Klasifikasi Kepribadian Berdasarkan Data Twitter dengan Menggunakan Metode *Support Vector Machine*

Alvini Fikriani[1], Ibnu Asror [2], Yusza Reditya Murti [3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]alvinifikriani@students.telkomuniversity.ac.id, [2]iasror@telkomuniversity.ac.id, [3]yuszaa@telkomuniversity.ac.id

Abstrak

Jejaring sosial menjadi media yang banyak digunakan dan populer untuk penyebaran informasi serta fasilitator interaksi sosial. Interaksi pengguna memberikan wawasan berharga tentang karakteristik dan perilaku individu. Twitter merupakan layanan *microblogging*. Aktifitas utama twitter adalah mem-posting teks yang pendek (tweet) melalui web atau *mobile*.Kepribadian merupakan sebuah sikap yang unik terhadap seseorang dalam berperilaku. Untuk mengetahui kepribadian seseorang berdasarkan status-status (tweet) yang mereka tulis di Twitter dilakukan teknik klasifikasi teks menggunakan metode *Multiclass Support Vector Machine*. Bahasa yang digunakan adalah Bahasa Indonesia. Hasil pengujian dengan menggunakan *10 Fold Cross Validation* menunjukan Akurasi yang didapatkan dengan menggunakan strategi One-against-One dan One-against-All dan membandingkan penggunaan *dengan* dan tanpa *Symbol removing*. Hasil percobaan didapatkan menggunakan *One-against-One* dengan dan tanpa *Symbol removing* menghasilkan akurasi 86.55% dan 86.88%, sedangkan menggunakan *One-against-All* dengan dan tanpa *Symbol removing* 85.24% dan 85.07%.

Kata kunci : Kepribadian, Twitter, klasifikasi Teks

Abstract

Social networks have become a widely used and popular media for disseminating information and facilitating social interaction. User interaction provides valuable insights about individual characteristics and behavior. Twitter is a microblogging service. The main activity of Twitter is posting short texts (tweets) via the web or mobile. Personality is a unique attitude towards someone in their behavior. To find out someone's personality based on the statuses (tweets) they wrote on Twitter, a text classification technique was performed using the Multiclass Support Vector Machine method. The language used is Indonesian. The test results using 10 Fold Cross Validation show the accuracy obtained by using the One-against-One and One-against-All strategies and compare the use with and without Symbol removing. The experimental results obtained using One-against-One with and without Symbol removing produces an accuracy of 86.55% and 86.88%, while using One-against-All with and without Symbol removing 85.24% and 85.07%.

Keywords: ***personality,Twitter,text classification***

1. Pendahuluan

Latar Belakang

Kemajuan teknologi informasi di Indonesia telah merambat ke berbagai lapisan masyarakat. Informasi saat ini dapat diperoleh dengan mudah menggunakan teknologi yang sudah canggih, seperti halnya internet[1]. Hal ini juga berpengaruh terhadap pengguna media sosial, menurut hasil survei *we are social* 2019 di Indonesia tercatat jumlah pengguna media sosial aktif mencapai 150 juta (naik 15% atau sekitar 20 dari tahun 2018)[2]. Jejaring sosial menjadi media yang banyak digunakan dan populer untuk penyebaran informasi serta fasilitator interaksi sosial. Interaksi pengguna memberikan wawasan berharga tentang karakteristik dan perilaku individu, salah satu media sosial yaitu Twitter[3]. Twitter merupakan layanan *microblogging* yang dirilis secara resmi pada 13 Juli 2006. Aktifitas utama twitter adalah mem-posting sesuatu yang pendek (tweet) melalui web atau *mobile*. Panjang maksimal dari tweet adalah 140 karakter[4]. pengguna akun twitter dapat menunjukan kepribadian mereka melalui tweet pengguna. Mengetahui kepribadian diri sendiri sangatlah penting karena dapat lebih mengenal diri sendiri, lebih mudah untuk memecahkan masalah, regulasi diri yang lebih baik, mampu menempatkan diri sendiri di situasi yang berbeda-beda dan dapat mengurangi stress.

Kepribadian merupakan sebuah sikap yang unik terhadap seseorang dalam berperilaku dan merupakan segala yang mengarah ke dalam atau keluar dirinya sehingga masing –masing orang memilki perbedaan[1]. Salah

satu teori kepribadian menurut Schwartz Model Kepribadian terbagi atas 10 nilai *Model Shwartz* yaitu *Achievement, hedonism, Benevolence, self-direction, Power, Comformity, Security, Tradition, Universalism*[5].

Untuk mengetahui kepribadian seseorang berdasarkan status-status (tweet) yang mereka tulis di Twitter dilakukan teknik klasifikasi teks menggunakan metode *Support Vector Machine*. Metode SVM dipilih karena dapat menghasilkan hasil yang optimal dalam klasifikasi. Pada riset sebelumnya dilakukan klasifikasi menggunakan *Multinomial Naïve Bayes* namun penelitian tersebut menggunakan data status facebook. Oleh karena itu, pada penelitian ini dilakukan klasifikasi sentimen untuk mengetahui kepribadian seseorang berdasarkan status (tweet) menggunakan SVM. Dataset diambil dari status (tweet) berbahasa indoensia. penelitian ini juga menggunakan metode *Multiclass Support Vector Machine* (SVM) Strategi *One-against-One* dan *One-against-All* dengan fitur TF-IDF. Metode SVM dipilih karena dapat menghasilkan hasil yang optimal dalam klasifikasi [6].

Topik dan Batasannya

Masalah yang dibahas dalam penelitian ini bagaimana mengetahui kepribadian seseorang berdasarkan status (tweet) pengguna twitter menggunakan metode *Multiclass Support Vector Machine* (SVM) Strategi *One-against-One* dan *One-against-All* dengan seleksi fitur TF-IDF.

Batas pekerjaan dalam Penelitian ini yaitu, dataset yang digunakan adalah dataset tweet pengguna Bahasa Indonesia dan jumlah dataset adalah 300 tweet. Penelitian ini dibutuhkan pelabelan oleh 3 orang menggunakan *vote* dengan pelabelan berdasarkan 10 nilai *model Schwartz* dan satu tweet hanya dapat diklasifikasikan dalam satu kelas nilai. *Preprocessing* yang digunakan adalah Tokenisasi, *Stemming*, *Case Folding*, *Symbol Removing* dan *Stopword Removal*. Fitur seleksi yang digunakan yaitu TF-IDF dan metode yang digunakan adalah *Multiclass Support Vector Machine* (SVM) Strategi *One-against-One* dan *One-against-All*.

Tujuan

Tujuan yang ingin dicapai pada penelitian ini adalah mampu mengetahui kepribadian seseorang berdasarkan status (tweet) pengguna twitter berbahasa Indonesia menggunakan metode *Multiclass Support Vector Machine* (SVM) Strategi *One-against-One* dan *One-against-All* dengan seleksi fitur TF-IDF dan mencari performa penggunaan dengan Tanpa *Case Folding.*

Organisasi Tulisan

Selanjutnya pada bab 2 dibahas mengenai studi terkait pada penelitian yang dilakukan, bab 3 membahas teori dan perancangan sistem penelitian, bab 4 membahas evaluasi model penelitian, dan bab 5 membahas tentang kesimpulan.

2. Studi Terkait

2.1 Text Mining

Text Mining adalah bidang yang mengekstrak informasi yang berarti dari teks bahasa alami. Hal itu bisa diartikan sebagai proses menganalisa teks untuk mengekstrak informasi yang berguna untuk tujuan tertentu. Teks pada *text mining* tidak terstruktur, ambigu, dan sulit untuk diproses seperti email, dokumen teks dan lain- lain. Meskipun demikian, teks adalah cara yang paling umum untuk pertukaran informasi formal[7].

2.2 Model Schwartz

Berdasarkan Schwartz, Shalom H. (2003), Model nilai *Schwartz* merupakan salah satu model yang paling banyak diterima dan banyak digunakan dengan menggunkan 10 nilai dasar dan telah menghasilkan keberhasilan besar dalam penelitian psikologi serta bidang lain. Model *Schwartz* memilikii 10 nilai dasar antara lain[5].

1. Pengarahan diri sendiri *(self-direction)* : Ingin bebas dan mandiri (Kreativitas, kebebasan, independen, penasaran, memilih tujuan sendiri).
2. Pencapaian *(Achievement)* : Menetapkan tujuan dan mencapainya (Sukses, mampu, ambisius, berpengaruh)
3. Kebajikan *(Benevolence)* : Berusaha membantu orang lain dan memberikan kesejahteraan umum (Jujur, pemaaf, setia, kehidupan spiritual/rohani, bertanggung jawab, makna dalam hidup, persahabatan sejati, percintaan)
4. Konformitas *(Conformity)* : Mematuhi aturan, hukum dan struktur (Kesopanan, patuh, disiplin diri, menghormati orang tua dan yang lebih tua)
5. Hedonisme *(hedonism)* : Mencari kesenangan dan kepuasaan untuk diri sendiri (Kesenagan, menikmati hidup)
6. Kekuasaan *(Power)* : Mengontrol, mendominasi dan mengendalikan orang lain. (Kekuatan sosial, otoritas, kekayaan)
7. Keamanan *(Security) :* Keselamatan, harmoni dan stabilitas masyarakat, hubungan dan diri. (Keamanan keluarga, keamanan nasional, tatanan sosial, bersih, balas budi)

8. Stimulasi *(stimulation)* : Kegembiraan, kebaruan, dan tantangan dalam hidup (Berani, kehidupan bervariasi, sebuah kehidupan yang menarik)
9. Tradisi *(Tradition)* : Menghormati, komitmen dan menerima adat istiadat dan ide-ide budaya tradisional atau agama (Rendah hati, menerima bagian dalam hidup , taat, menghormati tradisi, moderat)
10. Universalisme *(Universalimsm)* : Mencari perdamaian, keadilan sosial dan toleransi untuk semua (Berwawasan luas, kebijaksanaan, keadilan sosial, kesetaraan, dunia di kedamaian, menjaga lingkungan)

2.3 Perhitungan TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan suatu cara untuk pembobotan sebuah kata dalam sebuah dokumen atau korpus. Metode pembobotan ini merupakan salah satu yang paling populer, karena sekitar 83% sistem rekomendasi berbasis teks menggunakan metode ini [8].

Dalam sebuah dokumen ataupun tweet, setiap *term* yang muncul akan dihitung jumlahnya. *Term* tersebut sesuai dengan topik yang dibahas. Semakin besar jumlah kemunculannya, semakin besar pula nilai bobot yang dimiliki. Kuantitas term yang muncul disebut sebagai *Term Frequency* (TF).

Adapun *Inverse Document Frequency* (IDF) adalah banyaknya kemunculan sebuah *term* pada sebuah dokumen atau tweet yang tidak memiliki hubungan dengan topik sehingga menjadi kata yang tidak berbobot. Sebaliknya, *term* tidak umum yang ada pada sebuah dokumen mencirikan apakan dokumen tersebut sesuai atau tidak dengan topik yang dibahas. Semakin sedikit sebuah dokumen memiliki *term* yang tidak umum, semakin besar nilai IDF nya. Rumus dari IDF adalah [9]:

$$IDF = log \frac{D}{Df} \quad (1)$$

Keterangan:
IDF = Nilai inverse dari DFi
D = Banyaknya tweet pada datasets
Df = Banyaknya tweet pada dataset yang mengandung kata ke-i

Maka nilai TF-IDF adalah:

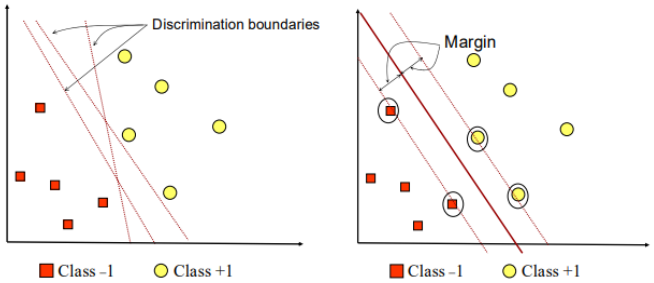
$$TFIDF = TF * IDF \quad (2)$$

Keterangan:
TFIDF = Nilai bobot kata dalam sebuah datasets
TF = Nilai jumlah kemunculan sebuah kata pada tweet
IDF = Nila kata yang muncul dalam sebuah datasets

Nilai TFIDF yang bernilai 0 bermakna kata ke-i memiliki jumlah kemunculan satu kali pada satu tweet dalam dataset yang ada.

2.4 Support Vector Machine

SVM dalam pembelajaran mesin adalah model pembelajaran *supervised* dengan mempelajari algoritma yang terkait untuk menguji data dan mengidentifikasi pola, yang mana digunakan untuk regresi dan analisis klasifikasi[10].Teori yang mendasari SVM sudah berkembang sejak 1960-an, tetapi baru diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992 dan sejak itu SVM berkembang dengan pesat[11].



(a) **(b)**

Gambar 1. Hyperplane class -1 dan class +1

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah class pada *input space.* Gambar 1a memperlihatkan beberapa pattern yang merupakan anggota dari dua buah class : +1 dan –1. Pattern yang tergabung pada class –1 disimbolkan dengan warna merah (kotak), sedangkan pattern pada class +1, disimbolkan dengan warna kuning(lingkaran). Problem klasifikasi dapat

diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar 1-a.

Hyperplane pemisah terbaik antara kedua class dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan pattern terdekat dari masing-masing *class*. [11][12].

Hyperplane terbaik adalah *hyperplane* yang terletak di tengah-tengah antara dua set obyek dari dua kelas. Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan *margin* atau jarak antara dua set obyek dari kelas yang berbeda. Berikut adalah rumus perhitungan *hyperplane*.

$$w \,.\, x + b = 0 \tag{3}$$

Dimana:
w: parameter *hyperplane* yang dicari (garis yang tegak lurus Antara garis *hyperplane* dan titik *support vector*)
x: data input SVM (x1 = index kata, x2 = bobot kata)
b: parameter *hyperplane* yang dicari (nilai bias)

Menentukan *hyperplane* terbaik yaitu dengan cara memaksimalkan jarak *margin.* Cara memaksimalkan *margin* yaitu dengan persamaan sebagai berikut:

Persamaan garis pada 2D:

$$ax + by + c = 0 \tag{4}$$

setelah itu, persamaan garis tersebut diubah *x* menjadi x_1, *y* menjadi x_2, *a* menjadi w_1, *b* menjadi w_2. Sehingga menjadi:

$$w_1 x_1 + w_2 x_2 + \mathrm{c} = 0 \tag{5}$$

Asumsi sekarang berada di dimensi d >1, maka persamaan diatas menjadi:

$$w_1 x_1 + \ldots\ldots + w_d x_d + \mathrm{c} = 0 \tag{6}$$

Kemudian rumus diatas diperumum menjadi:

$$\sum_{j=1}^{d} w_j x_j + \mathrm{c} = 0 \tag{7}$$

Cara lain untuk menuliskan persamaan diatas dalam notasi vector :

$$w_1 \; dan \; x_1 =$$
$$\mathrm{x} = [x_i, \ldots\ldots, x_d]^T$$
$$\mathrm{w} = [w_i, \ldots\ldots, w_d]^T$$

jadi persamaan (7) dapat ditulis menjadi:

$$g(x) = \langle w,x \rangle + c = 0 \tag{8}$$

Untuk kasus problem binary classification didefinisikan sebagai berikut :

$$f(x) \begin{cases} +1, jika \; g(x) \geq 1 \\ -1, jika \; g(x) \leq -1 \end{cases} \tag{9}$$

Cara menyatakan margin secara matematis,maka sederhanakan persamaan (9) sebagai berikut:

$$y \; (\langle w,x \rangle + c \;) \geq 1 \tag{10}$$

Dimana y=1 apabila x:= x positif, dan y = -1 apabila x:= negatif.

Margin merupakan proyeksi orthogonal dari vector $X_+ - X_-$ kedalam vector $\vec{w}$ maka,

$$S = \langle \vec{w}, X_+ - X_- \tag{11}$$

Kemudian persamaan di atas disederhanakan kembali menjadi:

$$\max \frac{2}{\|w\|} \rightarrow \frac{\|w\|}{2} \rightarrow \min \frac{1}{2} \|w\|^2 \tag{12}$$

2.4.1 Multiclass SVM

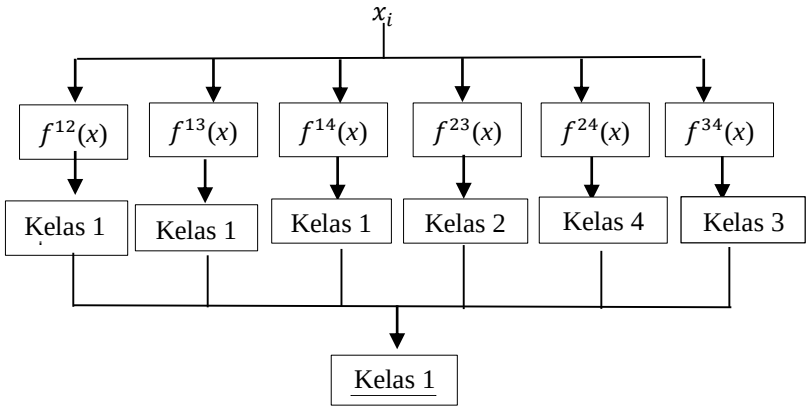
SVM saat pertama kali diperkenalkan oleh Vapnik, hanya dapat mengklasifikasikan data ke dalam dua kelas (klasifikasi biner). Namun, penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi data yang memiliki lebih dari dua kelas.sehingga digunakanlah metode Multiclass SVM. Pada penelitian ini digunakan pendekatan *One-against-One* dan *One-against-All* sebagai berikut:

a. *One-against-One*

Dengan menggunakan metode ini, dibangun sejumlah $\frac{k(k-1)}{2}$ buah model SVM biner, dengan *k* adalah jumlah kelas. Contohnya, untuk 2 masalah klasifikasi dengan 4 buah jumlah kelas, digunakan 6 buah SVM biner pada tabel di bawah ini dan penggunaannya pada pengklasifikasian data baru. Untuk lebih jelasnya perhatikan ilustrasi pada tabel 1 dan gambar 2.

Tabel 1. Contoh kombinasi biner dengan metode *One-against-One*

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Kelas 2	$f^{12}(x) = (w^{12})x + b^{12}$
Kelas 1	Kelas 3	$f^{13}(x) = (w^{13})x + b^{13}$
Kelas 1	Kelas 4	$f^{14}(x) = (w^{14})x + b^{14}$
Kelas 2	Kelas 3	$f^{23}(x) = (w^{23})x + b^{23}$
Kelas 2	Kelas 4	$f^{24}(x) = (w^{24})x + b^{24}$
Kelas 3	Kelas 4	$f^{34}(x) = (w^{34})x + b^{34}$



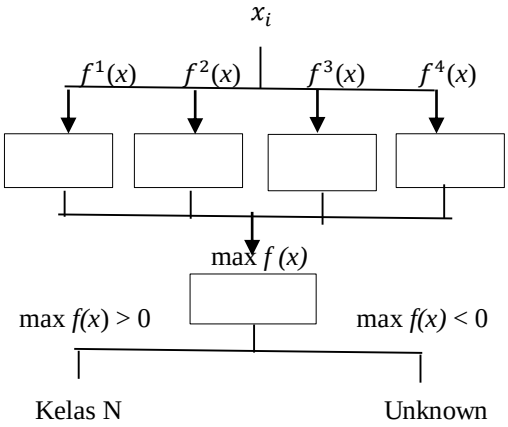
Gambar 2. Contoh klasifikasi dengan metode *One-against-One*

b. *One-against-All*

Dengan menggunakan metode ini, Dibangun sejumlah *k* SVM biner, dengan *k* adalah jumlah kelas. Contohnya, untuk persoalan klasifikasi dengan 4 buah jumlah kelas, digunakan 4 buah SVM biner pada tabel di bawah ini dan penggunaannya pada pengklasifikasian data baru [15]. Untuk lebih jelasnya perhatikan ilustrasi pada Tabel 2 dan Gambar 3

Tabel 2. Contoh kombinasi biner dengan metode *One-against-All*

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Kelas 1	$f^{1}(x) = (w^{1})x + b^{1}$
Kelas 2	Kelas 2	$f^{2}(x) = (w^{2})x + b^{2}$
Kelas 3	Kelas 3	$f^{3}(x) = (w^{3})x + b^{3}$
Kelas 4	Kelas 4	$f^{4}(x) = (w^{4})x + b^{4}$



Gambar 3. Contoh klasifikasi dengan metode *One-against-All*

Pada pendekatan *One-against-All*, misalkan permasalahan yang ditemui terdiri dari *N* kelas. Sehingga akan dibuat *N decision boundary*. *Decision boundary* yang dihasilkan merupakan hasil dari pencarian *hyperplane* dari setiap kelas *i* dengan kelas sisa yang lainnya[13][15].

2.5 Performance Evaluation

Precision dan *recall* merupakan teknik yang dapat digunakan untuk menghitung nilai performasi dari sistem pemrosesan teks. Perhitungan *precision* dan *recall* membutuhkan empat komponen yaitu TP (*True Positive*), TN (*True Negative*), FP (*False Positive*) dan FN (*False Negative*). Gambar 2 adalah tabel *confusion matrix* yaitu tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan[7].

Tabel. 3 *Multilabel Confusion Matrix*

		Prediction			
		Class 1	Class 2	…..	Class n
Actual	Class 1	Accurate			
	Class 2		Accurate		
	…..			Accurate	
	Class n				Accurate

Berikut performasi yang akan diuji antara lain:

c. *Precision* dapat diartikan sebagai rasio dari jumlah ketepatan prediksi suatu kelas terhadap jumlah total prediksi yang diklasifikasikan ke dalam kelas tersebut.

$$precision = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} TP_i + FP_i} \times 100\% \quad (13)$$

d. *Recall* dapat diartikan sebagai rasio dari jumlah ketepatan prediksi suatu kelas terhadap jumlah total fakta yang diklasifikasikan ke dalam kelas tersebut.

$$Recall = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} TP_i + FN_i} \times 100\% \quad (14)$$

e. *F1-measure* merupakan hasil rata-rata antara *precision* dan *recall*. *F1-measure* ini muncul untuk menyetarakan nilai *precision* dan *recall* yang sering terpaut jauh. *F1-measure* juga diartikan sebagai penyetaraan nilai *precision* dan *recall*.

$$F1 = \frac{2(precison \times recall)}{precison + recall} \quad (15)$$

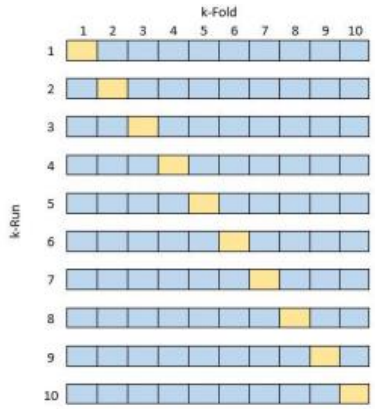
f. *Accuracy* merupakan perhitungan mengukur rasio prediksi yang benar mengenai jumlah total kasus yang dievaluasi

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (16)$$

Evaluasi untuk klasifikasi *multi-label* menggunakan *macro average*. *Macro average* dapat menghitung nilai rata-rata dengan memberikan bobot yang sama untuk setiap kelas. Untuk perhitungan *Macro average* dihitung berdasarkan jumlah TP, TN, FP, FN. Karena ukuran F1 mengabaikan TN dan sebagian besar ditentukan oleh jumlah nilai TP sehingga lebih memihak ke dalam kelas dominan.

2.6 K-Fold Cross Validation

K-fold Cross validation adalah metode statistik yang dapat digunakan untuk mengevaluasi kinerja model atau algoritma dimana data dipisahkan menjadi dua subset yaitu data proses pembelajaran dan data validasi / evaluasi[14].Penelitian ini menggunakan 10-fold cross validation untuk mengevaluasi kinerja klasifikasi. Dataset dipisah menjadi 10 fold seperti yang ditunjukkan pada gambar sebagai berikut[10]:

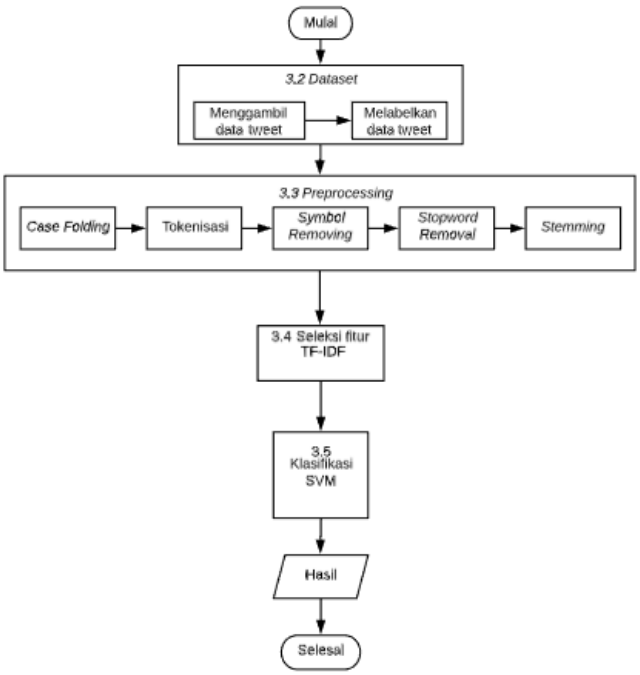


Gambar 5. *10 Cross validation*

3. Sistem yang Dibangun

3.1 Gambaran Umum Sistem

Sistem yang di bangun dalam penelitian ini adalah sistem yang dapat melakukan klasifikasi kepribadian pada data twitter dengan menggunakan metode *Multiclass Support Vector Machine* (SVM) Strategi *One-against-One* dan *One-against-All*. Data yang diambil yaitu status (tweet) pengguna dengan total 300 tweet. Data kemudian dilakukan *preprocessing* dan ditambahkan seleksi fitur TF-IDF pada klasifikasi. Berikut adalah gambaran dari sistem yang dibangun.



Gambar 6. Gambaran Umum Sistem

3.2 Dataset

a. Mengambil Data tweet

Dataset untuk penelitian ini diambil dari Twitter secara manual sebanyak 600 tweet kalimat Bahasa Indonesia dari 100 pengguna twitter dalam bentuk *csv*. dataset yang sudah dikumpulkan dilakukan pelabelan oleh 3 orang sesuai dengan tweet pengguna dan klasifikasi kedalam 10 model nilai *Schwarz*. Adapun kualifikasi sebagai berikut:

Expert 1	
Pendidikan	S1 Psikologi Universitas Muhammadiyah Malang
Umur	25 tahun
Profesi	Staff HRD

Expert 2
Pendidikan S1 Psikologi Universitas Muhammadiyah Malang
Umur 22 tahun
Profesi Guru
Expert 3
Pendidikan S1 Psikologi Universitas Muhammadiyah Malang
Umur 25 tahun
Profesi Konsultan

b. Melabelkan Data tweet

Pada proses ini data tweet yang berbentuk *.csv* dilabelkan oleh *expert* dan mengklasifikasikan data tweet berdasarkan 10 model nilai schwartz. Proses pelabelan menyesuaikan tweet tersebut dengan *keyword* sesuai dengan 10 Model Nilai Schwartz. Adapun contoh klasifikasi data sebagai berikut.

Tabel. 4 Contoh klasifikasi data

Tweets	**Keyword**	**Kelas**
Kita terlalu sibuk meminta nikmat yang belum kita punya, sehingga lupa mensyukuri nikmat yang sudah kita miliki.	Meminta nikmat, lupa mensyukuri	benevolence
Kadang bersikap bodo amat itu perlu, tp bersikap bodo amat yang baik	bersikap bodo amat	self_direction
Jangan karna alasan perdamaian dan persatuan kecurangan menjadi suatu yang baik..	perdamaian, persatuan kecurangan menjadi baik	security

Tweet yang sudah di klasifikasikan kedalam 10 model nilai schwartz, selanjutya tahap pengambilan hasil akhir dilakukan *vote* oleh *expert* sebagai berikut:

Tabel 5. Contoh hasil *vote*

Tweet	**Expert 1**	**Expert 2**	**Expert 3**	**Hasil**
Kita terlalu sibuk meminta nikmat yang belum kita punya, sehingga lupa mensyukuri nikmat yang sudah kita miliki.	benevolence	benevolence	self_direction	benevolence
Kadang bersikap bodo amat itu perlu, tp bersikap bodo amat yang baik	self_direction	self_direction	self_direction	self_direction
Jangan karna alasan perdamaian dan persatuan kecurangan menjadi suatu yang baik..	security	achivment	security	security

Dari hasil vote didapatkan Hasil akhir pelabelan yang merupakan label *self-direction* sebanyak 238 dokumen, *Achievement* sebanyak 26 dokumen, *Benevolence* sebanyak 118 dokumen, *Conformity* sebanyak 18 dokumen, *Hedonisme* sebanyak 14 dokumen, *Power* sebanyak 18 dokumen, *Security* sebanyak 10 dokumen, *Stimulation* sebanyak 100 dokumen, *Tradition* sebanyak 14 dokumen, *Universalisme* sebanyak 50 dokumen.Berikut ini hasil klasifikasi data tweet sebagai berikut:

Tabel 5. Contoh Klasifikasi Data

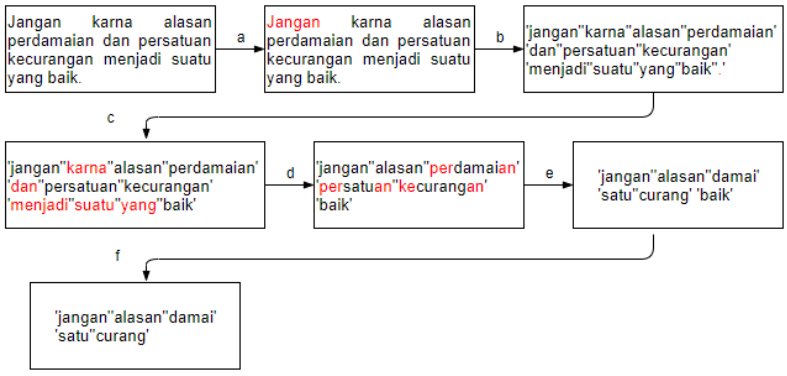
Tweet	**Kelas**
Kita terlalu sibuk meminta nikmat yang belum kita punya, sehingga lupa mensyukuri nikmat yang sudah kita miliki.	benevolence
Kadang bersikap bodo amat itu perlu, tp bersikap bodo amat yang baik	self_direction
Jangan karna alasan perdamaian dan persatuan kecurangan menjadi suatu yang baik..	Security

3.3 Preprocessing

Tahap *Prepocessing* digunakan untuk membersihkan terms-terms yang ada pada data teks namun tidak memiliki nilai yang menujukan kepribadian, hal ini dimaksudkan agar mempermudah penelitian untuk mengklasifikasikan kata. Tahap prepocessing terdiri dari beberapa tahapan, antara lain.[7].

a. *Case folding*
   Case Folding merupakan tahapan untuk mengubah semua huruf dalam teks menjadi huruf kecil. Sehingga tidak akan ditemukan huruf capital dalam teks.
b. Tokenisasi
   Tokenisasi untuk memisahkan antara kata dengan kata lain/tanda baca
c. *Symbol Removing*
   Symbol Removing adalah proses menghapuskan symbol pada data.
d. *Stopword Removal*
   Stopword Removal adalah proses menghilangkan kata yang tidak memiliki makna pada sebuah dokumen.
e. *Stemming*
   Stemming adalah proses mengubah kata berimbuhan menjadi kata dasar.
f. *Slang Handling*
   Slang Handling adalah proses pemeriksaan setiap kata dalam dataset.

Berikut adalah contoh *Preprocessing* pada salah satu dataset yang digunakan pada penelitian ini:



Gambar 7. Contoh Proses *Preprocessing*

3.4 Perhitungan TF-IDF

Pada metode TF-IDF ini, dilakukan perhitungan untuk bobot *term* t dalam sebuah tweet. Persamaan yang digunakan untuk perhitungan TF-IDF adalah persamaan (1) dan (2). Tweet pada perhitungan TF-IDF sudah melalui tahap *Preprocesing*. Berikut ini proses perhitungan TF-IDF sebagai berikut[16]:

Tabel 5. Contoh Dokumen perhitungan TF-IDF

T1 : mengajariku terbang berikanku kepastian tinggi angkasa aksaratu malamaksara
T2: terbang tinggi tinggalkanku peduli terkadang menyenangkan aksaratu malamaksara

Contoh perhitungan pembobotan TF-IDF akan diterapkan pada kata angkasa dalam tweet pertama (T1).

a. Menghitung *Term Frequency* (tf)
   Data hasil *preprocessing* dilakukan perhitungan kemunculan kata angkasa (*term frequency* (tf)) pada setiap tweet. tweet pada contoh kasus ini adalah T1 dan T2.

Tabel 6. *Term Frequency* kata angkasa

Kata	Term Frequency (tf)	
	T1	T2
angkasa	1	0

Dari table 6 dijelaskan bahwa kata angkasa pada tweet pertama (T1) muncul sebanyak satu kali sedangkan pada T2 kata angkasa tidak muncul sama sekali (0).

b. Menghitung *Document Frequency* (df)

Nilai *document frequency* (df) didapatkan dari jumlah tweet yang mengandung kata term yaitu 1 pada tweet (T1).

c. Menghitung *Inverse Document Frequency* (idf)
Nilai *inverse document frequency* (idf) kata angkasa didapatkan dengan cara dihitung menggunakan persamaan (1):

$$IDF = log\frac{D}{Df} = log\frac{2}{1} = \log(2) = 0.30103$$

D merupakan jumlah tweet, pada kasus ini jumlah tweet ada 2 yaitu T1 dan T2. Kemudian nilai df kata angkasa sudah didapatkan pada langkah sebelumnya. Jadi nilai idf pada kata angkasa adalah 0.30103.

d. Menghitung TF-IDF
Kata angkasa pada setiap pernyataan dihitung menggunakan persamaan (2). Berikut perhitungannya

$$TF - IDF_{D1} = tf_{D1} \times IDF = 1 \times 0.30103 = 0.30103$$
$$TF - IDF_{D2} = tf_{D2} \times IDF = 0 \times 0.30103 = 0$$

Dari perhitungan diatas diperoleh nilai TF-IDF untuk T1 adalah 0.30103 sedangkan untuk T2 adalah 0. Berikut hasil perhitungan pembobotan TF-IDF.

Tabel 7. Contoh Perhitungan TF-IDF

Term(t)	TF		Df	D/Df	IDF= log(D/Df)	TF-IDF= TF*IDF	
	T1	T2				T1	T2
mengajariku	1	0	1	2	0.30103	0.30103	0
terbang	1	1	2	1	0	0	0
berikanku	1	0	1	2	0.30103	0.30103	0
kepastian	1	0	1	2	0.30103	0.30103	0
tinggi	1	1	2	1	0	0	0
angkasa	1	0	1	2	0.30103	0.30103	0
aksaratu	1	1	2	1	0	0	0
malamaksara	1	1	2	1	0	0	0
tinggalkanku	0	1	1	2	0.30103	0	0.30103
peduli	0	1	1	2	0.30103	0	0.30103
terkadang	0	1	1	2	0.30103	0	0.30103
menyenangkan	0	1	1	2	0.30103	0	0.30103

3.5 Klasifikasi Muticlass Support Vector Machine

Pada proses ini dilakukan klasifikasi menggunakan metode multiclass SVM untuk mendapatkan hasil akhir dari pembuatan sistem. klasifikasi menggunakan SVM dimulai dengan mengubah text menjadi data vektor. Vektor dalam penelitian ini memiliki dua komponen yaitu dimensi (*word id*) dan bobot. Bobot ini sering dikombinasikan ke dalam sebuah nilai tf-idf. Proses Pembagian dataset dilakukan dengan menggunakan 10 *Fold Cross Validation*. Sistem akan melakukan 10 kali training dan validasi, dimana setiap eksperimen menggunakan data partisi ke1,2,3, dan seterusnya sampai 10 bergantian disetiap batch *cross validation*-nya sebagai data uji dan memanfaatkan sisa partisi lainnya sebagai data latih.

Pada tahap pelatihan metode *Multi Class Support Vector Machine* ini akan dicari fungsi pemisah (*hyperplane*) yang akan digunakan sebagai model dalam mengklasifikasikan kepribadian berdasarkan tweet. Pada kasus *Multi Class Support Vector Machine* dengan menggunakan strategi *One-against-One* dan *One-against-All*. Strategi *One-against-One* yaitu jumlah *hyperplane* yang dibangun sejumlah $\frac{k(k-1)}{2}$ dengan *k* adalah jumlah kelas.strategi *One-against-All* yaitu jumlah hyperplane yang terbentuk yaitu sebanyak *k* atau jumlah kelas. Kemudian proses *multiclass* SVM untuk melakukan perhitungan menggunakan *kernel* Linear.

3.6 Hasil Klasifikasi SVM

Pada tahap ini dilakukan *summary* terhadap hasil klasifikasi teks meggunakan metode *Support Vector Machine Multiclass*. Data tweet dengan menggunakan *K-fold Cross Validation.*

Tabel 8. Hasil Klasifikasi SVM

Tweet	Actual	Prediksi
T1	achievement	benevolence
T2	hedonism	benevolence
T3	power	benevolence

T4	power	benevolence
T5	self_direction	achievement
T6	self_direction	self_direction
T7	benevolence	universalism
T8	benevolence	benevolence
T9	stimulation	self_direction
T10	benevolence	self_direction

Pada table 8 terdapat hasil klasifikasi SVM dari beberapa tweet. Masing-masing tweet memiliki prediksi dan *actual*. *Actual* diambil pada proses pelabelan data yang dilakukan oleh *expert* dan Prediksi diambil dari hasil klasifikasi teks SVM.

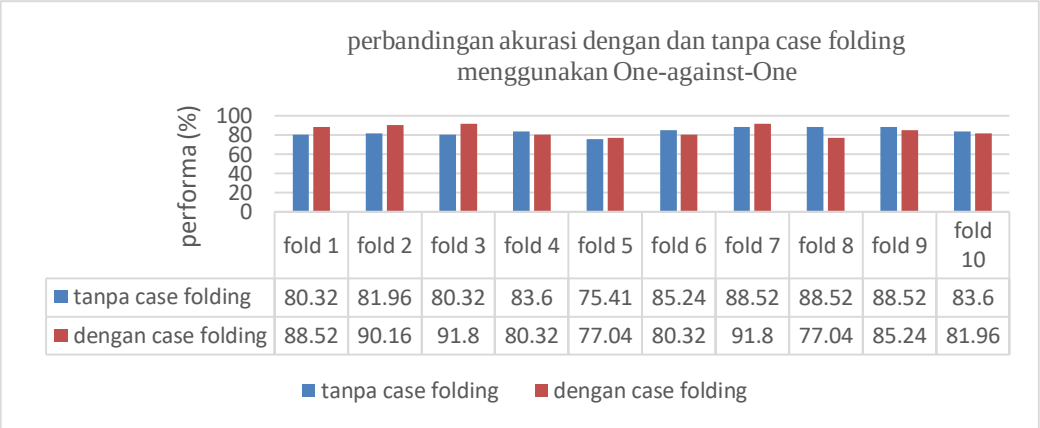
4 Evaluasi

Setelah sistem selesai dibuat, maka tahap selanjutnya yaitu pengujian terhadap sistem. Pengujian ini bertujuan untuk mengetahui seberapa baik sistem yang telah dibuat dengan memastikan bahwa input akan menghasilkan hasil yang sesuai dengan kebutuhan dan bertujuan untuk mengetahui hasil dari performa dari sistem yang sudah dibuat. Pengujian yang dilakukan adalah klasifikasi *Multiclass Support Vector Machine* menggunakan fitur seleksi TF-IDF. Setelah pengujian fitur dilakukan yang selanjutnya dilakukan adalah menguji fitur yang digunakan tanpa dan dengan penggunaan *Case folding*, *Slang handling* dan *symbol removing* yang menerapkan *K-Fold cross Validation* dengan nilai *fold* 1, *fold* 2, *fold* 3,*fold* 4,*fold* 5,*fold* 6,*fold* 7,*fold* 8,*fold* 9 dan *fold* 10.

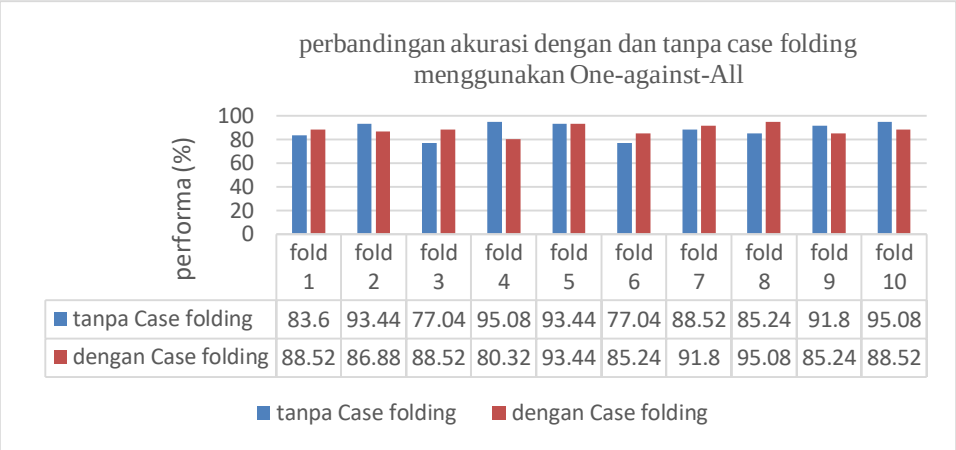
4.1 Skenario 1 Hasil Pengujian performa akurasi penggunaan *Case folding* pada *multiclass* SVM

4.1.1 Hasil dan analisis performa akurasi dengan dan tanpa *Case folding* menggunakan *One- against-One* dan One-against-All

Pada pengujian ini dilakukan perbandingan hasil performa *Multiclass Support Vector Machine* menggunakan fitur TF-IDF dengan *K-Fold Cross Validation* dengan dan tanpa *Case folding*. pengujian ini bertujuan untuk mengetahui apakah ada pengaruh terhadap hasil akurasi apabila bentuk suatu kata diubah. Pada pegujian ini dilakukan skenario dengan tanpa *Case folding* pada proses *Preprocessing*.



Gambar 8. Penggunaan dengan tanpa Case Folding menggunakan *One-against-One*



Gambar 9. Penggunaan dengan tanpa Case Folding menggunakan *One-against-All*

Berdasarkan gambar 8 dan 9 bahwa tahap *preprocessing* dengan dan tanpa *Case folding menggunakan One-against-One* memperoleh rata-rata akurasi dengan *case folding* adalah 84.42 % dan tanpa *Case Folding* 83.60% dan menggunakan *One-against-All* memperoleh rata-rata akurasi dengan *case folding* adalah 88.36 % dan tanpa *Case Folding* 88.03% hasil akurasi menunjukan bahwa klasifikasi dengan *case folding* menghasilkan akurasi lebih tinggi daripada tanpa *case folding* walaupun hasilnya tidak signifikan. Berikut merupakan analisis dari masing-masing skenario yang dilakukan:

a. Analisis proses klasifikasi tanpa *Case folding*

Tahapan *preprocessing* pada proses ini adalah Tokenisasi, *Symbol Removing*,*Stemming* dan *Stopword Removal.* Akurasi yang dihasilkan dengan menggunakan *One-against-One* tanpa *Case Folding* 83.60% dan hasil akurasi menggunakan *One-against-All* tanpa *Case folding* 88.03%. Proses tanpa *Case Folding* yaitu tanpa mengubah huruf kapital menjadi huruf kecil. Akurasi tanpa *Case Folding* lebih rendah karena terdapat beberapa kata dalam huruf kapital yang tidak diubah pada tahap preprocessing. Berikut contoh proses preprocessing tanpa *case folding* sebagai berikut.

```
[Terimakasih,untuk,semua,pelajaran,yang,…
[Persembahan,terbaik,dari,kita,untuk,nege…
```

b. Analisis proses klasifikasi dengan *Case folding*

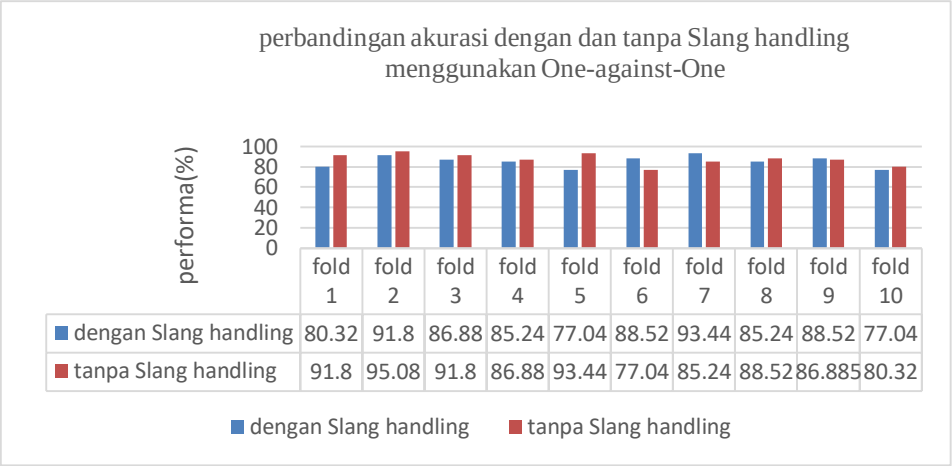
Tahapan *preprocessing* pada proses ini adalah *Case folding*, Tokenisasi, *Symbol Removing* dan *Stopword Removal.* Akurasi yang dihasilkan dengan menggunakan *One-against-One* dengan *Case Folding* 84.42% dan hasil akurasi menggunakan *One-against-All* dengan *Case folding* 88,36%. Proses dengan *Case Folding* yaitu mengubah huruf kapital menjadi huruf kecil. Akurasi dengan *Case Folding* lebih tinggi karena terdapat beberapa kata dalam huruf kapital yang diubah pada tahap *preprocessing*, sehingga dapat meningkatkan akurasi.Berikut contoh proses *preprocessing* dengan *case folding* sebagai berikut.

```
['terimakasih','semua','ajar,'sangat','harga','and','i','proud','to','be','a','part,','of','telu','ombtelu15','ombtelu2015']
['sembah','baik','negeri','welovesso']
```

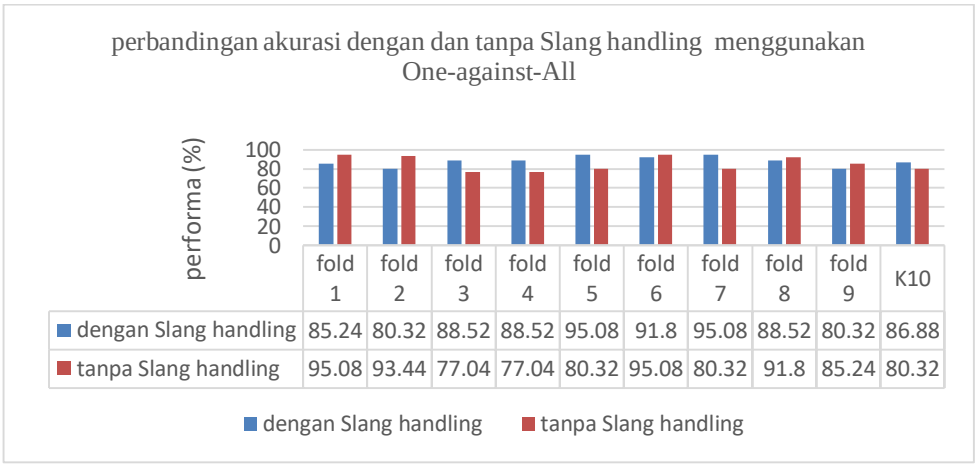
4.2 Skenario 2 Hasil Pengujian performa akurasi penggunaan *Slang handling* pada *multiclass* SVM

Pada pengujian ini dilakukan perbandingan hasil performa *Multiclass Support Vector Machine* menggunakan fitur TF-IDF dengan *K-Fold Cross Validation* dengan dan tanpa *Slang handling*. pengujian ini bertujuan untuk mengetahui apakah ada pengaruh terhadap hasil akurasi apabila beberapa kata dihilangkan. Pada pegujian ini dilakukan skenario dengan tanpa *Slang handling* pada proses *Preprocessing.*

4.2.1 Hasil dan analisis performa akurasi dengan dan tanpa *Slang handling* menggunakan *One- against-One* dan *One-against-All*



Gambar 10. Penggunaan dengan tanpa *Slang handling* menggunakan *One-against-One*



Gambar 11. Penggunaan dengan tanpa *Slang handling* menggunakan *One-against-All*

Berdasarkan gambar 10 dan 11 bahwa tahap *preprocessing* dengan dan tanpa *Slang handling* menggunakan *One-against-One* memperoleh rata-rata akurasi dengan *Slang handling* adalah 85.4% dan tanpa *Slang handling* 87.7% dan menggunakan *One-against-All* memperoleh rata-rata akurasi dengan *Slang handling* adalah 88.02 % dan tanpa *Slang handling* 85.56%. hasil akurasi menunjukan bahwa klasifikasi dengan *case folding menggunakan One-against-One* menghasilkan akurasi lebih rendah daripada tanpa *Slang handling* sebaliknya menggunakan *One-against-All* menunjukan dengan *Slang handling* lebih tinggi walaupun hasilnya tidak signifikan. Berikut merupakan analisis dari masing-masing skenario yang dilakukan:

a. Analisis proses klasifikasi tanpa *Slang handling*

Tahapan *preprocessing* pada proses ini adalah *Case folding*, Tokenisasi, *Symbol Removing Stemming* dan *Stopword Removal.* Akurasi yang dihasilkan dengan menggunakan *One-against-One* tanpa *Slang handling* 87.7% dan hasil akurasi menggunakan *One-against-All* tanpa *Slang handling* 85.56%. Proses tanpa *Slang handling* yaitu memeriksa kembali kata yang terdapat dalam teks. Akurasi tanpa *Slang handling* menggunakan *One-against-One* lebih tinggi dibandingkan dengan menggunakan *One-against-All*. Hal ini menunjukan bahwa proses *Slang handling* dapat memilki akurasi yang tinggi jika menggunakan *One-against-One*.contoh kata yang dapat di proses kedalam *Slang hadling* seperti [‘ga’, ‘gw’,’lu’,’wkwk’,’huft’]

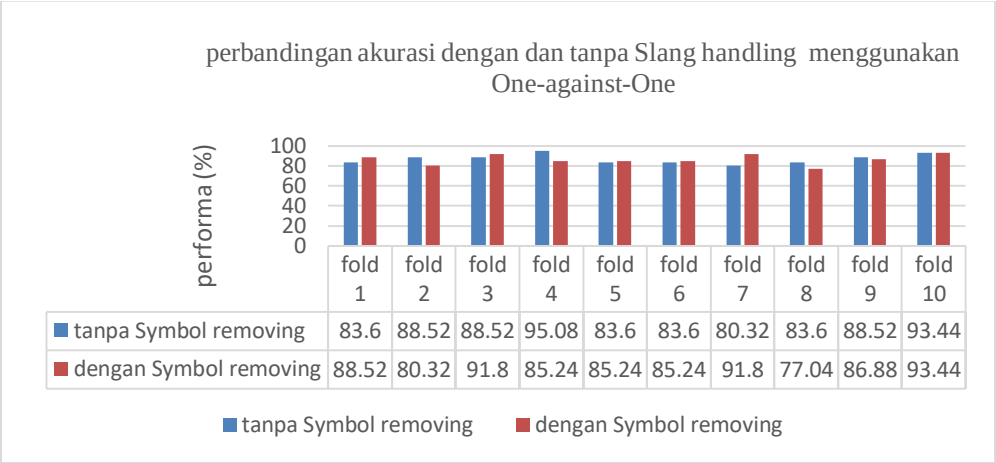
b. Analisis proses klasifikasi dengan *Slang handling*

Tahapan *preprocessing* pada proses ini adalah *Slang handling* Akurasi yang dihasilkan dengan menggunakan *One-against-One* dengan *Slang handling* 85.4% dan hasil akurasi menggunakan *One-against-All* dengan *Slang handling* 88.02%. Proses dengan *Slang handling* yaitu memeriksa kembali kata yang terdapat dalam teks, seperti [‘ga’, ‘gw’,’lu’,’wkwk’,’huft’] setelah itu akan dilakukan proses dengan *Slang handling*. Akurasi dengan *Slang handling* menggunakan *One-against-One* lebih rendah dibandingkan dengan menggunakan *One-against-All*. Hal ini menunjukan bahwa proses *Slang handling* dapat memilki akurasi yang tinggi jika menggunakan One-against-All.

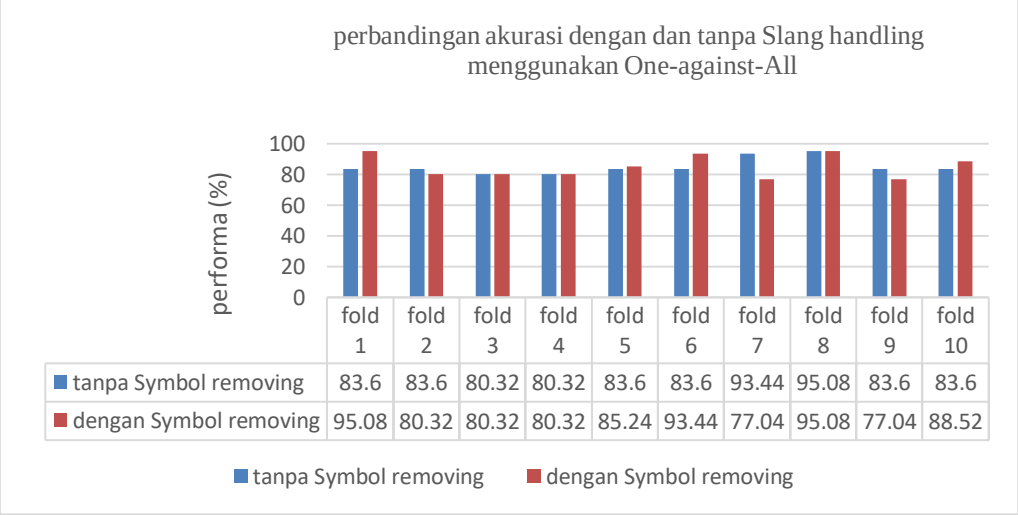
4.3 Skenario 3 Hasil Pengujian performa akurasi penggunaan *Symbol Removal* pada *multiclass* SVM

Pada skenario ini dilakukan perbandingan hasil performa *Multiclass Support Vector Machine* menggunakan fitur TF-IDF dengan *K-Fold Cross Validation* dengan dan tanpa *Symbol Removal*. pengujian ini bertujuan untuk mengetahui apakah ada pengaruh terhadap hasil akurasi apabila tanda baca(!) dan *emoticon* tidak dihilangkan. Pada pegujian ini dilakukan skenario dengan tanpa *Symbol Removal* pada proses *Preprocessing.*

4.3.1 Hasil dan analisis perbandingan performa akurasi dengan dan tanpa *Symbol Removal* menggunakan *One-against-One* dan *One-against-All*



Gambar 12. Penggunaan dengan tanpa *Symbol Removing* menggunakan *One-against-One*



Gambar 13. Penggunaan dengan tanpa *Symbol Removing* menggunakan *One-against-All*

Berdasarkan gambar 12 dan 13 bahwa tahap *preprocessing* dengan dan tanpa *Symbol removing* menggunakan *One-against-One* memperoleh rata-rata akurasi dengan *Symbol removing* adalah 86.55% dan tanpa *Symbol removing* 86.88% dan menggunakan *One-against-All* memperoleh rata-rata akurasi dengan *Symbol removing* adalah 85.24 % dan tanpa *Symbol removing* 85.07%. hasil akurasi menunjukan bahwa klasifikasi dengan *Symbol removing* menggunakan *One-against-One* menghasilkan akurasi lebih rendah daripada tanpa *Symbol removing* sebaliknya menggunakan *One-against-All* menunjukan dengan *Symbol removing* lebih tinggi walaupun hasilnya tidak signifikan. Berikut merupakan analisis dari masing-masing skenario yang dilakukan.

a. Analisis proses klasifikasi tanpa *Symbol removing*

Tahapan *preprocessing* pada proses ini adalah *Symbol Removing*. Akurasi yang dihasilkan dengan menggunakan *One-against-One* tanpa *Symbol removing* 86.88% dan hasil akurasi menggunakan *One-against-All* tanpa *Symbol removing* 85.07%. Proses tanpa *Symbol removing* yaitu menghilangkan *symbo*l pada teks. Pada pengujian ini dilakukan proses *preprocessing* tanpa menghilangkan *Symbol* tanda baca(!) dan *emoticon*. Akurasi

tanpa *Symbol removing* menggunakan *One-against-One* lebih tinggi dibandingkan dengan menggunakan *One-against-All*. Hal ini menunjukan bahwa proses *Symbol removing* dapat memilki akurasi yang tinggi jika menggunakan *One-against-One*.contoh *symbol* yang dapat di proses kedalam *Symbol removing* seperti['!','☺','☹'].

b. Analisis proses klasifikasi dengan *Symbol removing*

Tahapan *preprocessing* pada proses ini adalah *Symbol Removing.* Akurasi yang dihasilkan dengan menggunakan *One-against-One* dengan *Symbol removing* 86.55% dan hasil akurasi menggunakan *One-against-All* dengan *Symbol removing* 85.24%. Proses dengan *Symbol removing* yaitu menghilangkan *symbo*l pada teks, seperti *Symbol* tanda baca(!) dan *emoticon*. Adapun *symbol* yang dapat di proses kedalam *Symbol removing* seperti['!','☺','☹','#','@','*]. Akurasi dengan *Symbol removing* menggunakan *One-against-One* lebih tinggi dibandingkan dengan menggunakan *One-against-All*. Hal ini menunjukan bahwa proses *Symbol removing* dapat memilki akurasi yang tinggi jika menggunakan *One-against-One*.

4.4 Skenario 4 Hasil Pengujian Klasifikasi Kepribadian berdasarkan Data Tweet

Pada pengujian klasifikasi kepribadian data tweet berdasarkan 10 model nilai schwartz yang dilakukan oleh *expert*. Contoh klasifikasi kepribadian seseorang.

Tabel 9. Data Tweet

Username	Tweet 1	Tweet 2	Tweet 3
xxxx	Sidang bikin mules perut. Padahal baru sidang proposal hft	Masi aja cemburu sama mantannya elah	Percayalah. Semua perbuatan akan dibales. Bukan kita yg bls. Tp org lain dan bahkan pembalasan itu lebih menyakitkan drpd yg kamu lakukan.

Berdasarkan Tabel 8 bahwa untuk masing-masing data tweet membahas hal yang berbeda. Hasil klasifikasi kepribadian yang dilakukan oleh *expert* dengan menggunakan 10 model *nilai schwartz* dapat di analisis sebagai berikut:

a. Analisis terhadap Tweet pada klasifikasi kepribadian

Berdasarkan proses pelabelan data tweet username xxxx pada tweet 1 memiliki nilai *Stimulation* , tweet 2 memiliki nilai *Benevolence* dan tweet 3 memilki nilai *Benevolence*. Proses klasifikasi kepribadian terhadap username xxxx yaitu dominan nilai *Benevolence*. proses klasifikasi kepribadian bisa saja memilki nilai lebih dari 2 hal ini dikarenakan seseorang yang memposting teks atau tweet berdasarkan suasana yang dirasakan saat itu, sehingga proses yang terjadi berdasarkan kondisi saat seseorang memposting hal yang dia rasakan saat itu.

5 Kesimpulan dan Saran

Berdasarkan hasil dan analisis yang telah dilakukan dalam penelitian ini, maka dapat diambil kesimpulan bahwa:

1. Hasil penelitian *Multiclass Support Vector Machine* menggunakan fitur TF-IDF dengan *K-Fold Cross Validation* dengan dan tanpa *case folding* menghasilkan Akurasi rata-rata dengan menggunakan *One-against-One* yaitu 84.42% dan 83.6% sedangkan menggunakan *One-against-All* yaitu 88.36% dan 88.03%
2. Hasil penelitian rata akurasi dengan *Symbol removing* menghasilkan akurasi dengan dan tanpa *Symbol removing* menggunakan *One-against-One* adalah 86.55% dan 86.88%, sedangkan menggunakan *One-against-All* dengan dan tanpa *Symbol removing* mendapatkan rata-rata akurasi 85.24% dan 85.07%. hal ini menunjukan menggunakan *One-against-One* pada proses *Symbol removing* lebih baik dibandingkan dengan menggunakan *One-against-All*
3. Hasil penelitian klasifikasi kepribadian seseorang terhadap tweet yang diposting tergantung kepada kondisi dan suasana yang dirasakan oleh pengguna. Oleh karna itu seseorang dapat memilki lebih dari 1 kondisi tergantung suasana yang dirasakan saat itu.

Adapun saran yang dapat dipertimbangkan dalam penelitian ini adalah

1. Menambhkan jumlah dataset dengan kosakata yang lebih bervariasi agar data yang diolah menjadi lebih baik.
2. Menerapkan teknik untuk menangani singkatan pada kata pada proses *Preprocessing*
3. Menerapkan seleksi fitur dan *classifier* yang lain untuk mengembangkan penelitian ini