BAB II
TINJAUAN PUSTAKA

2.1 Kajian Terkait

Dalam penelitian ini terdapat hal yang mengacu kepada beberapa penelitian yang telah dilakukan oleh peneliti-peneliti sebelumnya dengan topik terkait. Hal-hal yang mencakup adalah Analisis Sentimen atau *Opinion Mining*, penggunaan sosial media Twitter sebagai objek penelitian, algoritma *Support Vector Machine*, dan Marketplace.

Pada penelitian yang dilakukan oleh Agustina et al. (2020) yaitu mengimplementasikan algoritma *Support Vector Machine* (SVM) untuk analisis sentimen terhadap *Marketplace* di Indonesia dengan membangun sebuah mesin yang dapat mengklasifikasikan sentimen ke dalam tiga kelas, sentimen positif, negatif, dan netral. Peneliti bertujuan untuk mengetahui performa klasifikasi dari algoritma *Support Vector Machine* (SVM) menggunakan *kernel linier* dan *kernel radial basis function* (RBF). Adapun *Marketplace* yang dipilih dalam penelitian ini yaitu Bukalapak, Shopee, dan Tokopedia. Pembobotan kata dan pengujian sistem dilakukan dengan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) dan *Confusion Matrix*. Dari penelitian yang dilakukan, peneliti mendapatkan hasil adalah sentimen lebih banyak sentimen positif daripada negatif yang diungguli oleh Shopee dan diikuti Tokopedia kemudian Bukalapak. Selain itu, performa klasifikasi menunjukkan nilai *G-mean* dan *AUC* pada Bukalapak sebesar 0,85 dan 0,86 di fold pertama, Shopee sebesar 0,76 dan 0,77 di fold ke tujuh, dan Tokopedia sebesar 0,82 dan 0,83 di fold ke enam,serta penggunaan kernel RBF lebih baik dibandingkan dengan kernel linier.

Penelitian terkait penggunaan metode *Support Vector Machine* (SVM) juga dilakukan oleh Hartanto dan Sari (2019) yaitu mengimplementasikan algoritma *Support Vector Machine* (SVM) dalam membangun aplikasi klasifikasi pengguna Twitter terhadap pelayanan Telkom dan Biznet dengan tujuan mengetahui persentase sentimen positif dan negatif dari Telkom dan Biznet untuk memberikan informasi agar kedua perusahaan dapat meningkatkan kualitas pelayanan. Pembobotan kata dilakukan dengan menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Penggunaan metode *10-Fold*

Cross Validation dan *Confusion Matrix* ditujukan untuk membagi data set, keakuratan model yang dibangun, dan mencari nilai *accuracy, precision,* dan *recall*. Dari penelitian tersebut peneliti mendapatkan hasil Telkom lebih baik pelayanannya dibandingkan dengan Biznet dengan persentase nilai positif 41,2% dan negatif 58,8% untuk Telkom serta nilai positif 35,2% dan negatif 64,8% untuk Biznet. Nilai *accuracy* model yang didapatkan dari penelitian ini yaitu 79,6% untuk klasifikasi sentimen Telkom dan 83,2% untuk klasifikasi sentimen Biznet. Dengan demikian penggunaan algoritma *Support Vector Machine* (SVM) dinilai cocok untuk analisis sentimen data *tweet* Telkom dan Biznet.

Wisudawati et al. (2020) melakukan penelitian analisis sentimen terhadap dampak Covid-19 pada performa Tokopedia. Penelitian ini bertujuan untuk mendapatkan nilai performa Tokopedia. Dalam melakukan penelitian peneliti menggunakan algoritma *Support Vector Machine* (SVM) dengan kernel *radial basis function* (RBF) dikarenakan memiliki tingkat akurasi paling tinggi dalam klasifikasi teks. Dari hasil penelitian dapat disimpulkan bahwa proses pelabelan dan pembobotan dilakukan dengan kamus *lexicon* dan manual. Data yang digunakan yaitu sebesar 133 review untuk bulan februari 2020 dan 209 review untuk bulan april dengan perhitungan akurasi menggunakan *confusion matrix* yaitu bulan Februari 2020 sebesar 87% dan bulan April 2020 sebesar 84% ini menunjukkan hasil klasifikasi di kategori ”Baik”. Hasil penelitian ini adalah sebanyak 43% sentimen negatif ditujukan pada Tokopedia di bulan Februari 2020 sedangkan 27% di bulan April 2020 dimana hasil ini menunjukkan kemunculan Covid-19 di Indonesia tidak mempengaruhi performa Tokopedia.

Sa’adah, et al (2020) juga melakukan penelitian analisis sentimen *review e-commerce* menggunakan algoritma *Support Vector Machine*. Penelitian dilakukan dari bulan Agustus 2019 hingga November 2019 dengan parameter Bukalapak, Shopee, Tokopedia, dan Lazada serta membagi data ke dalam 3 (tiga) kelas yaitu positif, negatif, dan netral. Penggunaan seleksi fitur Information Gain sebagai penguji sistem menghasilkan tingkat akurasi 80,33% sedangkan tanpa seleksi sebesar 78,16% dengan data uji sebesar 10%.

2.2 *Text Mining*

Text mining merupakan sebuah cara atau teknik yang digunakan dalam penambangan data berupa teks. *Text mining* adalah proses penemuan akan informasi atau trend baru yang sebelumnya tidak terungkap dengan memproses dan menganalisa data dalam jumlah besar (Adiwijaya, 2006). Hearst (2003) menyebutkan bahwa tujuan *text mining* yaitu untuk menemukan informasi yang sebelumnya belum diketahui, sesuatu yang belum pernah diketahui orang lain dimana tidak bisa didefinisikan. Mengutip Agustina et al. (2020) text mining diambil dari sekumpulan bahasa alami yang tidak terstruktur, sedangkan data mining pola yang diambil dari database yang terstruktur. Secara garis besar *text mining* merupakan metode yang dilakukan untuk menambang data berbentuk teks yang belum terstruktur untuk menemukan informasi yang sebelumnya belum diketahui.

2.3 Analisis Sentimen (*Opinion Mining*)

Analisis sentimen merupakan cabang dari *text* mining. Analisis sentimen atau *opinion mining* adalah bidang studi yang menganalisis opini publik, sentimen, evaluasi, sikap, dan emosi terhadap objek seperti produk, layanan, organisasi, individu, isu, peristiwa, dan topik (Liu, 2012). Analisis sentimen digunakan untuk mengekstrak data opini, memahami serta mengolah tekstual data secara otomatis untuk melihat sentimen yang terkandung dalam sebuah opini (Sari dan Wibowo, 2019). Dapat disimpulkan analisis sentimen atau *opinion mining* adalah sebuah proses dalam menganalisis opini publik untuk mengetahui sentimen dari opini tersebut. Adapun tahapan-tahapan yang dilakukan dalam analisis sentimen yaitu, pengumpulan data, *preprocessing* data, *feature extraction*, klasifikasi, dan evaluasi.

2.3.1 *Scraping*

Scraping merupakan metode atau teknik dalam pengambilan data. *Scraping* digunakan untuk mengubah data yang tidak terstruktur di web menjadi data yang terstruktur dan dapat disimpan ke dalam sebuah *database* atau *spreadsheet* (Sirisuriya, 2015). Manfaat dari web scraping adalah agar informasi yang dikeruk lebih terfokus sehingga memudahkan dalam melakukan pencarian

sesuatu (Ayani et al, 2019). Dengan adanya teknik *scraping* konten utama dari suatu halaman situs dapat diekstrak, dikoleksi dan selanjutnya dapat diproses oleh proses pengindeksan (Utomo, 2013).

2.3.2 *Text Preprocessing*

Text Preprocessing adalah sebuah tahapan untuk mempersiapkan data. *Text preprocessing* (pemrosesan teks) adalah proses pengubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai dengan kebutuhan untuk proses mining lebih lanjut (analisis sentimen, peringkasan, dan *clustering* dokumen, dan lain-lain) (Fauzi, 2016). *Text preprocessing* dilakukan untuk menghindari data yang kurang sempurna, gangguan pada data, dan data yang tidak konsisten (Adriani et al, 2019). Menurut Kotsiantis (2006) dalam Juniarsih et al. (2020) *text preprocessing* diperlukan untuk menyelesaikan beberapa jenis masalah termasuk *noisy* data, data redundansi, nilai data yang hilang, dan lain-lain. Pada penelitian ini, tahapan *text preprocessing* yang dilakukan adalah *case folding*, *cleaning*, *tokenizing, filtering*, normalisasi kata, dan *stemming*.

2.3.2.1 Case Folding

Case folding merupakan proses untuk mengubah semua teks dokumen menjadi huruf kecil (Prihatini, 2016). Menurut Adi (2018) di dalam Adriani et al (2019) . Tidak semua kata dalam teks konsisten dalam penggunaan huruf kapital disinilah tujuan dilakukan *case folding* untuk mengkonversi setiap karakter dalam kata menjadi huruf kecil.

2.3.2.2 Cleaning

Menurut Fauzi (2017) *Cleaning* adalah proses membersihkan dokumen dari komponen-komponen yang tidak memiliki hubungan dengan informasi yang ada pada dokumen, seperti tag html, link, dan *script*.

2.3.2.3 Tokenizing

Menurut Jumeilah F. S. (2017) Tahap *tokenizing* adalah tahap pemotongan string masukan berdasarkan kata-kata yang menyusunnya atau dengan kata lain pemecahan kalimat menjadi kata. *Tokenizing* merupakan proses untuk memecah

teks dokumen menjadi kata serta menghilangkan angka, tanda baca dan spasi (Prihatini, 2016).

2.3.2.4 Filtering

Menurut Dave et al. (2003) di dalam Juniarsih et al. (2020) Filtering merupakan suatu proses untuk menghilangkan bagian-bagian dari dokumen mentah yang tidak mempunyai relevansi atau tidak memiliki arti bagi proses klasifikasi. Filtering adalah fase menghilangkan kata-kata yang tidak mengandung makna atau stopword (Nurzahputra dan Muslim, 2016). Kata-kata tersebut seperti kata penghubung, kata ganti orang, kata seruan dan kata lainnya yang tidak begitu memiliki arti dalam penentuan kelas topik suatu dokumen (Wahyudin, 2019).

2.3.2.5 Normalisasi Kata

Menurut Pratama E. E. (2019) di dalam Juniarsih et al. (2020) normalisasi kata berfungsi untuk memperbaiki kata-kata yang kurang tepat penulisannya di dalam teks. Normalisasi kata akan mengubah kata yang tidak baku menjadi baku dan mengubah singkatan menjadi kata asalnya (Khairunnisa et al, 2021).

2.3.2.6 Stemming

Stemming merupakan proses untuk mencari kata dasar dari setiap kata dalam dokumen dengan membuang imbuhan, baik awalan maupun akhiran (Prihatini, 2016). Dengan dilakukanya proses stemming setiap kata berimbuhan akan berubah menjadi kata dasar, dengan demikian dapat lebih mengoptimalkan proses teks mining (Fauzi, 2016).

2.3.3 Pembobotan Kata

2.3.3.1 TF-IDF

Riyanto (2016) di dalam Juniarsih et al. (2020) mengatakan metode *term frequency - inverse document frequency* (TF-IDF) merupakan suatu cara untuk memperoleh pembobotan berdasarkan jumlah kemunculan suatu kata (*term*) dalam sebuah dokumen dan dalam keseluruhan dokumen. Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval* (Melita et al, 2018). Konsep dasar yang digunakan pada

information retrieval adalah mengukur kesamaan antara dua dokumen, akan dilihat sejauh mana kemiripan yang ada pada dokumen tersebut (Wiyanto et al, 2019). Metode TF-IDF menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut (Wahyuni et al, 2017). *Term frequency* dan *inverse document frequency* saling bertolak belakang karena *term frequency* bobot terbesar diberikan kepada *term* atau kata yang kemunculannya paling sering, sedangkan untuk *inverse document frequency* bobot terbesar dimiliki oleh *term* atau kata yang frekuensi kemunculannya kecil.

Term frequency (TF) akan memperlihatkan seberapa banyak atau sering term atau kata muncul dalam sebuah dokumen. *Term frequency* untuk menghitung *term frequency* akan didapatkan dari hasil persamaan (1).

$$tf = \frac{jumlah\ frekuensi\ kata}{panjang\ dokumen} \qquad (1)$$

Inverse document frequency (IDF) menunjukkan seberapa jarang sebuah *term* atau kata muncul di dalam seluruh dokumen. Kata yang jarang muncul berfungsi untuk membedakan satu dokumen dengan dokumen yang lainnya (Ailiyya, 2020). Hasil perhitungan IDF didapatkan melalui persamaan (2).

$$IDF_j = log\left(\frac{D}{df_j}\right) \qquad (2)$$

Keterangan:

D = Jumlah semua dokumen

df_j = Jumlah dokumen yang mengandung *term* atau kata (tj)

Untuk mendapatkan hasil pembobotan dari metode TF-IDF maka dari itu terdapat penggabungan dari persamaan hitung TF dan IDF yaitu dengan mengalikan kedua persamaan tersebut yang dijabarkan pada persamaan (3).

$$w_{ij} = tf_{ij}\ x\ IDF_j$$

$$w_{ij} = tf_{ij}\ x\ log\ log\left(\frac{D}{df_j}\right)$$

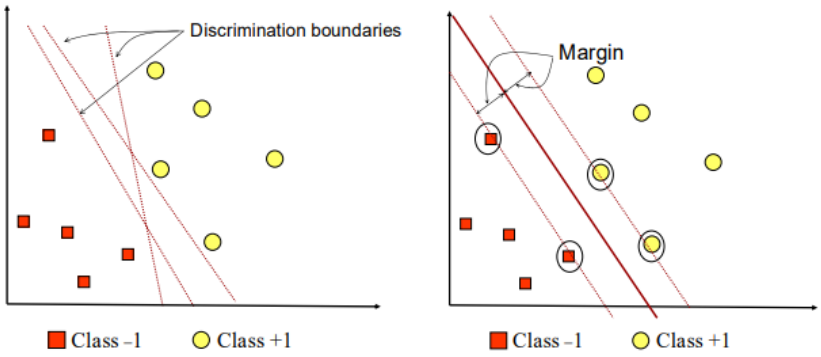
Keterangan:

w_{ij} = Bobot *term* atau kata ij terhadap dokumen di

Ailiyya (2020) menyatakan bahwa nilai TF-IDF tertinggi adalah saat suatu kata *t* muncul berkali-kali dalam jumlah dokumen yang sedikit sedangkan TF-IDF menjadi lebih rendah apabila suatu kata *t* muncul lebih sedikit dalam satu dokumen, atau dalam banyak dokumen.

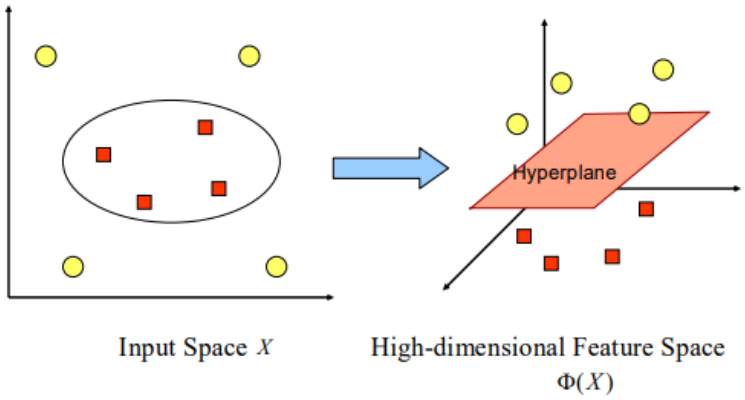
2.3.4 Klasifikasi *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah algoritma yang dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan di *Annual Workshop on Computational Learning Theory* pada tahun 1992. SVM merupakan salah satu metode klasifikasi dengan menggunakan metode machine learning (supervised learning) yang memprediksi kelas berdasarkan pola dari hasil proses training yang diciptakan oleh Vladimir Vapnik (Haranto dan Sari, 2019). Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane-hyperplane* terbaik yang berfungsi sebagai pemisah dua buah class pada *input space*. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya (Nugroho et al, 2003). Pada **Gambar 2.1** Margin adalah jarak antara *hyperplane* dengan pattern terdekat dari masing-masing *class* dan pattern yang terdekat disebut sebagai *support vector*.



Gambar 2. 1 Margin pada SVM

Pada **Gambar 2.2** Fungsi *kernel* pada SVM memungkinkan untuk mengimplementasikan suatu model pada ruang dimensi lebih tinggi (ruang fitur) tanpa harus mendefinisikan fungsi pemetaan dari ruang input ke ruang fitur.



Gambar 2. 2 Fungsi *Kernel*

Menurut Anjasmoros et al. (2020) *support vector machine* memiliki 4 (empat) *kernel trick* sebagai berikut.

a. *Kernel Linear*

Kernel linear adalah fungsi kernel yang sederhana dengan mengalikan titik dari dua vektor. *Linear kernel* digunakan ketika data yang dianalisis sudah terpisah secara *linear*. *Linear kernel* cocok ketika terdapat banyak fitur dikarenakan pemetaan ke ruang dimensi yang lebih tinggi tidak benar – benar meningkatkan kinerja seperti pada klasifikasi teks (Ningrum, 2018). Persamaan *kernel linear* dapat dilihat pada persamaan (4).

$$K(\mathrm{x},\mathrm{y}) = \mathrm{x}^T\mathrm{y} \qquad (4)$$

b. *Kernel* Polinomial

Kernel polinomial adalah fungsi *kernel* yang digunakan pada saat data tidak terpisah secara *linear*. Polinomial *kernel* sangat cocok untuk permasalahan dimana semua training dataset dinormalisasi (Ningrum, 2018). Persamaan *kernel* polinomial dapat dilihat pada persamaan (5).

$$K(\mathrm{x},\mathrm{y}) = (\mathrm{x}^T\mathrm{z})^d \; atau \; (1+\mathrm{x}^T\mathrm{z})^d \qquad (5)$$

Keterangan:

d = *Degree*

c. *Kernel Sigmoid*

Kernel Sigmoid (*The Hyperbolic Tangent Kernel*) adalah *Multilayer*

Perceptron (MLP) *kernel*. *Kernel Sigmoid* berasal dari *Neural Networks*, dimana fungsi *bipolar sigmoid* yang sering digunakan sebagai *activation function* atau *artificial neurons* (Cesarsouza, 2010).

$$K(\mathrm{x},\mathrm{y}) = (\alpha \mathrm{x}^{T}\mathrm{y} + c) \qquad (6)$$

Dimana $\frac{1}{(1+expexp\,(\alpha))}$

Keterangan:

c = *Constanta*

α = *Alpha*

d. *Kernel RBF* (*Radial Basis Function*)

Menurut Patel (2017) di dalam (Ningrum, 2018) *Kernel RBF* (*Radial Basis Function*) *RBF kernel* memiliki dua parameter yaitu *Gamma* dan *Cost*. Parameter *cost* atau biasa disebut sebagai C merupakan parameter yang bekerja sebagai pengoptimalan SVM untuk menghindari misklasifikasi disetiap sampel dalam *training dataset*. Parameter *gamma* menentukan seberapa jauh pengaruh dari satu sampel *training dataset* dengan nilai rendah berarti "jauh", dan nilai tinggi berarti "dekat". Dengan *gamma* yang rendah, titik yang berada jauh dari garis pemisah yang masuk akal dipertimbangkan dalam perhitungan untuk garis pemisah. Ketika *gamma* tinggi berarti titik – titik berada di sekitar garis myang masuk akal akan dipertimbangkan dalam perhitungan. Kusumaningrum (2017) menjelaskan bahwa pengaturan parameter C dan gamma dilakukan pada data *training* dimana nilai parameter ditentukan peneliti. Prangga (2017) menjelaskan bahwa dengan mengkombinasikan parameter C dan *Gamma* kita dapat menghasilkan nilai akurasi maksimum adapun dari penelitian tersebut didapatkan nilai 1, 10, 100 untuk C, sedangkan parameter Gamma sebesar 0,005; 0,05; 0,1; 0,5; 0,75. Dari rentang parameter penelitian Agustina et al (2020) dihasilkan parameter terbaik C sebesar 10 dan *gamma* sebesar 0,1 dengan rentang nilai C sebesar 0,01;0,1;1;10;100;1000;10000 dan parameter *gamma* 0,001;0,01;0,1;1;10;100;1000. Prangga (2017) menyatakan bahwa peneliti dapat melakukan pemilihan nilai-nilai parameter yang

memungkinkan untuk digunakan dalam penelitian. Persamaan *kernel* RBF dapat dilihat pada persamaan (7).

$$K(x,y) = exp\left(-\frac{||x-y||^2}{2\sigma^2}\right) \qquad (7)$$

Keterangan:

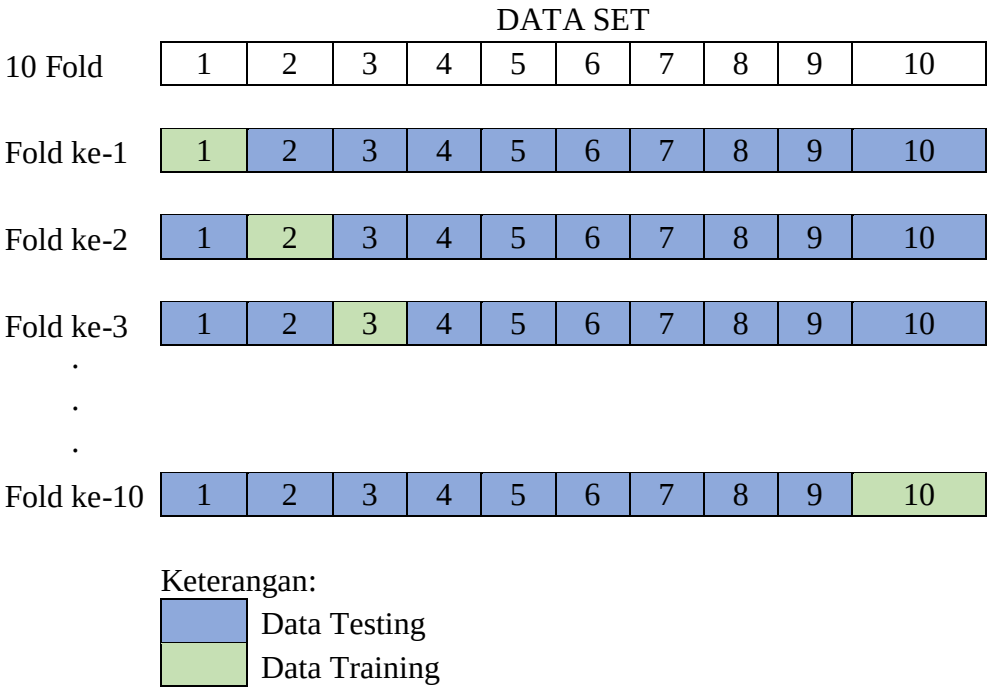
σ = sigma

Exp = eksponensial

2.3.5 Pengujian Model

2.3.5.1 K-Fold Cross Validation

Menurut (Widjaya, 2017) *k-fold cross validation* adalah metode validasi dengan membagi data ke dalam k-subset, kemudian melakukan pengulangan sebanyak k kali untuk pembelajaran dan pengujian. Pada setiap pengulangan, digunakan satu subset sebagai data uji dan subset lainnya sebagai data pembelajaran seperti yang diilustrasikan dalam **Gambar 2.3**. *K-fold cross validation* digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem atau model.



Gambar 2. 3 Percobaan Validasi *K-Fold Cross Validation*

Kinerja dari *k-fold cross validation* menurut (Hulu, 2020), yaitu:

1. Total *instance* dibagi menjadi N bagian.
2. Fold ke-1 adalah ketika bagian ke-1 menjadi data uji (testing data) dan sisanya menjadi data latih (training data). Selanjutnya, hitung akurasi atau kesamaan atau kedekatan suatu hasil pengukuran dengan angka atau data yang sebenarnya berdasarkan porsi data tersebut. Persamaan untuk menghitung nilai akurasi sebagai berikut.

$$Akurasi = \frac{\Sigma\ data\ uji\ benar\ klasifikasi}{\Sigma\ total\ data\ uji}\ x\ 100 \qquad (8)$$

3. Fold ke-2 adalah ketika bagian ke-2 menjadi data uji (testing data) dan sisanya menjadi data latih (training data). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.
4. Demikian seterusnya hingga mencapai fold ke-k. Hitung rata-rata akurasi dari k buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final.

Adapun untuk mengevaluasi performa dari beberapa model data set pada metode *K-Fold Cross Validation* diperlukan perhitungan untuk mengukur seberapa baik hasil prediksi sesuai dengan data yang telah dikumpulkan, yaitu menggunakan *mean squared error* (MSE).

$$MSE = \left(\frac{1}{n}\right) x\Sigma(y_i - f(x_i))^2 \quad (9)$$

Menurut Widyaningsih et al (2021) Prosedur ini akan diulang sebanyak k kali sampai semua bagian (fold) menjadi data testing. Pada setiap iterasi, mean square of error akan dihitung sehingga kita akan memiliki nilai dari $MSE1$, $MSE2$, … , $MSEk$. Untuk menentukan model mana yang terbaik, akan ditinjau berdasarkan performance metric model yaitu melalui rata-rata MSE yang dihasilkan pada setiap iterasi. Model terbaik merupakan model yang memiliki rata-rata nilai MSE yang terkecil. Nilai MSE yang rendah atau mendekati nol menunjukkan bahwa hasil prediksi sesuai dengan data aktual.

$$CV_{(k)} = \left(\frac{1}{k}\right) {}_{i=1}^{k}\Sigma\ MSE_I \quad (10)$$

2.3.5.2 Confusion Matrix

Menurut C.O. a. P. Ti (2014) di dalam Sa'adah (2020) Confusion Matrix

adalah sebuah alat untuk melakukan analisis yang biasanya digunakan dalam *Supervised Learning* yang digunakan untuk melihat hasil tes dari model yang telah diprediksi. Indirani (2014) di dalam Sajid (2021) menyebutkan matrix menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. *Confusion matrix* ini digunakan untuk menghitung nilai *accuracy*, *precision*, dan *recall*. Tabel *confusion matrix* dapat dilihat sebagai berikut (Sa'adah, 2020).

Tabel 2. 1 Tabel *Confusion Matrix*

		PREDICTION VALUES		
		TRUE	FALSE	NEUTRAL
ACTUAL VALUES	TRUE	TP	FPosNeg	FPosNet
	FALSE	FNegPos	TNeg	FNegNet
	NEUTRAL	FNetPos	FNetneg	TNet

Keterangan:

TP (*True Positive*) = Jumlah data positif terdeteksi benar
TNeg (*True Negative*) = Jumlah data negatif terdeteksi benar
TNet (*True Neutral*) = Jumlah data netral terdeteksi benar
FNegPos (*False Negative Positive*) = Jumlah data positif terdeteksi salah
FPosNeg (*False Positive Negative*) = Jumlah data negatif terdeteksi salah
FNetPos (*False Neutral Positive*) = Jumlah data netral terdeteksi benar
FPNet (*False Positive Netral*) = Jumlah data positif terdeteksi netral
FNegNet (*False Negative Neutral*)= Jumlah data negatif terdeteksi netral
FNetNeg (*False Neutral Negative*)= Jumlah data netral terdeteksi negatif

Dari hasil *confusion matrix* maka dapat dihitung nilai *accuracy*, *precision*, *recall* dari hasil klasifikasi tersebut. Sebagaimana perhitungan menurut Iskandar dan Suprapto (2015) dijabarkan berikut ini.

a. *Accuracy*

   Accuracy adalah nilai persentase pendekatan dari total data yang diidentifikasi dan dinilai. Perhitungan untuk *accuracy* dilakukan dengan persamaan rumus (9).

$$Accuracy = \left(\frac{TP + TNeg + TNet}{TP + FPosNeg + FPosNet + TNeg + FNegPost + FNegNet + TNet + FNetPos + FNetNeg}\right) \quad (9)$$

b. *Precision*

Precision adalah tingkat ketepatan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Perhitungan untuk *precision* dilakukan dengan persamaan rumus (10).

$$Presisi = \left(\frac{\frac{TP}{TP + FPosNet + FPosNeg} + \frac{TNeg}{TNeg + FNegNet + FNegPos} + \frac{TNet}{TNet + FNetPos + TNetNeg}}{3}\right) \quad (10)$$

c. *Recall*

Recall adalah nilai keberhasilan model dalam menemukan kembali sebuah informasi. Perhitungan untuk *recall* dilakukan dengan persamaan rumus (11).

$$Recall = \left(\frac{\frac{TP}{TP + FNegPos + FNetPos} + \frac{TNeg}{FPosNeg + TNeg + FNetNeg} + \frac{TNet}{TNet + FNegNet + FPosNet}}{3}\right) \quad (11)$$

d. *Specificity*

Specificity adalah kemampuan suatu tes untuk menyatakan negatif pada data positif dan netral. *Recall* dari kelas negatif adalah nilai spesifisitas. Nilai spesifisitas digunakan sebagai dasar perhitungan nilai AUC.

$$Specificity = \frac{TNeg}{TNeg + FPosNeg + FNetNeg} x100\%$$

e. *False Positive Rate* (FPR)

False Positive Rate (FPR) adalah proporsi data yang prediksinya adalah positif terhadap seluruh data negatif dan netral

$$FPR = 1 - Specificity$$

f. *Area Under Curve* (AUC)

Nilai *Area Under Curve* adalah nilai yang digunakan untuk mengukur kinerja deskriminatif menggunakan probabilitas hasil dari

sampel. Rentang nilai AUC yaitu dari 0 hingga 1. Semakin tinggi nilai AUC maka klasifikasi akan dinyatakan baik.

$$Nilai\ AUC = \frac{1 + recall - FPR}{2}$$

2.4 Bahasa Pemrograman *Python*

Menurut Sanner (1999) *Python* adalah bahasa pemrograman yang ditafsirkan, interaktif, dan *object-oriented*. *Python* menyediakan struktur data tingkat tinggi seperti *list* dan *associative arrays* (kamus), *typing* dan *binding* yang *dynamic*, *modules*, *classes*, manajemen penyimpanan otomatis, dan lainnya. Python merupakan bahasa pemrograman tingkat tinggi karena kode secara otomatis akan dikompilasi ke kode byte dan dieksekusi. *Python* cocok digunakan sebagai *scripting language*, implementasi bahasa pemrograman web, dan lainnya. *Python* dapat di *extended* ke dalam bahasa C dan C++ dan memberikan kecepatan yang memadai dalam mengkomputasi suatu pekerjaan komputasi yang intensif (Khulman, 2012).

2.5 Twitter

Twitter merupakan salah satu media sosial yang sangat digemari hingga sekarang. Twitter adalah sebuah situs *web* yang dimiliki dan dioperasikan oleh Twitter Inc., yang menawarkan jaringan sosial berupa *microblog* sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan yang disebut kicauan (*tweets*). Kicauan adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna (Hanafi, 2011). Twitter menghubungkan banyak orang dengan memudahkan pengguna untuk menjalin pertemanan bersama pengguna Twitter lainnya. Media sosial twitter digunakan selain untuk mencari suatu informasi, juga digunakan untuk memberikan suatu informasi baik itu mengenai hal yang umum sampai mengenai hal yang pribadi atau menyampaikan informasi mengenai data identitas pribadi (Basri, 2017).

2.6 *Marketplace*

Menurut Apriadi dan Saputra (2017) *Marketplace* merupakan sebuah wadah pemasaran produk secara elektronik yang mempertemukan banyak penjual dan pembeli untuk saling bertransaksi. Marketplace ini dibuat untuk mengurangi resiko proses bisnis yang kompleks sehingga tercipta efisiensi dan efektifitas (Ramadhan, 2021). Dapat disimpulkan bahwa *marketplace* adalah tempat yang mempertemukan penjual dan pembeli dengan konsep virtual agar proses bisnis dapat dilakukan dengan mudah.

2.6.1 Shopee

Shopee merupakan perusahaan *marketplace* yang menyediakan layanan penjualan melalui website dan aplikasi. Shopee menawarkan berbagai produk mulai dari pakaian, barang elektronik, makanan, perlengkapan rumah, dan lainnya. Shopee diluncurkan pertama kali oleh Forrest Li pada tahun 2015 di Singapura, Indonesia, Malaysia, Thailand, Taiwan, Vietnam, dan Filipina.

Gambar 2. 4 Logo Perusahaan Shopee

Dengan mengusung konsep belanja *online* yang mudah diakses, gampang, dan menyenangkan, Shopee menawarkan berbagai layanan yang mencakup sebagai wadah penjual dan pembeli dalam melakukan transaksi, menyediakan fitur pesan langsung sesama konsumen maupun penjual, memberikan kemudahan dengan metode pembayaran yang aman dan terdigitalisasi, serta layanan pengiriman yang terintegrasi dengan jasa pengiriman ekspedisi pihak ketiga.

2.6.2 Tokopedia

Tokopedia atau PT. Tokopedia adalah perusahaan *marketplace* karya anak bangsa Indonesia melalui *website* dan aplikasi. Tokopedia didirikan oleh William

Tanuwijaya dan Leontinus Alpha Edison pada tahun 2009. Mengambil permasalahan yang ada di Indonesia yakni banyaknya pulau di Indonesia dan keterbatasan akses untuk memenuhi kebutuhan, kedua pendirinya berpikir untuk mengatasi masalah tersebut dengan meluncurkan Tokopedia. Misi Tokopedia sendiri yakni mencapai pemerataan ekonomi secara digital untuk Indonesia.

Gambar 2. 5 Logo Perusahaan Tokopedia

Sama halnya dengan konsep *marketplace*, Tokopedia menyediakan layanan bagi penjual dan pembeli dalam melakukan transaksi digital. Tokopedia sendiri mendorong pelaku UMKM di Indonesia untuk berkontribusi dalam mendukung terpenuhinya kebutuhan masyarakat di Indonesia melalui *platform* digital. Menurut Tokopedia.com berbagai layanan diberikan oleh Tokopedia yaitu layanan yang gratis diberikan oleh Tokopedia artinya tidak ada biaya untuk memulai bisnis di *marketplace* Tokopedia, jangkauan penjual dan pembeli yang luas dengan mencakup 99% kecamatan di seluruh Indonesia, layanan logistik yang lengkap dengan opsi 13 mitra logistik dan dapat dipilih oleh penjual, serta banyak fitur pendukung yakni produk teknologi finansial dan fitur pendukung lain untuk menunjang perkembangan toko penjual.

2.6.3 Event Belanja *Marketplace*

Menurut data Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pengguna internet di Indonesia pada tahun 2019 hingga 2020 quartal ke 2 sebanyak 73,7% atau 196,71 juta jiwa dari total populasi 266,91 juta jiwa. Dengan banyaknya pengguna internet di Indonesia maka perusahaan-perusahaan *marketplace* harus mempersiapkan berbagai inovasi yaitu menerapkan strategi pemasaran untuk menarik minat dan mempertahankan pengguna, meningkatkan pangsa pasar, dan menunjang persaingan antar *marketplace*. Salah satu strategi pemasaran yang

diterapkan *marketplace* saat ini adalah mengadakan *event-event* promo spesial. Berbagai *event* belanja ini dijadikan sebagai ciri khas masing-masing *marketplace* dan dilakukan pada setiap tanggal cantik maupun hari perayaan.

Shopee adalah *marketplace* yang menerapkan strategi pemasaran dengan mengadakan berbagai *event* promo. Adapun *event* promo yang dilakukan Shopee adalah *Super Brand Day*, *Cashback Day*, Gratis Ongkir, *Flash Sale*, Merdeka *Sale*, Gajian *Sale*, *Event* Tanggal Kembar, 11.11 *Big Sale*, Shopee 12.12 *Birthday Sale*. Selain itu, Tokopedia juga merupakan *marketplace* di Indonesia yang menerapkan strategi pemasaran dengan mengadakan *event* promo. Berbagai *event* promo dilakukan oleh Tokopedia yaitu Kejar Diskon, Bebas Ongkir, Festival Kreasi, *Super Gadget Day*, Selalu *Official Store*, dan yang paling utama adalah *event* Waktu Indonesia Belanja (WIB) yang diadakan setiap tanggal 25 hingga akhir bulan.