BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen adalah sebuah penelitian yang dilakukan dengan tujuan untuk menganalisis opini, perasaan, dan emosi yang terkandung dalam dokumen atau kumpulan data [11]. Fokus utama dari analisis sentimen adalah untuk mengklasifikasikan teks yang termasuk dalam kalimat atau pendapat. Hasil dari analisis sentimen yang menunjukkan respon dari masyarakat dapat dijadikan sebagai evaluasi terkait tujuan dari topik yang dianalisis. Penerapan dari analisis sentimen dapat dilakukan terhadap berbagai subjek seperti produk, tokoh, dan sebagainya.

Secara umum, sentimen terbagi menjadi dua kategori utama yaitu positif dan negatif [12]. Dimana sentimen positif menunjukkan respon atau reaksi yang baik terhadap subjek penelitian dan sebaliknya sentimen negatif menunjukkan respon atau reaksi yang tidak baik terhadap subjek. Namun, terdapat satu kategori tambahan yang dapat diaplikasikan pada sentimen analisis, yaitu sentimen netral [13]. Dimana sentimen netral menunjukkan respon atau reaksi yang bukan merupakan sentimen positif maupun negatif tetapi berada diantara keduanya.

2.2 Bootcamp

Bootcamp merupakan program pelatihan intensif yang dilakukan dalam durasi waktu tertentu dengan fasilitas yang disediakan oleh penyedia program yang berfokus untuk mempelajari materi tertentu [14]. Melalui program *bootcamp*, peserta akan dibekali dengan ketrampilan-ketrampilan tertentu sesuai dengan materi yang diajarkan yang diikuti oleh peserta.

Program *bootcamp* memberikan akses kepada peserta untuk mempelajari baik teori maupun praktek agar menjadi sumber daya manusia yang siap untuk bekerja di industri digital. Oleh karena itu, program *bootcamp* memiliki peran yang signifikan dalam memenuhi kebutuhan akan sumber daya manusia di industri digital saat ini [15].

2.3 Binar Academy

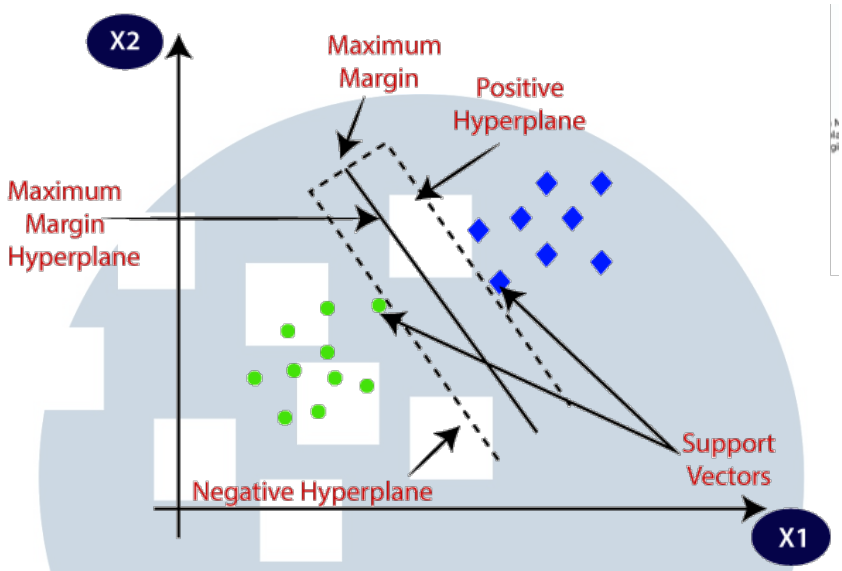
Binar Academy adalah salah satu perusahaan di Indonesia yang bergerak di bidang teknologi dan edukasi. *Binar Academy* didirikan pada tahun 2016 oleh Alamanda Shantika, Dita Aisyah dan Seto Loreno. Tujuan utama didirikannya *Binar Academy* adalah untuk mentransformasi masyarakat yang memiliki minat belajar khususnya di bidang digital menjadi orang-orang yang siap untuk bekerja di industri digital [16].

Program-program pelatihan yang disediakan oleh *Binar Academy* mencakup berbagai bidang digital seperti *web development*, *UI/UX*, *Data Science* dan sebagainya. Dengan adanya program pelatihan tersebut, diharapkan dapat memproduksi sumber daya manusia yang kompeten untuk memajukan industri di Indonesia.

2.3.1 Support Vector Machine

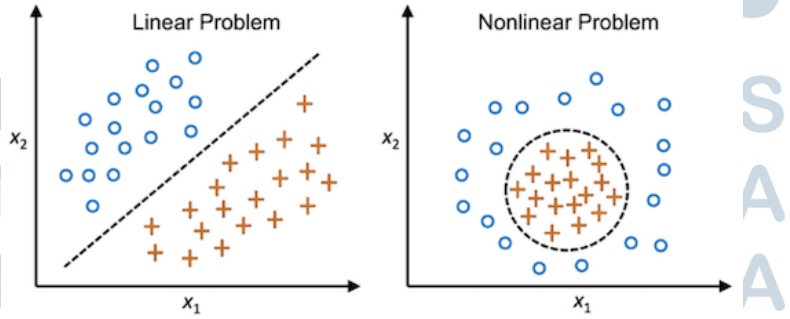
Algoritma *Support Vector Machine* (SVM) merupakan algoritma pembelajaran mesin yang digunakan untuk melakukan klasifikasi. Dengan algoritma SVM, pola-pola yang ada pada data dapat diidentifikasi dan diklasifikan kedalam kelas-kelas tertentu. Algoritma SVM mengidentifikasi data *training* yang memiliki label dan mengelompokkannya ke dalam kelas-kelas tertentu.

Algoritma SVM merupakan metode yang menggunakan *hyperplane* untuk memisahkan kelas-kelas pada data sesuai dengan persamaannya. Pemisahan tersebut dilakukan dengan mengukur *margin hyperplane* dan mengidentifikasi titik maksimumnya. *Margin* merupakan jarak antar *hyperplane* dengan titik data kelas terdekat yang disebut *support vector*.



Gambar 2.1. Algoritma *Support Vector Machine*

Algoritma *Support Vector Machine* dibagi kedalam dua jenis yaitu SVM linear dan non-linear. Umumnya, svm digunakan secara linear yang bekerja dengan mencari *hyperplane* yang memisahkan data dengan margin terbesar. SVM linear digunakan jika data memiliki pola yang jelas karena lebih mudah diimplemtasikan dan performanya baik. Namun untuk data kompleks yang tidak dapat dipisahkan secara linear, SVM non linear lebih baik untuk digunakan dengan menggunakan fungsi kernel [17]. SVM non-linear menggunakan fungsi kernel untuk memindahkan data ke ruang dimensi yang lebih tinggi. Di dimensi baru ini, pola data yang tadinya rumit dan tak terpisahkan secara linear, menjadi rapi dan mudah dipisahkan dengan *hyperplane*. Kemampuan ini menjadikan SVM non-linear pilihan tepat untuk data dengan pola yang rumit dan non-linear.



Gambar 2.2. Linear vs Non-Linear

Kernel merupakan fungsi yang menggunakan dua titik data sebagai masukan

untuk dapat menghasilkan suatu ukuran persamaan diantara keduanya. Ukuran persamaan tersebut kemudian akan digunakan untuk memisahkan ruang fitur yang berdimensi kompleks dimana terdapat data pada fitur berdimensi tinggi dan berdimensi rendah [18]. Dengan menggunakan kernel, transformasi ruang fitur dan perhitungan dapat dilakukan dengan lebih efisien karena kernel melakukan perhitungan di ruang masukan yang sebenarnya. Adapun kernel yang umum digunakan adalah kernel *linear*, kernel *polinomial*, kernel *Radial Basis Function* (RBF), dan kernel *sigmoid*. Setiap kernel memiliki karakteristik masing-masing dengan kesesuaian yang berbeda-beda terhadap data yang digunakan.

$$f(x) = w^T * x + b \tag{2.1}$$

Pada penelitian ini, kernel yang digunakan adalah kernel *linear*. Kernel ini memiliki keunggulan dari segi efisiensi komputasi dan sederhana untuk diaplikasikan. Dalam konteks SVM, kernel linear digunakan untuk memetakan data dari ruang input yang berdimensi rendah ke ruang fitur yang berdimensi lebih tinggi. Dengan menggunakan kernel linear, data yang tidak dapat dipisahkan secara linear di ruang input mungkin dapat dipisahkan secara linear [19]. Kernel linear digunakan untuk menghitung dan memprediksi label seperti pada rumus 2.1. Dimana w adalah vektor bobot yang didapat dari data pelatihan ditranspos dan dikalikan dengan x yang merupakan vektor input atau titik data baru. Kemudian, data akan ditambah dengan b yang merupakan bias. Adapun pengimplementasian SVM dengan kernel linear adalah dengan menemukan vektor bobot (w) dan bias (b) yang meminimalkan *margin* dan memaksimalkan jarak dari *hyperplane* ke data *support* untuk dapat memprediksi data baru.

Algortima *Support Vector Machine* memiliki kemampuan untuk meminimalisir kesalahan pada saat pelatihan data dan mencegah kesalahan yang terjadi karena pengaruh dimensi. Karena kemampuannya untuk menyelesaikan masalah berdimensi tinggi dengan jumlah sampel data yang terbatas membuat algoritma ini sering diimplementasikan pada berbagai penelitian yang berkaitan dengan klasifikasi. Pada penelitian ini, SVM digunakan untuk mengelompokkan data sentimen pengguna ke dalam kategori positif, netral dan negatif. Untuk dapat melakukan hal tersebut, algoritma SVM bekerja dengan mencari *margin hyperplane* yang memisahkan data dengan jarak maksimum. Ulasan tanpa label yang akan diuji nantinya akan dipetakan dan diklasifikasikan berdasarkan posisinya terhadap *hyperplane*.

2.3.2 TF-IDF

TF-IDF adalah metode *feature extraction* yang menggunakan frekuensi terma dan inversi dokumen. Jumlah kata yang muncul dalam suatu kalimat disebut TF, sedangkan IDF adalah inversi dari banyaknya kata yang muncul dalam sebuah artikel. Perhitungan TF dapat menggunakan banyaknya kata dalam suatu kalimat, sementara IDF merupakan kebalikan dari munculnya suatu kata dibandingkan dengan jumlah total kata dalam kalimat. Setiap kata dihitung sebagai satu dalam satu kalimat, dan dua dalam dua kalimat [20].

$$TF - IDF(i,j) = TFxIDF \tag{2.2}$$

TF-IDF diperlukan untuk melakukan pembobotan pada data agar algoritma Support Vector Machine dapat memahami kata yang diinput. TF-IDF dihitung dengan mengalikan antara *Term Frequency* dan *Inverse Document Frequency*. TF dihitung dengan membagi jumlah kemunculan i dalam dokumen j dengan total jumlah term dalam dokumen j. Sedangkan IDF dihitung dengan menghitung logaritma dari jumlah dokumen dibagi jumlah dokumen yang mengandung j[21]. Pada TF-IDF, kata-kata yang jarang muncul secara umum tetapi sering muncul dalam dokumen tertentu akan memiliki bobot yang lebih tinggi. Hal ini membantu dalam proses pencarian informasi karena kata-kata yang lebih spesifik cenderung memberikan informasi yang lebih akurat dan relevan terhadap kueri pencarian.Dengan fokus pada kata-kata spesifik yang lebih informatif dan relevan, TF-IDF membantu menemukan informasi yang lebih akurat dan sesuai dengan kebutuhan pengguna.

2.3.3 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) adalah metode menambahkan data pada kelas minoritas secara sintetis, di mana setiap sampel dari kelas minoritas diambil dan contoh sintetis dimasukkan ke dalam segmen garis yang menghubungkan setiap tetangga terdekat yang termasuk dalam kelas minoritas [22]. Hal ini perlu dilakukan untuk mengatasi masalah *class imbalance* dimana proporsi penyebaran data pada setiap kelasnya tidak seimbang.

Kelas minoritas merupakan kelas yang jumlah sampelnya paling sedikit dan sebaliknya kelas mayoritas merupakan kelas yang memiliki jumlah sampel paling banyak. Untuk dapat menyeimbangkan kelas minoritas dengan kelas mayoritas,

perlu dilakukan identifikasi terlebih dahulu akan kebutuhan kelas mana yang paling membutuhkan adanya penyeimbangan data. Sampel baru dibuat dari sampel acak yang bersumber dari kelas minoritas. Jumlah sampel sintetis yang dibuat dapat diubah sesuai kebutuhan hingga mendekati dan dianggap seimbang dengan jumlah kelas mayoritas. Pada SMOTE, seluruh sampel data minoritas memiliki peluang yang sama untuk dijadikan sampel sintetis [23].

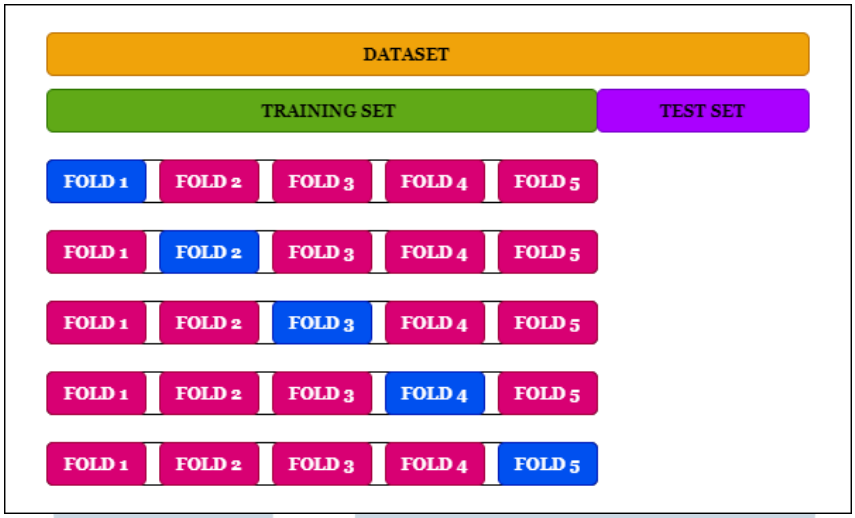
2.3.4 Adaptive Synthetic Sampling (ADA-SYN)

ADA-SYN *(Adaptive Synthetic Minority Oversampling Technique)* merupakan metode untuk mengatasi permasalahan ketidakseimbangan data pada klasifikasi. Ketidakseimbangan data terjadi ketika jumlah data untuk suatu kelas jauh lebih sedikit dibanding kelas lainnya. ADA-SYN bekerja dengan mensintesis data minoritas (kelas dengan jumlah sedikit) berdasarkan jaraknya dengan data minoritas lainnya. Dengan menambah data minoritas buatan, diharapkan model klasifikasi menjadi lebih akurat [24].

Sama seperti SMOTE, ADA-SYN juga merupakan salah satu metode untuk mengatasi masalah pada ketidakseimbangan data. Dalam implemetasinya, ADA-SYN lebih kompleks dibanding SMOTE karena adanya pembobotan pada sampel minoritas yang diukur berdasarkan kesulitan dalam tingkat klasifikasinya. Sehingga sampel minoritas yang lebih sulit diklasifikasikan akan memiliki bobot yang lebih tinggi dan memiliki peluang lebih besar untuk dijadikan sampel sintetis. Oleh karena adanya pembobotan tersebut, distribusi dari sampel data sintetis yang dibuat menggunakan ADA-SYN lebih merata dalam menyerupai kelas minoritas yang asli [25].

2.4 K-Fold Cross Validation

Selain membagi data menjadi data *training* dan data *testing*, perlu juga dilakukan validasi untuk memastikan bahwa seluruh dataset telah digunakan baik untuk pelatihan maupun pengujian. Hal ini dapat dilakukan dengan *K-Fold Cross Validation*. Metode ini digunakan dengan membagi setiap data ke dalam k atau *fold* dengan ukuran yang sebanding [26]. Data dibagi kedalam data *training* dan data *testing* secara beriterasi sesuai dengan jumlah lipatan (*fold*). Hal ini dilakukan untuk melakukan validasi akan hasil pengujian model.



Gambar 2.3. Confusion Matrix

Jumlah iterasi yang umum digunakan pada *K-fold cross validation* adalah 5 atau 10. Pertimbangan dalam memilih jumlah iterasi adalah ukuran dataset dan waktu komputasi. Pada dataset yang kecil atau kompleks, mungkin diperlukan lebih banyak iterasi untuk mencapai stabilitas hasil evaluasi. Namun pada dataset yang besar, iterasi yang lebih sedikit dapat digunakan [27].

Tabel 2.1. Confusion Matrix

Confusion Matrix		
	Actual Positive	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Untuk mengevaluasi model klasifikasi, Confusion Matrix berisi informasi yang membandingkan hasil klasifikasi model dengan hasil yang sebenarnya. Terdapat 4 macam kondisi dalam *confusion matrix* [28], yaitu:

1. TP (True Positive): data yang berhasil diklasifikasi ke dalam kelas positif.
2. TN (True Negative): data yang berhasil diklasifikasi ke dalam kelas negatif.
3. FP (False Positive): data yang gagal diklasifikasi ke dalam kelas positif.
4. FN (False Negative): data yang gagal diklasifikasi ke dalam kelas negatif.

Dengan hasil dari *confusion matrix*, pengujian performa dari model yang telah dibuat dapat dilakukan. Pengujian performa dapat dilakukan dengan melakukan perhitungan *akurasi, precision, recall, dan f1-score* [29].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.6}$$