

# Perbandingan Metode Klasifikasi *Support Vector Machine* dan *Extreme Gradient Boosting* pada Klasifikasi Sentimen Aplikasi *Paylater*

Salma Fitria Fatimatuz Zahrah dan Soeharjoepri  
Departemen Aktuaria, Institut Teknologi Sepuluh Nopember (ITS)  
e-mail: joepri@matematika.its.ac.id

**Abstrak**—Perkembangan *financial technology* (*fintech*) di Indonesia sangat pesat. Salah satu dari perkembangan *fintech* adalah sistem *Buy Now-Pay Later* atau yang biasa disebut *paylater* merupakan pembayaran yang ditunda, dengan kata lain seseorang dapat membeli barang saat ini tanpa membayar langsung namun sebagai gantinya mereka membayar tiap bulan beserta bunganya. Sistem *paylater* sama seperti sistem pada kartu kredit. Contoh aplikasi yang memberikan layanan *paylater* adalah Kredivo. Suatu layanan akan menghasilkan respons dari pengguna, yaitu berupa ulasan. Berdasarkan ulasan tersebut dapat di klasifikasikan berdasarkan sentimen positif dan negatif. Penelitian ini menggunakan dua metode klasifikasi untuk membandingkan ketepatan klasifikasi antara metode *Support Vector Machine* dan *Extreme Gradient Boosting*. Penelitian ini dilakukan analisis klasifikasi menggunakan metode *Support Vector Machine* dengan dua jenis *kernel*, yaitu Linier dan *Radial Basis Function* (RBF). Pada analisis *Support Vector Machine* dengan *kernel* Linier membutuhkan parameter *cost* (C), dimana nilai parameter C yang akan diuji coba adalah 0,5; 0,75; 1; 10; dan 100. Pada analisis *Support Vector Machine* dengan *kernel* RBF menggunakan parameter C dan  $\gamma$ , di mana nilai parameter  $\gamma$  yang akan diuji coba adalah 0,005; 0,05; 0,1; 0,5 dan 0,75. Pada analisis klasifikasi metode *Extreme Gradient Boosting* dilakukan dengan *hyperparameter tuning* dengan bantuan metode *grid-search*. Hasil dari penelitian ini menunjukkan bahwa metode *Support Vector Machine* non-linier dengan *kernel* RBF parameter C = 1 dan  $\gamma = 0,75$  memiliki ketepatan klasifikasi yang lebih baik daripada *Support Vector Machine* Linier dan *Extreme Gradient Boosting*. Dengan hasil rata-rata akurasi, *F-score*, dan AUC sebesar 94,45%; 96,18% dan 92,39%.

**Kata Kunci**—Klasifikasi, Sentimen, *Support Vector Machine*, Ulasan, XGBOOST.

## I. PENDAHULUAN

PERKEMBANGAN *financial technology* (*fintech*) di Indonesia sangat pesat. Menurut laporan Katadata pada tahun 2020 terdapat 33% *fintech* di Indonesia membukukan transaksi lebih dari 80 miliar. Berdasarkan data Asosiasi *Fintech* Pendanaan Bersama Indonesia (AFPI) terdapat 46,6 juta pelaku Usaha Mikro Kecil dan Menengah (UMKM) dan sekitar 132 juta individu yang belum memiliki akses kredit padahal kebutuhan pembiayaan masyarakat sangat tinggi. Sehingga *fintech* dapat menjadi solusi kebutuhan pendanaan.

Salah satu dari perkembangan *fintech* adalah sistem *Buy Now-Pay Later* atau yang biasa disebut *paylater* merupakan pembayaran yang ditunda, dengan kata lain seseorang dapat membeli barang saat ini tanpa membayar langsung namun sebagai gantinya mereka membayar tiap bulan beserta bunganya. Sistem *paylater* serupa seperti sistem kartu kredit. Jika berdasarkan penyedia layanan, *paylater* merupakan kerja sama antara sebuah platform *e-commerce* dengan sebuah perusahaan sedangkan pada kartu kredit disediakan oleh bank.

Juru Bicara Otoritas Jasa Keuangan (OJK), Sekar Putih Djarot menjelaskan bahwa *paylater* merupakan sebuah istilah yang merujuk pada transaksi pembayaran atau jasa. Pada dasarnya *paylater* merupakan layanan untuk menunda pembayaran atau berhutang yang wajib dilunasi di kemudian hari. Layanan yang ditawarkan memudahkan pengguna memenuhi kebutuhannya, namun karena kemudahannya dapat menimbulkan kebiasaan membeli tanpa memperhatikan konsekuensi bunga dan denda keterlambatannya. Hal ini sudah dijelaskan dalam persyaratan mengajukan *paylater*. Penggunaan *paylater* secara tidak sadar menimbulkan dorongan belanja impulsif dan mudah terpengaruh dengan diskon dan tawaran lainnya. *Paylater* juga menyebabkan pengaturan keuangan pribadi terganggu karena sering kali dana yang disisihkan untuk membayar cicilan terpakai untuk memenuhi kebutuhan yang lebih mendesak sehingga mengakibatkan tidak mampu membayar cicilan [3].

Suatu layanan selalu memiliki kekurangan dan kelebihan-nya masing-masing, di mana hal tersebut dapat menghasilkan berbagai respons dari pengguna layanan seperti kepuasan dan kekecewaan terhadap layanan tersebut. Salah satu contoh aplikasi yang menyediakan layanan tersebut adalah aplikasi Kredivo. Kredivo merupakan aplikasi yang menawarkan pinjaman *online* berupa *paylater* maupun cicilan yang tersedia di berbagai *merchant* di Indonesia.

Analisis sentimen merupakan studi komputasi linguistik dan pengolahan bahasa alami yang bertujuan untuk menganalisis pendapat, penilaian dan emosi seseorang mengenai topik tertentu. Opini dapat mengenai produk, layanan, individu, maupun organisasi. Analisis sentimen. Analisis sentimen bertujuan mengklasifikasi teks yang ada di sebuah dokumen yang dapat dikelompokkan berdasarkan sifat positif atau negatif. Terdapat banyak metode klasifikasi yang dapat digunakan dalam melakukan klasifikasi teks di antaranya adalah *Naive Bayes Classifier* (NBC) *K-Nearest Neighbour* (KNN), *Support Vector Machines* (SVM), *AdaBoost*, *Extreme Gradient Boosting* (XGBoost), dan lain sebagainya.

Terdapat penelitian mengenai *Support Vector Machine* yang pernah dilakukan, salah satunya oleh Mayasari (2022) dengan judul “*Text Mining* pada Akun Pemerintah Kota Surabaya dengan Metode Regresi Logistik, *Support Vector Machine* (SVM), dan *Naive Bayes Classifier*” [1]. Penelitian tersebut menjelaskan bahwa metode klasifikasi SVM *kernel* RBF lebih baik dari pada metode lainnya, yaitu *Naive Bayes Classifier* dan Regresi Logistik.

Kemudian terdapat penelitian lain mengenai metode *Extreme Gradient Boosting* yang dilakukan oleh Afifah *et al.* (2021) dengan judul “*Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier*”. Penelitian tersebut menjelaskan bahwa untuk memperoleh hasil XGBoost yang

maksimal perlu mengatur nilai parameter yang digunakan [2]. Peneliti lain, yaitu Yulianti dan Soesanto (2022) dengan judul “Penerapan Metode *Extreme Gradient Boosting* (XGBoost) pada Klasifikasi Nasabah Kartu Kredit” juga menjelaskan bahwa penggunaan *hyperparameter tuning* dapat meningkatkan kinerja metode XGBoost [3].

Oleh karena itu, peneliti tertarik untuk membandingkan metode klasifikasi SVM dan XGBoost pada analisis sentimen pengguna aplikasi *paylater*, yaitu pada ulasan aplikasi Kredivo. Data yang digunakan merupakan ulasan dari aplikasi Kredivo pada *Google Play Store*.

Penelitian ini bertujuan untuk mengetahui karakteristik sentimen pengguna aplikasi *paylater* dan menerapkan metode klasifikasi *Support Vector Machine* dan *Extreme Gradient Boosting* serta membandingkan ketepatan klasifikasi antara metode *Support Vector Machine* dan *Extreme Gradient Boosting* dalam sentimen pengguna aplikasi *paylater*.

## II. TINJAUAN PUSTAKA

### A. Machine Learning

*Machine learning* merupakan cabang dari *artificial intelligence* yang secara sistematis menerapkan kan algoritma untuk menyintesis hubungan mendasar antara data dan informasi. *Machine learning* bertujuan untuk memprediksi kejadian atau skenario di masa depan yang tidak diketahui oleh komputer. Proses *learning* memainkan peran penting dalam menggeneralisasi masalah sesuai dengan pengalaman. Pengalaman yang ada dalam bentuk data masa lalu dapat membantu mencapai hasil yang akurat di masa depan [4].

### B. Analisis Sentimen

Analisis sentimen merupakan studi komputasi linguistik dan pengolahan bahasa alami yang bertujuan untuk menganalisis pendapat, penilaian dan emosi seseorang mengenai topik tertentu. Opini dapat mengenai produk, layanan, individu, maupun organisasi. Analisis sentimen bertujuan mengklasifikasi teks yang ada di sebuah dokumen yang dapat dikelompokkan berdasarkan sifat positif atau negatif.

### C. Preprocessing text

*Pre-processing* merupakan proses mengubah data mentah ke dalam bentuk yang mudah dipahami. Proses ini dilakukan karena data mentah memiliki format yang tidak teratur. Berikut merupakan tahapan dalam *pre-processing*: (a) *Cleaning*, yaitu proses membersihkan ulasan dari *noise*. kata yang dihilangkan dalam adalah *emoticons*, angka. (b) *Case folding*, yaitu proses mengubah semua teks menjadi huruf kecil (non-kapital), (c) *Stemming*, yaitu proses untuk mendapatkan kata dasar dengan menghilangkan imbuhan pada kata. Kamus yang digunakan adalah kamus bawaan dari *Python*, yaitu *Sastrawi*, (d) *Stopwords*, yaitu proses menghapus kata penghubung dengan menggunakan kumpulan kata *stopwords* dalam bahasa Indonesia pada *library nltk.corpus*, (e) *Tokenizing*, yaitu proses pemisahan kalimat ulasan menjadi per kata.

### D. Term Frequency-Inverse Document Frequency

Metode ini mengukur seberapa penting suatu istilah dalam dokumen. Kata-kata dalam dokumen diubah menjadi angka penting dengan proses faktorisasi teks. Sesuai dengan namanya, TF-IDF memfaktorisasi/menilai sebuah kata dengan mengalikan *Term Frequency* (TF) kata tersebut dengan *Inverse Document Frequency* (IDF),

$$tf_i = \frac{f_{ij}}{\sum_{i \in j} f_{ij}} \quad (1)$$

$$idf = \log\left(\frac{N}{df}\right) \quad (2)$$

$$w_{ij} = tf \times idf \quad (3)$$

dengan  $i$  merupakan jumlah kata dan  $j$  merupakan jumlah ulasan sehingga  $w_{ij}$  adalah bobot dari kata ke- $i$  pada ulasan ke- $j$ .  $N$  merupakan jumlah kata seluruh ulasan,  $tf$  adalah jumlah kemunculan kata  $i$  pada ulasan, dan  $df$  adalah jumlah ulasan yang mengandung kata ke- $i$ . TD-IDF dilakukan agar data teks dapat dianalisis. *Idf* menghitung seberapa unik atau umum kata di seluruh kumpulan dokumen. Kata unik memiliki *idf* tinggi, sedangkan pada istilah umum memiliki *idf* rendah [5].

### E. K-fold Cross Validation

Pendekatan *k-fold cross validation* membagi kumpulan data secara acak menjadi  $k$  kelompok sampel dengan ukuran yang sama. Salah satu *subset* digunakan sebagai data pengujian dan lipatan lainnya ( $k - 1$ ) digunakan sebagai data pelatihan. Lipatan ( $k - 1$ ) digunakan untuk melatih model statistik *machine learning* dan sisanya digunakan untuk mengevaluasi kinerja prediksi [6].

### F. Synthetic Minority Oversampling Technique

*Synthetic Minority Over-Sampling Technique* (SMOTE) merupakan metode yang digunakan untuk mengatasi masalah *imbalanced* data. Metode ini berupa teknik penambahan jumlah sampel pada data kelas minoritas dengan cara mereplikasi kelas minoritas hingga menghasilkan jumlah data yang sama dengan kelas mayoritas. Metode yang digunakan pada algoritma SMOTE adalah *k-nearest neighbor* (ketetanggaan data). Metode ini termasuk kelompok metode nonparametrik. Metode ini mengelompokkan data terdekatnya berdasarkan jarak *Euclidean* antara kedua data. Misal terdapat dua struktur data dengan  $p$  dimensi, yaitu  $x^T = [x_1, x_2, \dots, x_p]$  dan  $y^T = [y_1, y_2, \dots, y_p]$  sehingga jarak *Euclidean* yang dihasilkan kedua data ditunjukkan pada persamaan sebagai berikut,

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_p - y_p)^2} \quad (4)$$

$$x_{syn} = x_i + (x_{knn} - x_i) \times \tau \quad (5)$$

dengan  $x_{syn}$  adalah data hasil replikasi,  $x_i$  adalah data yang akan direplikasi,  $x_{knn}$  adalah data yang memiliki jarak terdekat dari data yang akan direplikasi, dan  $\tau$  adalah bilangan random dari 0 sampai 1.

### G. Extreme Gradient Boosting

Metode XGBoost merupakan metode yang direkayasa untuk efisiensi waktu komputasi dan sumber daya memori. *Boosting gradient* merupakan pendekatan di mana model baru digunakan untuk memprediksi *residual* atau eror sebelumnya. *Extreme Gradient Boosting* bekerja dengan menggabungkan berbagai metode pengklasifikasi lemah menjadi kuat dengan melatih model satu demi satu menggunakan hasil klasifikasi yang didapat dari model sebelumnya yang disebut *residuals* atau eror. Metode ini mengontrol *overfitting* yang dapat memberikan performa model yang lebih baik. Nilai prediksi pada saat  $t$  dirumuskan sebagai,

$$\hat{y}_i^{(0)} = 0$$

$$\begin{aligned} \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ &\vdots \\ \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \end{aligned} \quad (6)$$

dengan  $f_k(x_i)$  merupakan model baru yang dibangun,  $\hat{y}_i^{(t)}$  adalah nilai prediksi pada saat  $t$ ,  $\hat{y}_i^{(t-1)}$  adalah nilai prediksi pada saat  $t - 1$ ,  $t$  adalah jumlah total model dari *tree base model*, dan  $x_i$  adalah fitur input. Fungsi *objective* dari Persamaan (6) dapat didefinisikan sebagai berikut,

$$Obj^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{t=1}^T \Omega(f_t) \quad (7)$$

dengan  $l$  menunjukkan *loss function*,  $n$  adalah jumlah pengamatan, dan  $\Omega(f_i)$  merupakan fungsi regulasi.

Model *ensemble tree* pada Persamaan (7) termasuk fungsi parameter sehingga tidak dapat dioptimalkan dengan metode pengoptimalan tradisional pada ruang *Euclidean*. Oleh karena itu, model dilatih dengan cara aditif dengan menggunakan  $\hat{y}_i^{(t)}$  pada prediksi *instance* ke- $i$  dan iterasi ke- $t$  [7]. *Loss function* dapat diminimalkan dengan melakukan substitusi Persamaan (6) dan (7) sehingga persamaan menjadi sebagai berikut,

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{t=1}^T \Omega(f_t) \\ &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{konstanta} \\ Obj^{(t)} &= \sum_{i=1}^n \left( y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)) \right)^2 + \sum_{t=1}^T \Omega(f_t) \\ &= \sum_{i=1}^n \left[ 2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2 \right] \\ &\quad + \Omega(f_t) + \text{konstanta} \end{aligned} \quad (8)$$

dengan menerapkan pendekatan *second order* sehingga persamaan menjadi sebagai berikut,

$$Obj^{(t)} = \sum_{i=1}^n \left[ l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + \text{konstanta} \quad (9)$$

dengan

$$\begin{aligned} g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \text{ dan} \\ h_i &= \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)}) \end{aligned}$$

merupakan statistik gradien pertama dan kedua pada *loss function*. Kemudian menyederhanakan persamaan dengan menghapus konstanta.

$$Obj^{(t)} = \sum_{i=1}^n \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (10)$$

Selanjutnya mendefinisikan fungsi regulasi dengan persamaan sebagai berikut,

$$(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (11)$$

$$\begin{aligned} Obj^{(t)} &\approx \sum_{i=1}^n \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T \left[ \left( \sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \end{aligned} \quad (12)$$

dengan  $w$  merupakan bobot "*leaves*" yang digunakan,  $\gamma$  adalah jumlah *leaves* yang ada pada *tree*, dan  $\lambda$  adalah parameter regulasi. Didefinisikan  $I_j = \{i | (x_i) = j\}$  sebagai *set instance* dari *leaf j* dengan  $q(x)$  struktur tetap sehingga dapat menghitung bobot optimal  $w_j^*$  dari *leaf j* sebagai berikut,

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (13)$$

$$Obj^{(t)}(q) = \frac{1}{2} \sum_{j=1}^T \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (14)$$

Persamaan (14) dapat digunakan sebagai fungsi skor untuk mengukur kualitas struktur *tree q*. Dikarenakan tidak mungkin menghitung semua kemungkinan struktur *tree*, *greedy algorithm* dapat membantu dengan memulai dari satu *leaf* dan menambahkan cabang *tree* secara berulang. Diasumsikan  $I_L$  dan  $I_R$  adalah *set instance* dari *node* kiri dan kanan setelah *split*, sehingga setelah dilakukan *split* penurunan *loss* menggunakan persamaan sebagai berikut,

$$\mathcal{L}_{split} = \frac{1}{2} \left[ \frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \quad (15)$$

Terdapat *hyperparameter tuning* yang dapat digunakan untuk mengoptimalkan kinerja XGBOOST, yaitu: (1) *max\_depth* yang menunjukkan kedalaman maksimum *tree* pada angka rentang  $[0, \infty)$  dengan nilai *default* = 0. (2) *min\_child\_weight* yang menunjukkan jumlah minimum berat *instance* yang dibutuhkan pada angka rentang  $[0, \infty)$  dengan nilai *default* = 1. (3) *gamma* yang merupakan pengurangan *loss* minimum yang diperlukan untuk membuat partisi lebih lanjut pada *leaf node tree* pada angka rentang  $[0, \infty)$  dengan nilai *default* = 0. (4) *Subsample* yang merupakan rasio *subsample* dari *training instance*. Misalnya 0,5 berarti XGBOOST secara acak mengumpulkan setengah dari *instance* data untuk menumbuhkan *tree*. Hal ini bertujuan untuk mencegah *overfitting* pada angka rentang  $[0, 1]$  dengan nilai *default* = 1. (5) *Colsample\_bytree* adalah rasio *subsample* kolom saat membangun setiap *tree* pada angka rentang  $[0, 1]$  dengan nilai *default* = 1. (6) *Scale\_pos\_weight* untuk mengatun keseimbangan bobot dengan nilai *default* = 1.

#### H. Support Vector Machine

Model SVM memetakan pengamatan sebagai titik-titik sedemikian rupa sehingga mereka diklasifikasikan ke dalam partisi. Pengamatan baru diprediksi berdasarkan sisi partisi mereka berada. *Support vector* adalah titik data terdekat dengan *hyperplane* yang membagi kelas pada *input* [4].

##### 1) Support Vector Machine Linier Separable

Metode SVM menggunakan fungsi  $g(x) = x_i w + b$  untuk memisahkan dua kelas dengan margin maksimum. Jarak ( $w_1, w_2$ ) setiap *instance* dari *hyperplane* sama dengan  $\frac{|g(x)|}{\|w\|}$ . SVM bertujuan untuk menemukan  $w, b$ , sehingga nilai  $g(x)$  sama dengan 1 untuk titik data terdekat kelas  $w_1$  dan -1 untuk titik data terdekat dari  $w_2$ . Nilai margin antara *hyperplane* adalah  $\frac{2}{\|w\|}$ , sehingga didapatkan optimasi primal SVM sebagai berikut,

$$J(w) = \min \frac{1}{2} \|w\|^2 \quad (16)$$

dengan  $y_i(w x_i + b) \geq 1, i = 1, 2, \dots, l$ .

Berdasarkan Persamaan (16) meminimalkan atau memaksimalkan kuantitas  $\|w\|^2$  atau  $w^T w$  dengan memperhatikan pembatas  $y_i(w x_i + b) \geq 1$ . Apabila output data  $y_i = +1$ , maka pembatas menjadi  $(w x_i + b) \geq 1$ . Begitu pula sebaliknya apabila  $y_i = -1$ , maka pembatas menjadi  $(w x_i + b) \leq -1$ . Permasalahan optimasi dengan pembatas ditentukan dengan mencari *saddle point* dari fungsi *Lagrangian* berikut.

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^l \alpha_i [y_i(w^T x_i + b) - 1] \quad (17)$$



Tabel 4.  
Hyperparameter Tuning XGBOOST

Parameter	Grid Search Values	Best Parameter
<i>max_depth</i>	8; 9; 10	9
<i>min_child_weight</i>	0; 1; 2	0
<i>gamma</i>	0; 0,1; 0,2; 0,3; 0,4	0,4
<i>subsample</i>	0,75; 0,8; 0,85	0,85
<i>colsample_bytree</i>	0,55; 0,6; 0,65	0,7
<i>reg_alpha</i>	$10^{-6}$ ; $10^{-5}$ ; $10^{-4}$	$1e-5$

Tabel 5.  
Confusion Matrix XGBOOST pada Data Training

		Kelas Prediksi	
		Positif	Negatif
Kelas Aktual	Positif	2036	3
	Negatif	14	591

yar cicilannya tiap bulan beserta bunganya. *Paylater* didasari oleh peraturan Otoritas Jasa Keuangan Nomor 29/POJK.05/2014 tentang Penyelenggara Usaha Perusahaan Pembiayaan. Sistem pembayaran *paylater* serupa dengan kartu kredit. Salah satu perbedaannya terdapat pada penyedia layanannya, *paylater* merupakan kerja sama antara sebuah platform *e-commerce* dengan sebuah perusahaan sedangkan pada kartu kredit disediakan oleh bank.

#### L. Kepuasan Pelanggan

Menurut Kamus Besar Bahasa Indonesia (KBBI) kepuasan berasal dari kata puas yang berarti kesenangan, kelegaan, dan lain sebagainya. Kepuasan timbul akibat seseorang membandingkan antara hasil kinerja dengan keinginannya, dan tingkat kepuasan merupakan hasil perbedaan antara harapan dengan kinerja. Apabila kinerja sesuai dengan harapan, maka akan timbul rasa puas. Begitu pula sebaliknya, apabila kinerja tidak sesuai harapan maka akan timbul rasa kecewa.

### III. METODE PENELITIAN

Pada penelitian ini, data yang digunakan adalah kumpulan ulasan yang diperoleh dari *Google Play Store* pada aplikasi keuangan, yaitu aplikasi *Kredivo*. Data ulasan diperoleh dengan menggunakan *Google Play Scraper* sebanyak 3000 ulasan terbaru yang diakses pada 20 Februari 2023.

Langkah-langkah pada penelitian ini adalah sebagai berikut: (1) melakukan klasifikasi sentimen, (2) melakukan *preprocessing text*, (3) melakukan visualisasi dari kata yang sering muncul, (4) menghitung frekuensi kemunculan kata pada tiap ulasan dengan menggunakan *Term Frequency-Inverse Document Frequency*, (5) membagi data menjadi data *training* dan *testing*, (6) melakukan klasifikasi *Support Vector Machine* dan *Extreme Gradient Boosting*, (7) membandingkan ketepatan klasifikasi model *Support Vector Machine* dan *Extreme Gradient Boosting*, dan terakhir (8) mengambil kesimpulan dari hasil pengolahan data.

### IV. HASIL DAN PEMBAHASAN

#### A. Karakteristik Data

Total data ulasan aplikasi yang terkumpul terdapat 98% ulasan yang mengandung sentimen dan sisanya tidak mengandung sentimen atau netral. Data ulasan yang mengandung sentimen diklasifikasikan ke dalam kelas positif dan negatif.

Berdasarkan Gambar 1 menunjukkan bahwa dari total data 2.939 ulasan yang mengandung sentimen terdapat 77% ulasan

Tabel 6.  
Confusion Matrix XGBOOST pada Data Testing

		Kelas Prediksi	
		Positif	Negatif
Kelas Aktual	Positif	218	8
	Negatif	11	56

Tabel 7.  
Ketepatan Klasifikasi XGBOOST

	Akurasi	F-score	AUC
<i>Training</i>	99,36%	99,59%	99,41%
<i>Testing</i>	93,33%	95,70%	91,07%

an mengandung sentimen positif dan sisanya 23% ulasan mengandung sentimen negatif. Sehingga dapat disimpulkan bahwa mayoritas pengguna aplikasi *Kredivo* merasa terbantu dan menyukai layanan *paylater* yang diberikan.

#### B. Preprocessing Text

Data ulasan aplikasi *paylater* yang mengandung sentimen kemudian akan dilakukan *preprocessing text* yang meliputi *cleaning*, *case folding*, *stemming*, *stopwords*, dan *tokenizing*. Pada tahap ini, data ulasan di proses hingga menjadi kata penting.

#### C. Visualisasi Data

Visualisasi data dengan menggunakan *Word Cloud* untuk mengetahui kata-kata yang sering muncul pada data. Frekuensi kemunculan kata dapat dilihat melalui ukuran huruf kata tersebut. Berikut merupakan visualisasi data ulasan aplikasi *paylater* pada sentimen negatif.

Pada Gambar 2 dapat dilihat kata yang sering muncul pada ulasan negatif adalah 'bayar', 'pinjam', dan 'akun'. Hal ini mengungkapkan bahwa aplikasi *paylater* mendapat sentimen negatif dari pengguna disebabkan adanya ketidakpuasan pengguna dengan layanan yang diberikan.

Pada Gambar 3 dapat dilihat bahwa dua kata yang sering muncul adalah kata 'bantu', 'kasih, dan 'mantap'. Hal ini mengungkapkan bahwa aplikasi *paylater* sudah membantu pengguna dengan memberikan pinjaman yang dapat meringankan keuangannya.

#### D. Term Frequency-Inverse Document Frequency

Pada proses ini kata-kata dalam ulasan diubah menjadi angka penting dengan proses faktorisasi teks. Proses faktorisasi teks dilakukan dengan mengalikan *Term Frequency* (TF) kata tersebut dengan *Inverse Document Frequency* (IDF). Berikut merupakan contoh hasil pembobotan kata dalam ulasan. Pada Tabel 3 dapat dilihat bahwa kata 'limit' pada ulasan pertama memiliki bobot sebesar 0,2182 dan kata 'lebih' memiliki bobot sebesar 0,2754. Pada ulasan kedua kata 'bunga' memiliki bobot sebesar 0,4345 dan kata 'cepat' memiliki bobot sebesar 0,3512. Pembobotan ini dilakukan hingga ulasan terakhir.

#### E. Klasifikasi menggunakan Extreme Gradient Boosting

Analisis klasifikasi menggunakan metode *XGBoost* menggunakan *hyperparameter tuning* dengan bantuan *gridsearch-CV*. *Hyperparameter tuning* yang digunakan adalah seperti pada Tabel 4. Tabel 4 merupakan nilai *hyperparameter* yang digunakan untuk mencari *hyperparameter* yang optimal dengan *gridsearchCV*. Untuk mengetahui nilai *hyperparameter* yang optimum dilakukan membagi data menjadi data *training* dan *testing* dengan menggunakan *stratified 10-fold cross*

Tabel 8.  
Pemilihan Parameter C Optimum

C	0,5	0,75	1	10	100
Rata-rata	0,9422	0,9432	0,9427	0,9317	0,9241

Tabel 9.  
Confusion Matrix SVM kernel Linier pada Data Training

Kelas Aktual	Positif	Kelas Prediksi	
		Positif	Negatif
Positif		1981	57
Negatif		11	594

Tabel 10.  
Confusion Matrix SVM Kernel Linier pada Data Testing

Kelas Aktual	Positif	Kelas Prediksi	
		Positif	Negatif
Positif		217	10
Negatif		7	60

Tabel 11.  
Ketepatan Klasifikasi SVM Kernel Linier

	Akurasi	F-score	AUC
Training	97,43%	98,32%	95,34%
Testing	94,35%	96,31%	91,56%

validation. Di mana data ulasan sebanyak 2.938 dibagi menjadi 10 subset yang terdiri dari 10% data testing dan 90% data training yang berbeda dengan proporsi kelas yang sama, sehingga pada setiap subset memiliki data training sebanyak 2644 ulasan dan data testing sebanyak 294 ulasan.

Berdasarkan karakteristik data diketahui bahwa jumlah ulasan yang bersentimen positif lebih banyak daripada negatif. Persentase perbandingan antara ulasan dengan sentimen positif dan negatif adalah 77:23 di mana angka tersebut menunjukkan bahwa data imbalance. Oleh karena itu dilakukan resampling dengan metode Synthetic Minority Oversampling Technique (SMOTE). Di mana jumlah ulasan bersentimen negatif akan di replikasi untuk menyamakan jumlah dengan ulasan bersentimen positif. Data training yang pada awalnya berisi ulasan bersentimen positif sebanyak 2.036 ulasan dan negatif sebanyak 608 ulasan akan disamakan menjadi 2.036 ulasan bersentimen positif dan 2036 ulasan bersentimen negatif. Data training setelah dilakukan SMOTE dapat digunakan untuk analisis klasifikasi.

Kemudian data training setelah SMOTE digunakan untuk analisis klasifikasi menggunakan metode XGBOOST dengan melakukan hyperparameter tuning untuk mendapatkan nilai hyperparameter yang optimum perlu dilakukan perhitungan ketepatan klasifikasi pada masing-masing parameter yang dicoba. Sebagaimana yang telah dijelaskan sebelumnya bahwa pembagian data training dan testing dilakukan dengan stratified 10-fold cross validation, sehingga dapat diperoleh best hyperparameter yang telah di cantumkan pada Tabel 4, sehingga didapatkan confusion matrix pada data training yang digambarkan sebagai berikut.

Berdasarkan Tabel 5 dapat dilihat bahwa sebanyak 2.036 data positif yang diklasifikasikan benar positif, sedangkan terdapat 3 data positif diklasifikasikan negatif. Data negatif yang diklasifikasikan benar negatif adalah sebanyak 591 dan terdapat 14 data negatif yang diklasifikasikan positif. Selanjutnya, yaitu hasil klasifikasi XGBOOST pada data testing digambarkan pada confusion matrix di Tabel 6.

Berdasarkan Tabel 6 dapat dilihat bahwa sebanyak 218 data positif yang diklasifikasikan benar positif, sedangkan terdapat 8 data positif diklasifikasikan negatif. Data negatif

Tabel 12.  
Confusion Matrix SVM Kernel RBF pada Data Training

Kelas Aktual	Positif	Kelas Prediksi	
		Positif	Negatif
Positif		2016	23
Negatif		4	601

Tabel 13.  
Confusion Matrix SVM Kernel RBF pada Data Testing

Kelas Aktual	Positif	Kelas Prediksi	
		Positif	Negatif
Positif		219	8
Negatif		9	59

Tabel 14.  
Ketepatan Klasifikasi SVM Kernel RBF

	Akurasi	F-score	AUC
Training	99,00%	99,35%	98,07%
Testing	94,45%	96,41%	92,34%

yang diklasifikasikan benar negatif adalah sebanyak 56 dan terdapat 11 data negatif yang diklasifikasikan positif. Berdasarkan confusion matrix dapat diketahui ketepatan klasifikasi sebagai pada Tabel 7.

Tabel 7 dapat dilihat bahwa analisis klasifikasi dengan menggunakan metode XGBOOST pada data training menghasilkan rata-rata nilai akurasi sebesar 99,36%. Pada rata-rata nilai F<sub>1</sub>-score sebesar 99,59% maka dapat disimpulkan bahwa model memiliki nilai presisi dan recall yang tinggi sehingga dapat mengklasifikasikan kelas positif dan negatif dengan baik. Pada rata-rata nilai AUC sebesar 99,41% maka dapat disimpulkan model mampu mengklasifikasi kelas positif dan negatif mendekati sempurna. Kemudian ketetapan klasifikasi pada data testing menghasilkan rata-rata nilai akurasi sebesar 93,33%. Pada rata-rata nilai F<sub>1</sub>-score sebesar 95,70% maka dapat disimpulkan bahwa model memiliki nilai presisi dan recall yang tinggi sehingga dapat mengklasifikasikan kelas positif dan negatif dengan baik. Pada rata-rata nilai AUC sebesar 89,77% maka dapat disimpulkan model mampu mengklasifikasi kelas positif dan negatif dengan sangat baik.

#### F. Klasifikasi Menggunakan Support Vector Machine

Pada penelitian ini menggunakan dua jenis kernel, yaitu linier dan Radial Basis Function (RBF). Pada analisis SVM dengan kernel linier membutuhkan parameter cost (C), di mana nilai parameter C yang akan diuji coba adalah 0,5; 0,75; 1; 10 dan 100. Pada analisis SVM dengan kernel RBF menggunakan parameter C dan  $\gamma$ , dimana nilai parameter  $\gamma$  yang akan diuji coba adalah 0,005; 0,05; 0,1; 0,5 dan 0,75. Untuk mengetahui nilai parameter yang optimum dilakukan pembagian data menjadi data training dan testing dengan menggunakan stratified 10-fold cross validation. Di mana data ulasan sebanyak 2938 dibagi menjadi 10 subset yang terdiri dari 10% data testing dan 90% data training yang berbeda dengan proporsi kelas yang sama, sehingga pada setiap subset memiliki data training sebanyak 2.644 ulasan dan data testing sebanyak 294 ulasan.

Berdasarkan karakteristik data diketahui bahwa jumlah ulasan yang bersentimen positif lebih banyak daripada negatif. Persentase perbandingan antara ulasan dengan sentimen positif dan negatif adalah 77:23 di mana angka tersebut menunjukkan bahwa data imbalance. Oleh karena itu dilakukan resampling dengan metode Synthetic Minority Oversampling Technique (SMOTE). Di mana jumlah ulasan ber-

Tabel 15.  
Perbandingan Klasifikasi pada Data *Training*

Model	Akurasi	F-score	AUC
XGBoost	93,33%	95,70%	91,18%
SVM Linear	94,35%	96,31%	91,61%
SVM Non-Linear	94,45%	96,41%	92,39%

Tabel 16.  
Perbandingan Klasifikasi pada Data *Testing*

Model	Akurasi	F-score	AUC
XGBoost	93,33%	95,70%	91,18%
SVM Linier	94,35%	96,31%	91,61%
SVM Non-Linier	94,45%	96,41%	92,39%

sentimen negatif akan di replikasi untuk menyamakan jumlah dengan ulasan bersentimen positif. Data *training* yang pada awalnya berisi ulasan bersentimen positif sebanyak 2036 ulasan dan negatif sebanyak 608 ulasan akan disamakan menjadi 2.036 ulasan bersentimen positif dan 2.036 ulasan bersentimen negatif. Data *training* setelah dilakukan SMOTE dapat digunakan untuk analisis klasifikasi.

### 1) Support Vector Machine Linier

Dengan menggunakan langkah yang telah dijelaskan sebelumnya dilakukan klasifikasi SVM *kernel* linier dengan menentukan nilai parameter C yang optimum. Tabel 8 menunjukkan bahwa parameter  $C = 0,75$  memiliki rata-rata nilai akurasi tertinggi. Selanjutnya dilakukan pengujian ketepatan klasifikasi sehingga didapatkan *confusion matrix* pada data *training* sebagai pada Tabel 9.

Berdasarkan Tabel 9 dapat dilihat bahwa sebanyak 1981 data positif yang diklasifikasikan benar positif, sedangkan terdapat 57 data positif diklasifikasikan negatif. Data negatif yang diklasifikasikan benar negatif adalah sebanyak 594 dan terdapat 11 data negatif yang diklasifikasikan positif. Selanjutnya, yaitu hasil klasifikasi SVM *kernel* linier pada data *testing* digambarkan pada *confusion matrix* sebagai berikut.

Tabel 10 menunjukkan bahwa sebanyak 217 data positif yang diklasifikasikan benar positif, sedangkan terdapat 10 data positif diklasifikasikan negatif. Data negatif yang diklasifikasikan benar negatif adalah sebanyak 60 dan terdapat 7 data negatif yang diklasifikasikan positif. Berdasarkan *confusion matrix* dapat diketahui ketepatan klasifikasi sebagai pada Tabel 11.

Berdasarkan Tabel 11 dapat dilihat bahwa analisis klasifikasi dengan menggunakan metode SVM *kernel* Linear pada data *training* menghasilkan rata-rata nilai akurasi sebesar 97,43%. Pada rata-rata nilai  $F_1$ -score sebesar 98,32% maka dapat disimpulkan bahwa model memiliki nilai presisi dan *recall* yang tinggi sehingga dapat mengklasifikasikan kelas positif dan negatif dengan baik. Pada rata-rata nilai AUC sebesar 95,34% maka dapat disimpulkan model mampu mengklasifikasi kelas positif dan negatif mendekati sempurna. Kemudian ketetapan klasifikasi pada data *testing* menghasilkan rata-rata nilai akurasi sebesar 94,35%. Pada rata-rata nilai  $F_1$ -score sebesar 96,31% maka dapat disimpulkan bahwa model memiliki nilai presisi dan *recall* yang tinggi sehingga dapat mengklasifikasikan kelas positif dan negatif dengan baik. Pada rata-rata nilai AUC sebesar 91,56% maka dapat disimpulkan model mampu mengklasifikasi kelas positif dan negatif dengan sangat baik.

### 2) Support Vector Machine Non-Linier

Dengan menggunakan langkah yang sama dilakukan klasifikasi SVM *kernel* RBF dengan menentukan nilai parameter

C dan  $\gamma$  yang optimum. Kemudian didapatkan bahwa pada parameter  $\gamma = 0,75$  dengan parameter  $C = 1$  menghasilkan ketepatan klasifikasi yang sama tinggi, yaitu sebesar 0,9443, sehingga dapat disimpulkan bahwa parameter adalah  $C = 1$  dan  $\gamma = 0,75$  optimum pada klasifikasi data ulasan aplikasi *paylater* menggunakan metode SVM dengan *kernel* RBF. Selanjutnya hasil klasifikasi pada data *training* yang digambarkan *confusion matrix* sebagai pada Tabel 12.

Berdasarkan Tabel 12 dapat dilihat bahwa sebanyak 2016 data positif yang diklasifikasikan benar positif, sedangkan terdapat 23 data positif diklasifikasikan negatif. Data negatif yang diklasifikasikan benar negatif adalah sebanyak 601 dan terdapat 4 data negatif yang diklasifikasikan positif. Selanjutnya, yaitu hasil klasifikasi SVM *kernel* RBF pada data *testing* digambarkan pada *confusion matrix* sebagai pada Tabel 13.

Tabel 13 menunjukkan bahwa sebanyak 219 data positif yang diklasifikasikan benar positif, sedangkan terdapat 8 data positif diklasifikasikan negatif. Data negatif yang diklasifikasikan benar negatif adalah sebanyak 59 dan terdapat 9 data negatif yang diklasifikasikan positif. Dari *confusion matrix* dapat diketahui ketepatan klasifikasi sebagai pada Tabel 14.

Berdasarkan Tabel 14 dapat dilihat bahwa analisis klasifikasi dengan menggunakan metode SVM *kernel* RBF pada data *training* menghasilkan rata-rata nilai akurasi sebesar 99,00%. Pada rata-rata nilai  $F_1$ -score sebesar 99,35% maka dapat disimpulkan bahwa model memiliki nilai presisi dan *recall* yang tinggi sehingga dapat mengklasifikasikan kelas positif dan negatif dengan baik. Pada rata-rata nilai AUC sebesar 98,07% maka dapat disimpulkan model mampu mengklasifikasi kelas positif dan negatif mendekati sempurna. Kemudian ketetapan klasifikasi pada data *testing* menghasilkan rata-rata nilai akurasi sebesar 94,45%. Pada rata-rata nilai  $F_1$ -score sebesar 96,41% maka dapat disimpulkan bahwa model memiliki nilai presisi dan *recall* yang tinggi sehingga dapat mengklasifikasikan kelas positif dan negatif dengan baik. Pada rata-rata nilai AUC sebesar 92,34% maka dapat disimpulkan model mampu mengklasifikasi kelas positif dan negatif dengan sangat baik.

### G. Perbandingan Support Vector Machine dan Extreme Gradient Boosting

Setelah memperoleh nilai ketepatan klasifikasi dari parameter optimum dari setiap metode klasifikasi. Selanjutnya akan dilakukan perbandingan nilai rata-rata ketepatan klasifikasi untuk memperoleh metode klasifikasi yang baik digunakan untuk data ulasan. Berikut merupakan ringkasan dari hasil nilai rata-rata akurasi,  $F_1$ -score, dan AUC untuk setiap metode.

Berdasarkan Tabel 15 dapat dilihat bahwa ketiga model memiliki nilai AUC yang besar, yang artinya ketiga metode dapat mengklasifikasikan data ulasan aplikasi *paylater* mendekati sempurna. Pada metode XGBOOST memiliki nilai akurasi dan  $F_1$ -score tertinggi kemudian diikuti SVM non-linier dengan *kernel* RBF dan SVM linier. Selanjutnya adalah perbandingan ketepatan klasifikasi pada data *testing* seperti pada Tabel 16.

Berdasarkan Tabel 16 dapat dilihat bahwa ketiga model memiliki nilai AUC yang besar yang artinya ketiga metode dapat mengklasifikasikan data ulasan aplikasi *paylater* mendekati sempurna. Pada metode SVM Non-Linier dengan *kernel* RBF memiliki nilai akurasi dan  $F_1$ -score tertinggi.

Pada metode XGBOOST pada data *training* menghasilkan ketepatan klasifikasi tertinggi, namun pada data *testing* menghasilkan ketepatan klasifikasi yang lebih rendah dari metode SVM. Hal ini membuktikan bahwa metode SVM baik digunakan untuk mengklasifikasikan data teks daripada metode XGBOOST.

#### V. KESIMPULAN/RINGKASAN

Berdasarkan penelitian yang telah dilakukan, didapatkan kesimpulan sebagai berikut. (1) Pengguna aplikasi *paylater*, yaitu aplikasi Kredivo didominasi dengan ulasan yang mengandung sentimen positif sebesar 77% dan untuk ulasan dengan sentimen negatif sebanyak 23%. (2) Hasil rata-rata ketepatan klasifikasi pada metode *Support Vector Machine* Linier, *Support Vector Machine* Non-Linier, dan *Extreme Gradient Boosting* masing-masing sebesar 94,35%; 94,45% dan 93,33%. (3) Perbandingan ketepatan klasifikasi pada metode *Support Vector Machine* Linier, *Support Vector Machine* Non-Linier, dan *Extreme Gradient Boosting* menunjukkan bahwa metode *Support Vector Machine* memiliki ketepatan klasifikasi yang lebih baik daripada metode *Extreme Gradient Boosting* dalam mengklasifikasikan data ulasan aplikasi *paylater*, yaitu Kredivo.

Saran yang dapat diberikan kepada peneliti selanjutnya dapat menggunakan metode *Support Vector Machine* untuk mengklasifikasikan data teks dan diharapkan untuk menambahkan atau mengatasi kata-kata khusus pada proses *stem-*

*ming* karena terdapat beberapa kata yang tidak terproses dengan baik.

#### DAFTAR PUSTAKA

- [1] R. W. Mayasari, "Text Mining pada Akun Pemerintah Kota Surabaya dengan Metode Regresi Logistik, Support Vector Machine (SVM), dan Naïve Bayes Classifier," Departemen Statistika, Institut Teknologi Sepuluh Nopember, Surabaya, 2018.
- [2] K. Afifah, I. N. Yulita, dan I. Sarathan, "Sentiment analysis on telemedicine app reviews using xgboost classifier," in *2021 International Conference on Artificial Intelligence and Big Data Analytics*, Medan, 2021, hal. 22–27. doi: 10.1109/ICAIBDA53487.2021.9689762.
- [3] S. E. H. Yulianti dan O. Soesanto, "Penerapan metode extreme gradient boosting (XGBoost) pada klasifikasi nasabah kartu kredit," *J. Math. Theory Appl.*, vol. 4, no. 1, hal. 21–26, 2022, doi: <https://doi.org/10.31605/jomta.v4i1.1792>.
- [4] M. Awad dan R. Khanna, *Efficient Learning Machines: Theories, Concept, and Applications for Engineers and System Designers*. New York, USA: Apress Media, 2015. ISBN: 978-1-4302-5990-9.
- [5] H. Dalianis, *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Cham, Switzerland: Springer International Publishing, 2018. ISBN: 978-3-319-78503-5.
- [6] M. L. Osval Antonio, M. L. Abelardo, dan C. José, *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Cham, Switzerland: Springer Nature, 2022. ISBN: 978-3-030-89010-0.
- [7] T. Chen dan C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 2016, hal. 785–794. doi: 10.1145/2939672.2939785.
- [8] C. C. Aggarwal, *Data Mining*, 1st ed. Cham, Switzerland: Springer International Publishing, 2015. ISBN: 978-3-319-14142-8.
- [9] Q. Castellà dan C. Sutton, "Word storms: multiples of word clouds for visual comparison of documents," in *Proceedings of the 23rd International Conference on World Wide Web*, New York, 2014, hal. 665–676. doi: 10.1145/2566486.2567977.