yang berfungsi sebagai himpunan berlabel awal (*seed labeled set*).

2. Membangun model awal (*teacher*) dengan memanfaatkan himpunan berlabel awal tersebut.

3. Memprediksi label pada data tak berlabel, lalu menyeleksi *pseudo-label* yang probabilitas tertingginya melampaui nilai ambang.

4. Memadukan *pseudo-label* terpilih dengan data hasil anotasi manual sebagai bahan pelatihan model akhir (*student*).

2.10 Cohen's Kappa

Cohen's Kappa merupakan metrik statistik yang mengukur tingkat kesepakatan antara dua pengamat atau anotator sambil memperhitungkan kesepakatan yang mungkin terjadi secara kebetulan [22]. Metrik ini lebih informatif dibandingkan persentase kesepakatan sederhana karena menghilangkan pengaruh kesepakatan yang diharapkan terjadi hanya melalui peluang acak.

Nilai *Cohen's Kappa* dihitung menggunakan rumus berikut:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{2.1}$$

Dengan P_o adalah proporsi kesepakatan yang teramati (*observed agreement*) dan P_e adalah proporsi kesepakatan yang diharapkan secara kebetulan (*expected agreement*). Kesepakatan teramati dihitung sebagai jumlah prediksi yang sesuai dibagi total jumlah sampel, sedangkan kesepakatan harapan dihitung berdasarkan distribusi marginal dari kedua anotator. Untuk data dengan kategori multi-kelas, kesepakatan teramati dihitung sebagai:

$$P_o = \frac{n_{kk}}{N} \tag{2.2}$$

Dengan n_{kk} adalah jumlah sampel yang sama-sama diklasifikasikan ke kategori yang sama oleh kedua anotator, dan N adalah total jumlah sampel. Kesepakatan harapan (P_e) adalah probabilitas kesepakatan yang terjadi secara kebetulan jika kedua anotator membuat keputusan secara independen. Rumusnya adalah:

$$P_e = \sum_{k=1}^{c} \left(\frac{n_{1k}}{N} \times \frac{n_{2k}}{N} \right) \tag{2.3}$$

Dengan n_{1k} adalah jumlah sampel yang diklasifikasikan ke kategori k oleh Anotator 1, n_{2k} adalah jumlah sampel yang diklasifikasikan ke kategori k oleh Anotator 2, c adalah jumlah kategori, dan N adalah total jumlah sampel. Interpretasi nilai *Cohen's Kappa* mengikuti standar yang telah ditetapkan dalam literatur, sebagaimana ditunjukkan pada Tabel 2.2.

Tabel 2.2. Interpretasi nilai *Cohen's Kappa*

Nilai Kappa	**Kategori**	**Tingkat Reliabilitas**
$\kappa < 0$	Kurang dari kebetulan	Tidak dapat diterima
$0 \leq \kappa < 0,21$	Slight (Sangat Lemah)	Sangat buruk
$0,21 \leq \kappa < 0,41$	Fair (Lemah)	Buruk
$0,41 \leq \kappa < 0,61$	Moderate (Sedang)	Cukup baik
$0,61 \leq \kappa < 0,81$	Substantial (Substansial)	Baik
$\kappa \geq 0,81$	Almost Perfect (Hampir Sempurna)	Sangat baik

2.11 Class Weight

Ketidakseimbangan kelas (*class imbalance*) pada dataset klasifikasi dapat ditangani secara efektif melalui penerapan teknik *class weight*. Pada penelitian analisis sentimen, kondisi tidak seimbang ini kerap muncul akibat distribusi sampel antarkelas positif, negatif, dan netral yang tidak merata, sehingga model berpotensi condong memihak kelas mayoritas [23]. Prinsip kerja *class weight* terletak pada pemberian bobot penalti yang lebih besar bagi kelas minoritas dan bobot yang lebih kecil bagi kelas mayoritas selama proses pelatihan. Dengan mekanisme tersebut, setiap kekeliruan prediksi terhadap kelas minoritas dikenai sanksi (*penalty*) yang berat, sehingga model terdorong untuk lebih jeli mengenali pola pada data berjumlah sedikit. Dibandingkan *oversampling* maupun *undersampling*, keunggulan pendekatan ini terletak pada efisiensinya dalam menyesuaikan kepekaan model terhadap kesalahan tanpa mengubah struktur ataupun banyaknya sampel pada dataset awal [24].

2.12 Random Oversampling (ROS)

Random Oversampling (ROS) merupakan teknik penyeimbangan data yang bekerja dengan menggandakan sampel pada kelas minoritas secara acak hingga jumlahnya mendekati atau setara dengan kelas mayoritas [25]. Sampel yang digandakan merupakan duplikasi langsung dari data yang sudah ada, sehingga tidak

menghasilkan informasi baru, melainkan hanya menambah frekuensi kemunculan sampel minoritas pada data latih. Kelebihan ROS terletak pada kesederhanaan dan kemudahan implementasinya. Akan tetapi, karena sampel yang ditambahkan bersifat identik dengan data aslinya, ROS berpotensi menyebabkan *overfitting*, yaitu kondisi ketika model terlalu menghafal pola spesifik dari sampel yang berulang sehingga kemampuan generalisasinya terhadap data baru menurun [25].

2.13 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan teknik *oversampling* yang dikembangkan untuk mengatasi keterbatasan duplikasi pada ROS dengan cara menghasilkan sampel sintetis baru, bukan sekadar menggandakan sampel yang sudah ada [26]. SMOTE bekerja dengan memilih sebuah sampel pada kelas minoritas, kemudian menentukan sejumlah k tetangga terdekatnya (*k-nearest neighbors*) dari kelas yang sama. Sampel sintetis baru dibentuk melalui interpolasi linear, yaitu dengan mengambil titik di sepanjang garis yang menghubungkan sampel asli dengan salah satu tetangga terdekatnya yang dipilih secara acak [26]. Dengan menghasilkan sampel baru yang tidak identik dengan data aslinya, SMOTE dapat memperluas area keputusan (*decision region*) kelas minoritas sehingga membantu model mengenali polanya secara lebih baik dan mengurangi risiko *overfitting* dibandingkan ROS. Meskipun demikian, sampel sintetis yang dihasilkan tidak selalu sesuai dengan distribusi asli kelas minoritas, sehingga pada kondisi tertentu dapat menimbulkan noise pada data [26].

2.14 Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF merupakan metode pembobotan kata yang mengubah teks menjadi representasi numerik dengan menggabungkan frekuensi kemunculan kata pada sebuah dokumen (TF) dan kelangkaan kata tersebut pada keseluruhan korpus (IDF). Sebuah kata dinilai semakin penting apabila sering muncul pada satu dokumen namun jarang muncul pada dokumen lain [27]. Nilai TF, IDF, dan bobot akhir TF-IDF dihitung melalui Persamaan (2.4), (2.5), dan (2.6).

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.4}$$

$$idf(t) = \log \frac{N}{df(t)} \tag{2.5}$$

$$w(t,d) = tf(t,d) \times idf(t) \tag{2.6}$$

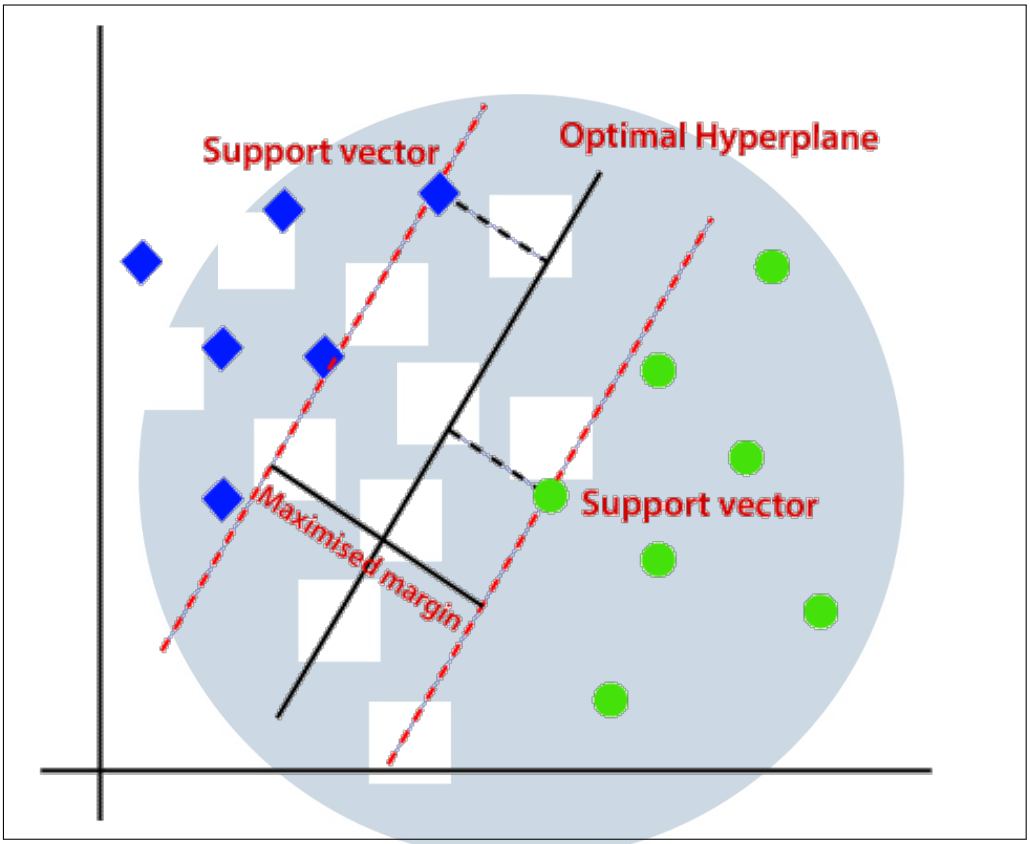
dengan $f_{t,d}$ adalah jumlah kemunculan kata t pada dokumen d, N adalah jumlah seluruh dokumen, dan $df(t)$ adalah jumlah dokumen yang mengandung kata t.

2.15 Pembagian Data (Data Splitting)

Pembagian data (*data splitting*) merupakan tahap memisahkan dataset menjadi data latih (*training set*) dan data uji (*testing set*). Data latih digunakan untuk membangun model klasifikasi, sedangkan data uji digunakan untuk mengukur kinerja model terhadap data yang belum pernah dilihat sebelumnya. Rasio pembagian yang umum digunakan antara lain 80:20 dan 70:30. Pada rasio 80:20, sebanyak 80% data digunakan sebagai data latih dan 20% sebagai data uji, sedangkan pada rasio 70:30 sebanyak 70% data digunakan sebagai data latih dan 30% sebagai data uji. Rasio 80:20 dinilai memberikan keseimbangan yang baik antara jumlah data pelatihan dan keandalan pengujian [28], sementara rasio 70:30 menyediakan porsi data uji yang lebih besar sehingga dapat memberikan estimasi kinerja model yang lebih bervariasi [29]. Hingga saat ini belum terdapat panduan baku mengenai rasio pembagian yang paling optimal untuk suatu dataset, sehingga pemilihan rasio pada praktiknya masih bersifat empiris [30]. Oleh karena itu, penggunaan kedua rasio tersebut dalam penelitian ini bertujuan untuk membandingkan pengaruh proporsi pembagian data terhadap kinerja model klasifikasi.

2.16 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma pembelajaran mesin terbimbing yang mencari bidang pemisah (*hyperplane*) optimal dengan margin terlebar untuk memisahkan antarkelas, di mana data terdekat yang menentukan posisi *hyperplane* disebut *support vector* [27].



Gambar 2.1. Visualisasi SVM linear

Margin yang lebar memberikan kemampuan generalisasi yang baik, sehingga SVM efektif untuk klasifikasi pada ruang fitur berdimensi tinggi seperti representasi TF-IDF [31]. Bidang pemisah tersebut dinyatakan pada Persamaan (2.7).

$$w \cdot x + b = 0 \tag{2.7}$$

Dengan w adalah vektor bobot, x adalah vektor fitur, dan b adalah bias. Pada penelitian ini digunakan kernel linear yang umum dipilih untuk klasifikasi teks karena efektif pada data berdimensi tinggi dan jarang (*sparse*) seperti representasi TF-IDF. Fungsi kernel linear dinyatakan pada Persamaan (2.8).

$$K(x_i, x_j) = x_i^T x_j \tag{2.8}$$

Dengan x_i dan x_j merupakan dua vektor fitur yang dihitung hasil perkalian titiknya (*dot product*).

2.16.1 Kernel Trick

Pada praktiknya, data yang akan diklasifikasikan seringkali tidak dapat dipisahkan secara linear pada ruang fitur aslinya. Untuk mengatasi hal tersebut, SVM memanfaatkan *kernel trick*, yaitu teknik yang memungkinkan algoritma memetakan data ke ruang berdimensi lebih tinggi di mana pemisahan secara linear menjadi mungkin dilakukan [32]. Keunggulan utama teknik ini adalah kemampuannya menghitung nilai skalar vektor (*dot product*) antar data pada ruang berdimensi tinggi tanpa perlu melakukan transformasi secara eksplisit, sehingga beban komputasi dapat ditekan [32]. Secara matematis, fungsi kernel didefinisikan sebagai:

$$K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle \tag{2.9}$$

dengan $\varphi(x)$ merupakan hasil *dot product* dari fungsi pemetaan yang memproyeksikan data masukan ke ruang fitur berdimensi lebih tinggi. Melalui mekanisme ini, batas keputusan yang berbentuk lengkung (non-linear) pada ruang asli dapat diperlakukan sebagai garis lurus pada ruang hasil pemetaan [33]. Beberapa fungsi kernel yang umum digunakan antara lain kernel linear yaitu kernel polinomial dan kernel Radial Basis Function (RBF) [32, 33]. Fungsi kernel polinomial didefinisikan matematis sebagai:

$$K(x_i, x_j) = (x_i^T x_j + c)^d \tag{2.10}$$

dan kernel RBF sebagai:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{2.11}$$

Pemilihan fungsi kernel bergantung pada karakteristik data; kernel RBF banyak dipilih untuk data dengan relasi non-linear yang kompleks [33], sedangkan kernel linear cenderung memadai untuk data berdimensi tinggi dan bersifat *sparse* seperti representasi TF-IDF, karena pada kondisi tersebut data umumnya telah dapat dipisahkan secara linear. Dengan hanya bergantung pada *support vector* sebagai titik penentu batas, penggunaan kernel juga membuat SVM tetap tangguh terhadap permasalahan *curse of dimensionality* [33].

2.16.2 Platt Scaling

Support Vector Machine dioptimasi menggunakan *hinge loss*, yang menghasilkan keluaran berupa skor keputusan (*decision scores*) dalam bentuk bilangan real tanpa interpretasi probabilistik bawaan. Dalam banyak aplikasi, terutama ketika diperlukan nilai kepercayaan (*confidence*) untuk pengambilan keputusan, skor keputusan SVM perlu dikonversi menjadi estimasi probabilitas yang berkalibrasi dengan baik. *Platt scaling* adalah teknik kalibrasi probabilitas yang dikembangkan oleh Platt [34] khusus untuk mengubah keluaran SVM menjadi probabilitas antara 0 dan 1. Metode ini bekerja dengan menyesuaikan fungsi sigmoid logistik terhadap skor keputusan model, sehingga model dapat menghasilkan estimasi probabilitas yang terkalibrasi.

Secara matematis, Platt scaling mentransformasi skor SVM (s) menggunakan fungsi sigmoid sebagai berikut:

$$P(y=1|s) = \frac{1}{1+\exp(-w \cdot s - b)} \tag{2.12}$$

dengan w adalah parameter slope (bobot) dan b adalah parameter intercept (bias) yang dioptimasi menggunakan logistic regression pada himpunan data kalibrasi terpisah. Aplikasi praktis Platt scaling dalam membedakan gempa bumi dari ledakan bawah tanah telah ditunjukkan oleh Pignatelli di mana mereka menggunakan Platt scaling untuk mengkonversi output SVM menjadi probabilitas posteriori, memungkinkan penentuan tingkat kepercayaan (*confidence level*) untuk setiap prediksi [35].

2.17 Hyperparameter Tuning

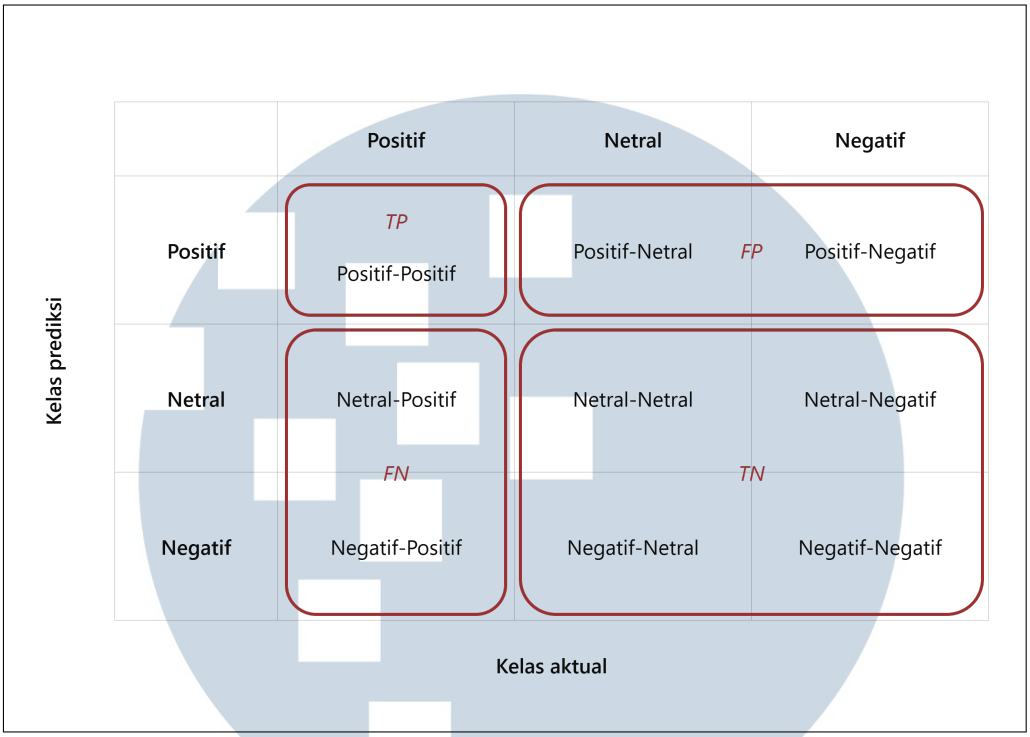
Hyperparameter adalah parameter yang nilainya ditetapkan sebelum proses pelatihan dan tidak dipelajari langsung dari data, misalnya parameter *C* pada SVM dengan kernel linear. *Hyperparameter tuning* merupakan proses pencarian kombinasi nilai *hyperparameter* yang menghasilkan kinerja model terbaik.

2.18 K-Fold Cross Validation

Pengujian model pada penelitian ini menggunakan skema *5-fold cross validation* yang dipadukan dengan pendekatan *stratified split*. Pemilihan lima *fold* ditetapkan untuk mencapai keseimbangan antara efisiensi komputasi dan kestabilan estimasi kinerja model. Pendekatan ini mengacu pada penelitian sebelumnya yang menerapkan *stratified five-fold cross validation* pada model SVM untuk klasifikasi sentimen [36]. Metode tersebut terbukti mampu menjaga konsistensi proporsi kelas pada setiap *fold* sekaligus menghasilkan evaluasi yang stabil.

2.19 Confusion Matrix

Confusion matrix merupakan tabel yang digunakan untuk mengukur kinerja model klasifikasi dengan membandingkan kelas hasil prediksi terhadap kelas sebenarnya [37]. Tabel ini memuat empat komponen utama, yaitu *True Positive* (TP), yang menunjukkan jumlah data yang berhasil diklasifikasikan secara benar sebagai kelas target; *True Negative* (TN), yang menunjukkan jumlah data di luar kelas target yang berhasil dikenali dengan benar; *False Negative* (FN), yang menunjukkan jumlah data dari kelas target yang gagal diidentifikasi oleh model; serta *False Positive* (FP), yang menunjukkan jumlah data dari kelas lain yang keliru diklasifikasikan sebagai kelas target. Keempat komponen tersebut disajikan pada Gambar 2.2.



Gambar 2.2. Confusion matrix 3 kelas

Dengan fokus pada kelas positif, sel Positif-Positif merupakan TP, bagian baris dari kelas positif merupakan FP, bagian kolom dari kelas positif merupakan FN, dan sisanya selain dari kolom dan baris positif merupakan TN. Berdasarkan nilai pada *confusion matrix*, kinerja model dapat diukur melalui empat metrik, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*, sebagaimana dirumuskan pada Persamaan (2.13) sampai (2.16).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.13}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.14}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.15}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.16}$$

Accuracy mengukur proporsi prediksi yang benar terhadap keseluruhan data, *precision* mengukur ketepatan prediksi pada suatu kelas, *recall* mengukur kemampuan model dalam menemukan seluruh data pada kelas tersebut, sedangkan *F1-score* merupakan rata-rata harmonik antara *precision* dan *recall*.