PENGGUNAAN METODE SUPPORT VECTOR MACHINE UNTUK KLASIFIKASI SENTIMEN E-WALLET

Bryan Filemon [1] Viny Christanti Mawardi [2] Novario Jaya Perdana [3]

[1] [2] [3] Teknik Informatika, FTI, Universitas Tarumanaraga
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia
[1]email : bryan.535170106@stu.untar.ac.id, [2]email :vinym@fti.untar.ac.id, [3]email :novariojp@fti.untar.ac.id

ABSTRACT

In Google Play Store there are lots of application ready to be explored and downloaded. Google Play Store is a place where many developers can sell applications that they have made. Apart from being a place for searching and downloading applications, Google Play Store can also be used to conduct a research. E-Wallet is one of a technological development that can be used to do many transactions. Doing transaction with e-wallet can be done anywhere you want. E-wallet in Indonesia is growing very rapidly especially in the present time where covid-19 is growing rapidly. This is one of the reasons why many people now using e-wallet for doing transcations. Many interesting promotions that were given is also one of the reason why people start using e-wallet. This research had the objective to visualize people's emotion on e-wallet based on user opinion in Google Play Store. The research stage starts from scrapping data from Google Play Store, preprocessing data, classification with Support Vector Machine, evaluation with confusion matrix. Data were scrapped from google play store using google_play_scrapper API. This research uses OVO review of 500 data, DANA review of 500 data, LinkAja review of 500 data. The classification results will then be evaluated using a confusion matrix. The highest accuracy results will be used as a model for the classification stage. The classification results will be displayed in the form of tables and pie charts that describe the percentage results of sentiment classification.

Keywords: *Confussion Matrix, Google Play Store, preprocessing, Support Vector Machine*

1. Pendahuluan

Pada zaman sekarang ini, teknologi sudah berkembang sangat pesat dan cepat sehingga mendorong banyak kegiatan sehari-hari dapat dilakukan secara digital. Pembayaran secara non-tunai ini akan terus digunakan ke depannya karena teknologi ini sudah sangat menempel di masyarakat dan ditambah lagi pembayaran secara non-tunai ini sangat mudah untuk dilakukan. Seiring dengan pesatnya perkembangan teknologi, pembayaran secara non-tunai atau *cashless* akan menjadi metode pembayaran yang terus digunakan dalam kehidupan sehari-hari karena sangat mudah dan cukup menggunakan *smartphone*[1]. Dikarenakan adanya pandemic covid-19 seperti saat ini, pembayaran secara non-tunai dikabarkan semakin meningkat karena terdapat banyak orang yang takut untuk melakukan kontak fisik dengan orang lain sehingga pembayaran yang umumnya dilakukan secara tunai telah berubah menjadi *cashless* atau non-tunai untuk mencegah penularan virus Covid-19. Menurut data Bank Indonesia dari bulan Januari 2021 sampai dengan bulan Juli 2021, transaksi e-wallet di Indonesia mencapai Rp. 157,4 trilliun. Ada banyak e-wallet yang digunakan di Indonesia seperti OVO, ShopeePay, GO-PAY,DANA,dan LinkAja[2]. Pada penelitian ini akan menganalisis OVO,DANA,LinkAja karena e-wallet ini sudah banyak digunakan di Jakarta. Analisis sentiment adalah ilmu yang melakukan analisis mengenai sentiment, dan emosi terhadap suatu produk. Untuk dapat menilai feedback dari user terhadap suatu layanan e-wallet diperlukan analisis sentiment terhadap ulasan user yang masuk pada google play store[3]. Support Vector Machine adalah salah satu metode klasifikasi dengan *machine learning* yang didasari oleh pola dari hasil proses *training* yang diciptakan oleh Vladimir Vapnik[4]. Sentimen user akan diklasifikasikan menjadi sentiment positif, sentiment negative, sentiment netral. Peneliti akan melakukan analisis sentiment terhadap OVO, DANA, dan LinkAja. Hasil dari analisis akan divisualisasikan ke dalam bentuk chart supaya data atau informasi yang dihasilkan akan lebih mudah untuk dipahami. Visualisasi data juga akan mempermudah dalan proses pengambilan keputusan karena hubungan antar beberapa variiabel data dapat lebih mudah untuk dipahami. Hasil klasifikasi akan ditampilkan dalam bentuk pie chart.

2. Metode

2.1 Scrapping Data

Peneliti akan mengambil data ulasan pada aplikasi OVO, DANA, dan LinkAja yang berada dalam aplikasi google play store. Total data yang diambil adalah sebanyak 1500 data dengan mengambil 500 data ulasan untuk setiap aplikasinya. Pengambilan data dilakukan dengan menggunakan library google_play_scrapper_API dengan bahasa pemograman python

2.2 Preprocessing

Data yang sudah berhasil diambil akan dibersihkan dengan tahap-tahap preprocessing supaya data dapat diolah. Ada beberapa tahap dalam preprocessing yaitu:

1. Cleansing

Pada tahap ini, teks akan dibersihkan dari kata-kata yang dianggap tidak perlu untuk menghilangkan komponen yang dinilai sebagai noise. Contoh komponen yang akan dihilangkan: alamat website (URL), username, hashtag, angka.

2. Case Folding

Pada tahap ini, seluruh kata dalam sebuah kalimat akan diubah menjadi huruf kecil. Proses ini diperlukan supaya semua kata menjadi setara dan supaya tidak terjadi kesalahan dalam memberi bobot kepada suatu kata.

3. Stemming

Pada ttahap ini , setiap kata dalam sebuah kalimat akan diperiksa satu-satu lalu untuk kata-kata yang memiliki imbuhan maka, imbuhan yang terdapat di dalam kata-kata tersebut akan dihilangkan. Tujuan dari proses stemming ini adalah untuk mengubah suatu kata yang memiliki imbuhan menjadi kata dasarnya. Contoh dari stemming: kata “pembayaran” akan dirubah menjadi kata “bayar”

4. Tokenisasi

Pada tahap ini, semua kata dalam suatu kalimat akan dipisahkan satu-satu sehingga menjadi token. Proses ini menggunakan spasi sebagai pembatas yang menjadi tanda pemisah untuk setiap katanya sehingga setiap kata yang dipisahkan oleh spasi akan diubah menjadi bentuk token. Hal ini harus dilakukan supaya dapat dihitung bobotnya dengan TF-IDF.

2.3 Pembobotan TF-IDF

Pada tahapan TF fungsinya untuk menghitung jumlah kemunculan kata pada suatu dokumen. Term frequency ini didasari pada aspek lokal pada TF-IDF *monocity*. *Term frequency* (TF) berfungsi untuk menentukan bobot suatu kata dalam sebuah dokumen yang berdasarkan banyak munculnya suatu kata didalam suatu dokumen. Persamaan *term frequency(tf)* dapat dilihat pada **persamaan(1)**

$$tf_{td} = f_{td} \quad (1)$$

tf_{td} = Nilai TF

f_{td}= frekuensi kata t dalam data d.

Inverse Document Frequency (IDF) merupakan jumlah teks berisi *term* yang dicari dalam suatu dataset. IDF biasa disebut *Global Weight* yang fungsiinya adalah mendefinisikan kontribusi dari kata ke dalam dokumen. Persamaan IDF dapat dilihat pada **persamaan(2)**

$$idf = \log_{10} \frac{N}{df} \quad (2)$$

N= Jumlah dokumen

df= frekuensi kata dalam suatu dokumen

Metode pembobotan yang diintegerasikan dari *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) dapat dilihat pada **persamaan (3).**

$$w(t,d) = tf * idf \quad (3)$$

TF-IDF ini berfungsi untuk menemukan representasi nilai dari kumpulan data training yang akan dibentuk vector antara dokumen dan kata disatukan berdasarkan kesamaan antara dokumen dan kata.

2.4 Support Vector Machine

Support Vector Machine adalah suatu metode klasifikasi yang menggunakan metode machine learning yang dapat digunakan untuk melakukan prediksi kelas yang didasari oleh model dari hasil proses training dengan prinsip *Structural Risk Minimization.* Klasifikasi dilakukan dengan bidang pembatas yang akan memisahkan kelas positive dan kelas negative. Tujuan dari algoritma Support Vector Machine adalah mencari bidang pembatas yang terbaik sebagai garis pembatas antara dua buah kelas. Penelitian ini akan menggunakan support vector machine kernel linear sebagai model prediksinya. Bentuk support vector machine dengan model klasifikasi linear sebagai bidang pembatas dapat dilihat pada **persamaan (4).**

$$g(\vec{x}) = sign(w^T x + b) \quad (4)$$

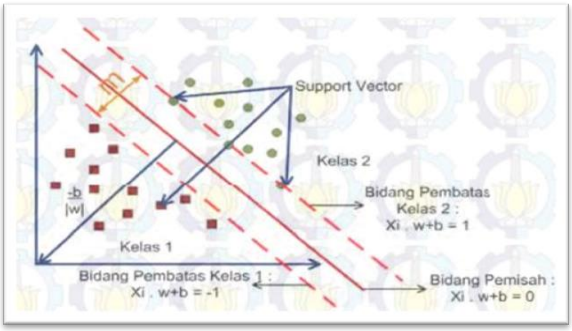
Dimana:

$g(\vec{x})$ = Nilai target $g(\vec{x}) \in \{1, -1\}$

$(\vec{x})$ = vector input

w = vektor bobot

b = bias



Gambar 1 Klasifikasi Linear SVM

Pada dasarnya konsep kerja algoritma Support Vector Machine diartikan sebagai usaha dalam mencari bidang pembatas terbaik yang nantinya akan menjadi pemisah antara 2 buah kelas. Dua kelas, +1 dan -1, masing-masing datanya akan digambarkan dengan simbol kotak merah(-1) dan lingkaran hijau(+1). Data yang termasuk pada kelas +1 dapat dirumuskan seperti pada persamaan(5) dan data yang termasuk pada kelas -1 dapat dirumuskan seperti pada **persamaan (6)**.

$$g(\vec{x}) \geq 1, untuk\ y_i = +1 \quad (5)$$

$$g(\vec{x}) \leq -1, untuk\ y_i = -1 \quad (6)$$

Penentuan hyperlane terbaik dapat dilihat pada persamaan (7)

$$min\ \frac{1}{2}|w^2| \quad (7)$$
$$s.t\ y_i(x_i.w + b) - 1 \geq 0$$

Untuk membuat masalah menjadi lebih efisien untuk dikerjakan maka, permasalahan ini dapat diubah ke bentuk *dual space* sehingga persamaan (7) dapat diubah menjadi fungsi Langrangian. Persamaan (8) adalah bentuk dari fungsi Langragian.

$$L(w, b, a) = \frac{1}{2}|w^2| - \sum \alpha i\ [yi\ (xi\ .\ w + b) - 1] \quad (8)$$

Untuk meminimalkan variabel *w* dan *b* dan dimaksimalkan terhadap variabel *a*, maka diperlukan pencarian turunan pertama dari fungsi *L(w,b,a)* sehingga akan didapatkan dua kondisi optimal seperti pada persamaan(9) dan persamaan(10)

$$\frac{\partial}{\partial w} Lp(w, b, a) = 0 \rightarrow kondisi\ 1 \quad (9)$$

$$\frac{\partial}{\partial b} Lp(w, b, a) = 0 \rightarrow kondisi\ 2 \quad (10)$$

Rumus dalam mencari *hyprlane* ini termasuk ke dalam permasalahan *quadratic programming* sehingga nilai maksimum global dari a_i akan selalu dapat ditemukan. Kelas dari data testing x dapat ditentukan dengan persamaan(11)

$$f(x_d) = sign(\sum_{i=1}^{ns} a_i y_i K(x_i x_{d)} + b) \quad (11)$$

Xi= support vector

ns=banyak vector

Xd= data yang mau diklasifikasi

2.4 Confussion Matrix

Confusion matriks adalah ringkasan hasil keakuratan prediksi pada masalah klasifikasi yang digambarkan dengan tabel. Terdapat 4 istilah nilai pada *confusion matrix* yaitu True Positive (TP), False Positive (FP), False Negative (FN), dan True Negative (TN). *True positive* adalah data yang dideteksi benar positif oleh mesin. True Negative adalah data yang dideteksi benar negative oleh mesin. False Negative adalah data negative tetapi terdeteksi salah oleh mesin. False Positive adalah data positive yang terdeteksi salah oleh mesin. Setelah

mendapatkan masing-masing keempat nilai pada *confussion matrix*, maka akan dilakukan perhitungan *precision*, *recall*, *F1 Score* sebagai evaluasi untuk menilai keakuratan pada model yang telah didapatkan. Perhitungan confussion matriks dilakukan dengan **persamaan (12),(14),(15)**.

$$Precision = TP/(TP + FP) \quad (12)$$

$$Recall = TP/(TP + FN) \quad (14)$$

$$F1Score = 2 * precision * recall/(precision + recall) \quad (15)$$

Dimana:

TP= True Positive

FP= False Positive

FN= False Negative

3. Hasil Percobaan

Pengujian pada program penggunaan metode support vector machine untuk klasifikasi sentiment *e-wallet* dilakukan untuk mengetahui keakuratan hasil klasifikasi dari label. Jumlah data perkelas dapat dilihat pada **Tabel 1**

Tabel 1. Jumlah data label per kelas

Kelas	**Jumlah**
Positive	89
Negative	1241
Netral	170
Jumlah data	**1500**

Untuk pengujian evaluasi dengan Confussion Matrix maka, akan digunakan semua data yang ada dan data akan diuji sebanyak 3 kali untuk setiap kernel linear dan RBF, dengan melakukan pembagian data latih dan data uji. Untuk tabel pengujian Confusion Matrix dapat dilihat pada **Tabel 2.**

Tabel 2. Tabel pengujian Confussion Matrix

Kernel	Data Training	Data Testing	Pembagian data
RBF & Linear	1200	300	80/20
RBF & Linear	1050	450	70/30
RBF & Linear	900	600	60/40

Hasil confussion matrix pada percobaan pertama dapat dilihat pada **Tabel 3**. Sumbu X adalah hasil prediksi dan Y adalah label yang sudah diinput secara manual.

Tabel 3. Confussion Matrix pengujian ke-1 (RBF)

Kernel	**Pembagian Data**		**Banyak Data**		**Akurasi**
	Data Training	Data Uji	Data Training	Data Uji	
RBF	80%	20%	1200	300	87,67%

Pada pengujian pertama untuk kernel RBF, hasil akurasi yang didapatkan sebesar 86,67%, dengan jumlah data training sebesar 1200, dan jumlah data testing sebanyak 300 data. Setelah pengujian pertama untuk kernel RBF selesai maka, akan dilakukan pengujian pertama untuk kernel linear yang dapat dilihat pada **Tabel 4**.

Tabel 4. Confussion Matrix pengujian ke-1 (Linear)

Kernel	Pembagian Data		Banyak Data		Hasil Akurasi
	Data Training	Data Uji	Data Training	Data Uji	
Linear	80%	20%	1200	300	90,67%

Pada pengujian pertama untuk kernel linear, hasil akurasi yang didapatkan sebesar 90,67%, dengan jumlah data latih sebesar 1200, dan jumlah data uji sebesar 300 data. Selanjutnya, akan dilakukan pengujian kedua untuk kernel linear dan kernel RBF.

Tabel 5. Confussion Matrix pengujian ke-2 (Linear)

Kernel	Pembagian Data		Banyak Data		Hasil Akurasi
	Data Training	Data Uji	Data Training	Data Uji	
Linear	70%	30%	1050	450	89,11%

Tabel 6. Confussion Matrix pengujian ke-2 (RBF)

Kernel	Pembagian Data		Banyak Data		Hasil Akurasi
	Data Training	Data Uji	Data Training	Data Uji	
RBF	70%	30%	70%	30%	87,11%

Pengujian kedua untuk kernel linear seperti pada **Tabel 5**, hasil akurasi yang didapatkan sebesar 89,11%, dengan jumlah data latih sebesar 1050 data, dan jumlah data uji sebesar 450 data. Pada pengujian kedua untuk kernel RBF seperti pada **Tabel 6,** hasil akurasi yang didapatkan sebesar 87,11%, dengan jumlah data latih sebesar 1050 data, dan jumlah data uji sebesar 450 data. Setelah pengujian kedua sudah selesai maka, akan dilakukan pengujian ketiga untuk kernel linear dan kernel RBF.

Tabel 7. Confussion Matrix pengujian ke-3(linear)

Kernel	Pembagian Data		Banyak Data		Hasil Akurasi
	Data Training	Data Uji	Data Training	Data Uji	
Linear	60%	40%	60%	40%	88,67%

Tabel 8. Confussion Matrix pengujian ke-3(RBF)

Kernel	Pembagian Data		Banyak Data		Hasil Akurasi
	Data Training	Data Uji	Data Training	Data Uji	
RBF	60%	40%	60%	40%	86,17%

Pengujian ketiga untuk kernel linear seperti pada **Tabel 7**, hasil akurasi yang didapatkan sebesar 88,67%, dengan jumlah data latih sebesar 900 data, dan jumlah data uji sebesar 600 data. Pada pengujian ketiga untuk kernel RBF seperti pada **Tabel 8,** hasil akurasi yang didapatkan sebesar 86,17%, dengan jumlah data latih sebesar 900 data, dan jumlah data uji sebesar 600 data.

Dari ketiga hasil percobaan yang sudah selesai dilakukan, percobaan kesatu dengan membagi data latih sebesar 1200 data dan data uji sebesar 300 data maka, hasil akurasi yang didapatkan untuk kernel linear sebesar 90,67% dan 86,67% untuk kernel RBF. Pada pengujian kedua dengan membagi data latih sebesar 1050 data dan data uji sebesar 450 data maka, hasil akurasi yang didapatkan untuk kernel linear sebesar 89,11% dan 87,11% untuk kernel RBF. Pada pengujian ketiga dengan membagi data latih 900 data dan data uji 600 data, hasil akurasi yang didapatkan sebesar 88,67% untuk kernel linear dan 86,17% untuk kernel RBF. Semua hasil pengujian tersebut dilakukan menggunakan confussion matrix. Untuk jumlah tabel data per kelas terdiri dari 170 kelas positive, 1241 kelas negative, dan 98 kelas netral. Hasil tersebut didapatkan dari pembagian data untuk proses training, di tabel hasil pengujian confussion matrix dapat dilihat bahwa akurasi terbaik untuk kernel linear dan RBF berada di proposi pembagian data 80%/20%. hal ini didapat dikarenakan data yang dibutuhkan data testing dan data training cukup untuk mencapai akurasi yang bagus. Hasil akurasi terkecil berada di 88.67% untuk kernel linear dan 86,17% untuk kernel RBF pada pembagian data 60%/40% hal ini terjadi karena data yang dibutuhkan untuk data training kurang banyak.

4. Kesimpulan

Terdapat beberapa kesimpulan yang dapat diambil setelah menyelesaikan semua pengujian yang harus dilakukan. Beberapa kesimpulan tersebut adalah:

1) Sistem yang dibuat dapat menscrapping data ulasan sebanyak 500 data untuk setiap aplikasi OVO,DANA, dan LinkAja.
2) Sistem yang dibuat dapat dipakai untuk analisis dan klasifikasi ulasan user pada aplikasi e-wallet OVO, DANA, dan LinkAja yang berada dalam Google Play Store.

3) Semakin banyak data yang diproses maka, semakin lama proses klasifikasinya.
4) Sistem dapat mengklasifikasi sentiment menjadi 3 kelas yaitu, sentiment postif, sentiment negatif, sentiment netral.
5) Hasil akurasi terbaik untuk kernel linear didapatkan pada pengujian pertama sebesar 90,67%
6) Sistem yang dibuat ini menggunakan Support Vector Machine kernel linear sebagai model prediksinya.

REFERENSI

[1] Wino, Adi Putra, dan Susanti, Erlin, dan Herwin. "Analisis Sentimen Dompet Elektronik Pada Media Sosial Twitter Menggunakan Naïve Bayess Classifier", IT Journal Research and Development. Vol.5, No.1 2020.
[2] Kristiyanti, Dinar Ajeng. "E-Wallet Sentiment Analysis Using Naïve Bayes and Support Vector Machine". Journal of Physics:Conference Series. Vol.1641, No.2 ,(Agustus 2020)
[3] Saidah, Siti dan Joanna, Mary. "Analisis Sentimen Pengguna Twitter Terhadap Dompet Elektronik Dengan Metode Lexicon Based Dan K-Nearest Neighbor". Jurnal Ilmiah Informatika Komputer. Vol. 25 , No.1 (April 2020).
[4] Mahendrajaya, Rachmad ,dan Buntoro, Asrofi Ghulam, dan Setyawan, Bhanu Muhammad. "Analisis Sentimen Pengguna Gopay Menggunakan Metode Lexicon Based Dan K-Nearest Neighbor". Jurnal Teknik Universitas Muhammadiyah Ponorogo. Vol.3 , No.2 (Oktober 2019)

Bryan Filemon Natawidjaja, seorang Mahasiswa program studi Teknik Informatika Universitas Tarumanagara, Jakarta.

Viny Christanti Mawardi, Memperoleh gelar S.Kom. dari Universitas Tarumanagara tahun 2004. Kemudian memperoleh gelar M.Kom. dari Universitas Indonesia tahun 2008. Saat ini aktif sebagai Dosen Tetap Fakultas Teknologi Informasi Tarumanagara, Jakarta.

Novario Jaya Perdana. Memperoleh gelar S.Kom dari Institut Teknologi Sepuluh Nopember tahun 2011. Kemudian memperoleh gelar M.T. dari Universitas Indonesia tahun 2016. Saat ini aktif sebagai Dosen Tetap Perjanjian Fakultas Teknologi Informasi Tarumanagara, Jakarta.