BAB 2
LANDASAN TEORI

Berikut adalah teori dan definisi yang menjelaskan setiap aspek yang terlibat pada penelitian ini.

2.1 Aplikasi JMO

Aplikasi JMO (Jamsostek Mobile) adalah aplikasi resmi dari BPJS Ketenagakerjaan yang diluncurkan pada tahun 2021. Aplikasi ini menggantikan aplikasi BPJSTKU yang dirilis pada tahun 2018. Aplikasi JMO dirancang untuk memudahkan peserta dalam mengakses informasi dan layanan program jaminan sosial ketenagakerjaan. Dengan adanya aplikasi ini, peserta tidak perlu lagi datang ke kantor cabang BPJS Ketenagakerjaan untuk mendapatkan informasi atau layanan. Aplikasi JMO dapat diakses melalui *Google Play* dan *App Store*. Aplikasi ini memiliki berbagai fitur, seperti informasi lengkap tentang program jaminan BPJS Kesehatan, dan BPJS Ketenagakerjaan, layanan cek saldo Jaminan Hari Tua (JHT), Jaminan Kecelakaan Kerja (JKK), dan Jaminan Kematian (JKM), layanan klaim JHT, JKK, dan JKM, layanan pelaporan kecelakaan kerja, layanan informasi tentang kantor cabang BPJS Ketenagakerjaan di seluruh Indonesia [9]. Aplikasi JMO juga memiliki beberapa keunggulan, antara lain menghemat waktu, efisien, serta akses informasi yang mudah.

2.2 Support Vector Machine

Support Vector Machine (SVM) adalah algoritma klasifikasi yang dikembangkan pada pertengahan 1990-an. *Support Vector Machine* memiliki dasar teori yang kuat dan mampu mencapai kinerja yang lebih unggul daripada metode tradisional dalam sejumlah besar aplikasi, termasuk pengenalan angka tulisan tangan, klasifikasi teks, dan lain-lain. SVM dapat digunakan untuk regresi tiga jenis hasil yang berbeda: Bernoulli (biner), multinomial, atau kontinu. Regresi dengan hasil Bernoulli sering disebut sebagai klasifikasi dalam pembelajaran statistik. Regresi dengan hasil multinomial disebut klasifikasi multi kelas, sedangkan regresi dengan hasil yang berkelanjutan disebut regresi [10].

2.2.1 Klasifikasi Support Vector Machine

SVM pada awalnya dirancang untuk klasifikasi dua kelas. Pendekatan ini kemudian diperluas untuk mengatasi hasil berkelanjutan dan klasifikasi dengan lebih dari dua kelas. Gambar 2.1 memberikan contoh dengan dua variabel. Hasilnya, keanggotaan kelas (y = 1 or y = −1), hal ini ditunjukkan dengan simbol plot. Hal ini dapat menggambar banyak garis yang benar-benar memisahkan kedua kelas tersebut, dua di antaranya ditunjukkan pada gambar 2.1. Untuk menentukan hasil terbaik dilakukanlah pemilihan garis pemisah sedemikian rupa sehingga margin jarak minimum antara garis dan titik terdekat dapat dimaksimalkan. Hal ini menyebabkan masalah optimasi berikut:

$$\underset{\beta j, j=1,\ldots,p}{\text{maximize } M} \tag{2.1}$$

$$\text{Subject to } \left\{ \begin{array}{l} \sum_{j=1}^{p} B_j^2 = 1 \\ y_i(B_0 + B_1 x_{i1} + \ldots B_p x_{ip}) \geq M \end{array} \right\} \tag{2.2}$$

Dimana M adalah margin, p adalah banyaknya variabel x, dan $B_p \in \{-1,1\}$,i=1,...,n. Baris kedua pada gambar 2.2 mewakili sebuah hyperplane. Dalam dua dimensi, seperti pada gambar 2.1, bidang hiper berbentuk garis.

Pengamatan diatas *hyperplane*:

$$(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip}) > 0 \tag{2.3}$$

Pengamatan dibawah *hyperplane*:

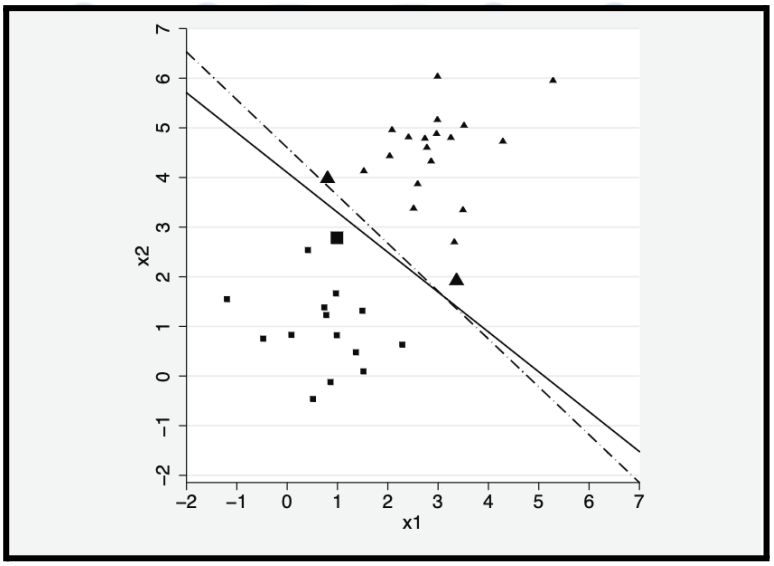
$$(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip}) < 0 \tag{2.4}$$

Pengamatan dibawah hyperplane dengan label kelas −1:

$$y(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip}) \tag{2.5}$$

Hasil selalu positif selama kedua kelas tersebut dapat dipisahkan. Kemudahan matematis ini menjadi alasan mengapa label kelas lebih $\{-1,1\}$ dibandingkan label biasa, $\{0,1\}$, yang digunakan dalam regresi logistik. Batasan pada β_j (2) tidak diperlukan tetapi memiliki keuntungan bahwa jarak pengamatan apapun ke hyperplane sama dengan

$$y(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip}) \tag{2.5}$$



Gambar 2.1. Klasifikasi Linear Dua Dimensi
Sumber: [10]

Dua kelas observasi yang dapat dipisahkan dengan garis pemisah margin maksimum (*solid*) dan garis pemisah kedua (*dashed*). Segitiga dan kotak menunjukkan dua kelas (nilai y). Segitiga dan kotak yang lebih besar menunjukkan vektor pendukung. Masalah optimasi pada gambar 2.1 dan gambar 2.2 dapat diselesaikan dengan program pemecah kuadrat. Solusinya bergantung pada pengamatan yang terletak pada margin. Hal ini sangat mengurangi komputasi karena hanya sebagian kecil data yang diperlukan untuk menghitung solusi optimal. Sub kumpulan observasi tersebut disebut vektor pendukung. Contoh pada gambar 2.1 memiliki tiga vektor pendukung.

Pada sebagian besar aplikasi, kedua kelas tidak dapat dipisahkan, dan masalah optimasi tidak mempunyai solusi. Namun hal ini dapat diatasi dengan memberikan ruangan terhadap sejumlah kesalahan:

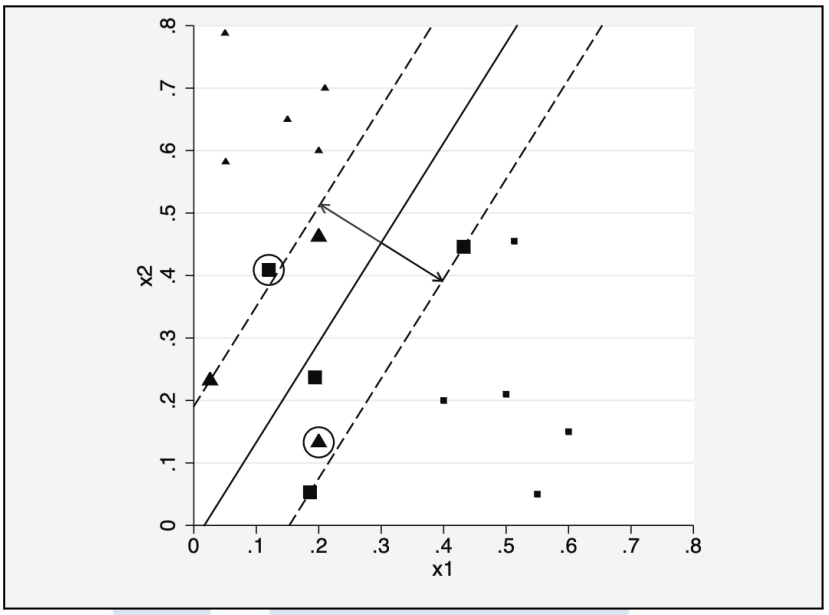
$$\underset{\beta ij\ ,\ i,i=1,\ldots,\ n;\ j=1,\ldots,p}{maximize\ M} \tag{2.6}$$

$$\text{Subject to} \left\{ \begin{array}{c} \sum_{j=1}^{p} \beta_j^2 = 1 \\ y_i(\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}) \geq M\,(1-\epsilon_i) \\ \epsilon_i \geq 0;\ \sum_{i=1}^{n} \epsilon_i \leq \end{array} \right\} \tag{2.7}$$

Variabel slack, i, memungkinkan observasi berada pada sisi margin yang salah. Namun, jumlah kesalahan tersebut tidak boleh melebihi anggaran kesalahan total, C. C ditentukan dengan meneruskan c (#) ke perintah mesin.

Solusi untuk masalah optimasi hanya bergantung pada vektor pendukung pengamatan yang terletak pada margin atau pengamatan yang melanggar margin. Pada contoh gambar 2.2. Garis *solid* memisahkan sebagian besar segitiga dan persegi, kecuali satu persegi di atas garis dan satu segitiga di bawah garis (keduanya dilingkari). Garis putus-putus atas adalah margin segitiga, dan garis putus-putus bawah adalah margin segitiga. kotak. Seperti sebelumnya, ada beberapa observasi yang langsung berada di margin. Berbeda dengan sebelumnya, ada juga observasi yang melanggar margin. Pengamatan yang melanggar batas tidak akan salah diklasifikasikan sepanjang pengamatan tersebut berada pada sisi batas keputusan yang benar. Perhatikan salah satu pengamatan yang salah

klasifikasi, sebuah persegi, muncul di luar dua margin. Meskipun demikian, ini merupakan vektor dukungan karena melanggar margin bawah. Pelanggarannya sangat besar sehingga melampaui margin atas.



Gambar 2.2 Contoh *Non separable class*
Sumber: [10]

Garis *solid* mewakili batas keputusan, sedangkan garis putus-putus mewakili margin atas dan bawah. Segitiga dan kotak menunjukkan dua kelas (nilai y). Segitiga dan kotak yang lebih besar menunjukkan vektor pendukung. Pengamatan yang salah klasifikasi pada sisi yang salah dari batas keputusan dilingkari. Nilai C yang besar menambah hukuman atas kesalahan klasifikasi pada data pelatihan. Hal ini berkaitan dengan margin yang lebih ketat. Nilai C yang besar juga berhubungan dengan vektor pendukung yang lebih sedikit karena lebih sedikit observasi yang salah diklasifikasikan dan karena lebih sedikit observasi berada dalam margin yang lebih sempit atau dapat disebut juga dengan *soft margin* [11].

2.2.2 Klasifikasi *Multiclass*

Pada pengklasifikasikan lebih dari dua kelas, SVM dapat menggunakan dua teknik multi class, yaitu One Versus One (OVO) dan One Versus All (OVA). OVO membandingkan satu kelas dengan semua kelas lainnya, sedangkan OVA membandingkan satu kelas dengan gabungan semua kelas lainnya. SVM menggunakan teknik multi class karena pada dasarnya hanya dapat mengklasifikasikan dua kelas secara linear akurasi pada OVA lebih tinggi daripada OVO, meskipun OVO lebih cepat dari segi waktu pemrosesan.

Tabel 2.1 *Multiclass One Versus All (OVA)*

$h_i = 1$	$h_i = -1$	Hipotesis
Kelas 1	Bukan Kelas 1	$f^1(g) = (w^1)g + b^1$
Kelas 2	Bukan Kelas 2	$f^3(g) = (w^3)g + b^3$
Kelas 3	Bukan Kelas 3	$f^3(g) = (w^3)g + b^3$
Kelas 4	Bukan Kelas 4	$f^4(g) = (w^4)g + b^4$

Dalam menerapkan metode OVA akan dibangun z buah model SVM biner. z disini adalah jumlah kelas. Dalam mengklasifikasikan hal tersebut dapat dilihat pada persamaan berikut:

$$Kelas\ g = arg \max_{r=1..z} ((w^{(r)})^T . \varphi [\ {pi \atop qi}\ ] + b^{(r)}) \quad (2.8)$$

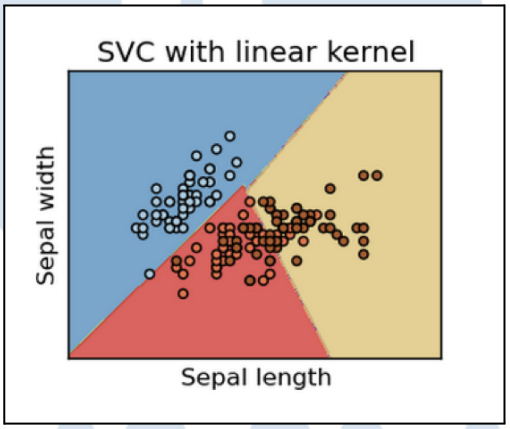
Persamaan tersebut digunakan pada tahap terakhir proses testing setelah support vector dari data testing didapatkan. $(w^{(r)})^T$ adalah hyperplane yang jumlahnya sebanyak z atau jumlah kelas dari data training. Hyperplane merupakan model atau pembatas kelas dari hasil perhitungan training SVM. Selain itu, ada juga hasil perhitungan training SVM lainnya $b^{(r)}$ atau bias. Support vector data testing $\varphi [\ {pi \atop qi}\ ]$ dimasukan ke dalam persamaan sebagai komponen dari data testing yang akan diklasifikasikan. Hasil klasifikasi ditentukan dengan arg max bahwa nilai tertinggilah yang akan diambil dari hasil perhitungan semua hyperplane dengan support vector data testing sebagai kelas targetnya atau juga bisa disebut sebagai kelas prediksi.

2.2.3 Kernel Linear

Dalam pemodelan *Support Vector Machine kernel linear* merupakan fungsi sederhana yang dapat digunakan untuk menganalisis data yang terpisah secara *linear*. Kernel dapat dimanfaatkan ketika terdapat banyak fitur dalam pemetaan ke dalam ruang dimensi yang lebih tinggi kernel ini tidak benar benar meningkatkan kinerja [12]. Persamaan fungsi untuk *kernel linear* dapat dilihat pada persamaan 2.8.

$$K(x,xi) = sum(x*xi) \tag{2.9}$$

Pada persamaan *linear x* merupakan titik data yang dimasukkan ke dalam SVM. Dapat dilihat pada Gambar 2.1 garis pemisah kernel linear berbentuk garis lurus.



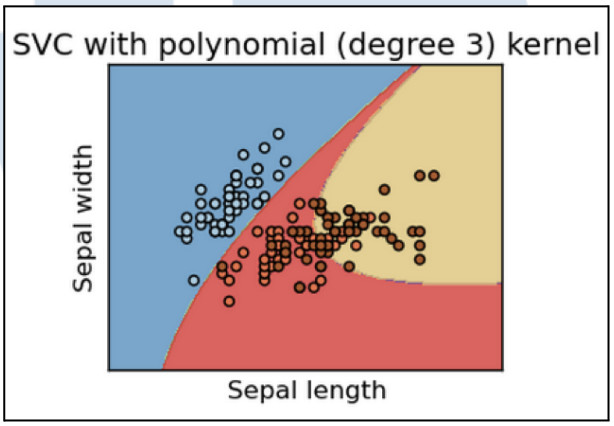
Gambar 2.3. Kernel Linear

Sumber: [13]

2.2.4 *Kernel Polynomial*

Kernel polynomial digunakan untuk mengklasifikasi data yang tidak dapat dipisahkan secara linear. Dalam machine learning, kernel polynomial adalah fungsi yang digunakan dalam metode SVM dan kernelisasi lainnya untuk mengukur kesamaan vektor data latih dalam ruang fitur [14]. *Kernel polynomial* juga dapat digunakan untuk memecahkan masalah klasifikasi dataset pelatihan yang telah dinormalisasi. Persamaan terkait *kernel polynomial* dapat dilihat pada persamaan 2.9.

$$K(x,xi) = \exp\,(1+x_i^T x_j)^p \tag{2.10}$$



Gambar 2.4. Kernel Polynomial

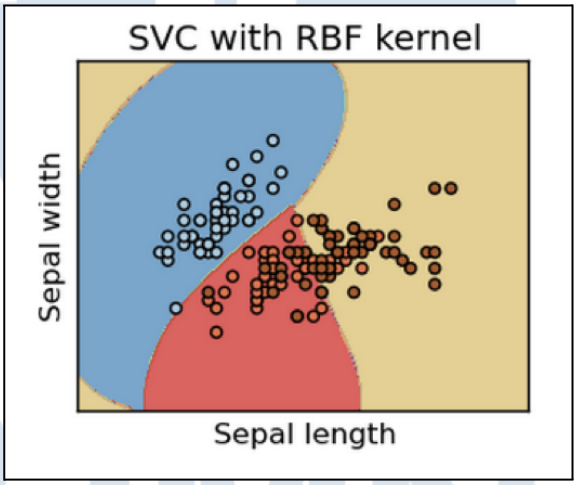
Sumber: [13]

2.2.5 Kernel RBF

Kernel RBF adalah kernel yang digunakan untuk mengklasifikasi data *non-linier*. Kernel RBF juga dikenal sebagai *kernel Gaussian*. Kernel RBF dapat memberikan performa yang baik dengan parameter yang tepat, dan juga menghasilkan nilai error yang lebih kecil daripada kernel lain untuk data latih [15]. Persamaan kernel RBF adalah sebagai pada persamaan 2.10.

$$K(x,xi) = exp(-gamma*sum((x-xi2)) \quad (2.11)$$

Parameter gamma dalam persamaan RBF menentukan seberapa besar pengaruh dari data latih terhadap keputusan klasifikasi. Hal ini dapat dilihat pada Gambar 3, dimana pemisah berbentuk *non-linier* dan menyerupai peta.



Gambar 2.5. Kernel RBF

Sumber: [2]

2.3 Analisa Sentimen

Analisis sentimen merupakan proses untuk mengetahui sentimen yang terkandung dalam suatu kalimat dengan cara mengekstrak, mengolah, dan memahami data teks secara otomatis [6]. Analisis sentimen adalah proses komputasi untuk mengidentifikasi dan mengukur sentimen dalam teks. Analisis sentimen dibagi menjadi dua jenis, yaitu klasifikasi subjektivitas dan klasifikasi polaritas. Klasifikasi subjektivitas menentukan apakah teks tersebut merupakan pendapat atau fakta. Klasifikasi polaritas menentukan apakah teks tersebut memiliki sentimen positif, netral, atau negatif [16].

2.4 TD-IDF

Metode TF-IDF adalah metode yang digunakan untuk memberikan bobot pada kata-kata dalam suatu dokumen atau kalimat. Metode ini menggabungkan dua konsep, yaitu frekuensi kemunculan kata (*term frequency*) dan frekuensi dokumen yang mengandung kata tersebut (*inversed document frequency*) [17]. Frekuensi kemunculan kata dalam suatu dokumen menunjukkan seberapa penting kata tersebut untuk dokumen tersebut. Frekuensi dokumen yang mengandung suatu kata menunjukkan seberapa umum kata tersebut. Hal ini menyebabkan hubungan antara kata dan dokumen akan tinggi jika kata tersebut sering muncul dalam dokumen tersebut dan jarang muncul dalam dokumen lain. berikut merupakan rumus yang dapat digunakan untuk mengkalkulasi TD-IDF.

$$\text{Rumus IDF: } idf_t = \log\left(\frac{D}{dft}\right) \tag{2.12}$$

$$\text{Rumus TF: } tf(t,d) = \frac{tf}{max(tf)} \tag{2.13}$$

$$\text{Rumus TF: } W_{t,d} = tf(t,d) \times idf_t \tag{2.14}$$

Keterangan:

max(tf) = total kata yang ada pada suatu dokumen.

tf (t,d) = TF (Frekuensi term)

tf = jumlah kemunculan term terbanyak. pada dokumen yang sama.

idf(t) = bobot kemunculan term t di seluruh dokumen.

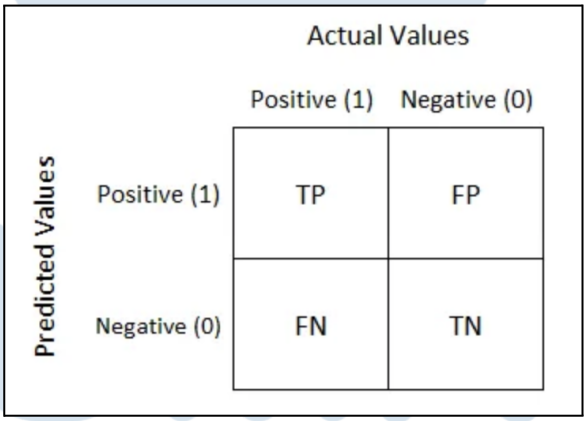
$W_{t,d}$ = bobot term dalam suatu dokumen

df_t = jumlah dokumen yang mengandung term t

D = total dokumen menyeluruh.

2.5 Confusion Matrix

Confusion matrix adalah pengukur performa model klasifikasi dengan keluaran 2 kelas atau lebih. *Confusion matrix* memiliki empat kombinasi hasil prediksi dan nilai aktual, yaitu *true positive*, *false positive*, *false negative*, dan *true negative* [18]. Pada Gambar 4 dapat dilihat contoh dari *confusion matrix*.



Gambar 2.4. *Confusion Matrix*

Sumber: [18]

Keterangan:

True Positive (TP) :1

Interpretasi: Anda memprediksi positif dan itu benar.

Contoh: Anda memprediksikan bahwa seorang wanita hamil dan wanita tsb memang benar hamil.

True Negative (TN):

Interpretasi: Anda memprediksi negatif dan itu benar.

Contoh: Anda memprediksikan bahwa seorang pria tidak hamil dan benar karena pria tidak mungkin hamil.

False Positive (FP):

Interpretasi: Anda memprediksi positif dan itu salah.

Contoh: Anda memprediksikan bahwa seorang pria hamil tetapi pria tidak mungkin hamil.

False Negative (FN):

Interpretasi: Anda memprediksi negatif dan itu salah.

Anda memperkirakan bahwa seorang wanita tidak hamil tetapi sebenarnya wanita tersebut hamil.

Untuk mengkalkulasi *confusion matrix* kita dapat memanfaatkan rumus yang digunakan untuk menghitung *precision, accuracy, recall dan F-measure*.

2.5.1. Precision

Precision adalah rumus yang dapat digunakan untuk menggambarkan akurasi antara data dengan hasil klasifikasi terhadap model. Contoh rumus yang dapat digunakan seperti pada persamaan 2.13.

$$Precision = \frac{TP}{TP+FP} \tag{2.15}$$

2.5.2. Accuracy

Accuracy dalam *confusion matrix* dapat digunakan untuk mengukur seberapa akurat model klasifikasi yang telah dirancang dengan baik. Contoh rumus yang dapat digunakan seperti pada persamaan 2.14.

$$accuracy = \frac{TP + TN}{TP+TN+FP+FP} \tag{2.16}$$

2.5.3. Recall

Recall dalam *confusion matrix* adalah ukuran keberhasilan model dalam menemukan kembali informasi. Contoh rumus yang dapat digunakan seperti pada:

$$Recall = \frac{TP}{TP+FN} \tag{2.17}$$

2.5.4. F-Measure

F-Measure dalam *confusion matrix* didapatkan dari perbandingan rata-rata antara hasil recall dan precision yang telah dibobotkan. Contoh rumus yang dapat digunakan seperti pada:

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall+Precision} \tag{2.18}$$

2.6 Preprocessing Data

Dataset akan diproses melalui tahap preprocessing. Tahap pertama adalah standarisasi, yaitu mengubah seluruh elemen menjadi huruf kecil. Tahap selanjutnya adalah tokenisasi, yaitu membagi data menjadi kata tunggal atau token. Setelah itu, kata-kata yang tidak diperlukan dalam analisis sentimen akan dihapus dalam tahap stop words. Tahap terakhir adalah normalisasi atau stemming, yaitu menghilangkan kata imbuhan menjadi kata dasar.

2.7. Pelabelan Data dan Pembobotan Kata

Setelah perancangan selesai dilaksanakan, tahapan selanjutnya yang dapat dilakukan adalah melakukan pembobotan dengan memanfaatkan TF-IDF, kemudian melakukan klasifikasi dengan model SVM.

2.8 Pengujian

Tahapan ini dilakukan untuk menguji sistem yang telah dibangun dengan melakukan pengujian terhadap code untuk menemukan *error* atau *bug*. Pengujian juga dilakukan untuk mengetahui apakah sistem sudah berjalan dengan sebagaimana mestinya. Dalam pengujian ini, dilakukan perbandingan kernel pada algoritma SVM dan pembagian data menjadi data training dan data testing dengan perbandingan 60/40, 70/30, dan 80/20. Hasil pengujian kemudian dianalisis menggunakan *confusion matrix* untuk menghitung tingkat *accuracy, precision, recall,* dan *f-measurement.*

2.9 Dokumentasi

Proses penelitian, perancangan, dan pembuatan aplikasi hingga selesai didokumentasikan secara bertahap, mulai dari pendahuluan hingga kesimpulan dan saran. Dokumentasi ini bermanfaat untuk memberikan informasi dan referensi bagi penelitian lainnya.