

Perbandingan Performa Metode Berbasis *Support Vector Machine* untuk Penanganan Klasifikasi Multi Kelas Tidak Seimbang

QORRY MEIDIANINGSIH¹, DEVI EKA WARDANI², ELLIS SALSABILA³, LINA NAFISAH⁴, AFIFAH NUR MUTIA⁵

^{1,3}) Program Studi Pendidikan Matematika Fakultas MIPA Universitas Negeri Jakarta, Indonesia

²) Program Studi Matematika Fakultas MIPA Universitas Negeri Jakarta, Indonesia

^{4,5}) Program Studi Statistika Fakultas MIPA Universitas Negeri Jakarta, Indonesia

e-mail: qorrymeidianingsih@unj.ac.id

ABSTRAK

Permasalahan data multi kelas tidak seimbang mulai mendapatkan perhatian dari komunitas peneliti dalam beberapa tahun terakhir. Permasalahan klasifikasi pada kasus multi kelas tidak seimbang menjadi lebih rumit karena sebagian besar teknik klasifikasi multi kelas diterapkan pada kondisi kelas yang seimbang, sedangkan dalam realisasinya data yang ditemukan lebih sering memiliki kelas tidak seimbang. Penelitian ini fokus pada membandingkan performa tiga metode klasifikasi berbasis *support vector machine*, yaitu SVM standar, SVM-SMOTE, dan *granular support vector machines-repetitive undersampling* (GSVM-RU) dimana metode dekomposisi *one-versus-one* (OVO) diterapkan. Terdapat tiga jenis data hasil bangkitan *software* R yang dirancang berdasarkan kombinasi jumlah kelas mayoritas dan minoritas yang mungkin terjadi. Hasil penelitian menunjukkan bahwa ketiga model klasifikasi menunjukkan tingkat akurasi tertinggi pada data simulasi yang memiliki perbandingan persentase antara jumlah amatan kelas mayoritas dan minoritasnya paling tinggi. Berdasarkan kriteria sensitivitas dan spesifisitas, model klasifikasi SVM standar dan SVM-SMOTE memberikan performa yang sama baiknya pada kelas mayoritas, sedangkan model klasifikasi GSVM-RU memiliki performa yang baik dalam mendeteksi kelas minoritas.

Kata Kunci: support vector machine, imbalanced multi-class, smote, granular support vector machines-repetitive undersampling, confusion matrix

ABSTRACT

The problem of data with imbalances in multi-class has begun to receive attention from the research community in recent years. Classification problems in imbalanced multi-class cases become more complicated because most of the classification techniques in multi-class are applied to balanced class conditions, whereas in reality, the data found more often have unbalanced classes. This study focuses on comparing the performance of three support vector machine-based classification methods, namely standard SVM, SVM-SMOTE, and granular support vector machines-repetitive undersampling (GSVM-RU) where the one-versus-one (OVO) decomposition method is applied. There are three types of data generated by R software that are designed based on a combination of the number of possible majority and minority classes. The results showed that the three classification models showed the highest level of accuracy in the simulation data which had the highest percentage comparison between the number of observations of the majority and minority classes. Based on the sensitivity and specificity criteria, the standard SVM and SVM-SMOTE classification models provide equally good performance in the majority class, while the GSVM-RU classification model has good performance in detecting the minority class.

Keywords: support vector machine, imbalanced multi-class, smote, granular support vector machines-repetitive undersampling, confusion matrix

1. PENDAHULUAN

Permasalahan klasifikasi menjadi kajian yang banyak dipelajari oleh komunitas *data mining* dan *machine learning*. Permasalahan utama yang ditemui pada algoritma *machine learning* adalah ketidakseimbangan amatan di antara kelas yang ada. Beberapa penelitian telah dilakukan dalam pengembangan teknik untuk kasus data tidak seimbang. He dan Garcia (2009) serta Galar *et al.* (2011) menjelaskan beberapa metode yang dapat digunakan untuk mengatasi masalah data tidak seimbang. Metode klasifikasi biasa tidak dapat menghasilkan keputusan akhir yang tepat ketika kondisi kelasnya tidak seimbang. Hal ini disebabkan salah satu kelas memiliki amatan yang lebih banyak sehingga akan menghasilkan kesalahan klasifikasi. Beberapa pendekatan solusi yang terkenal dilakukan dalam menangani permasalahan data dengan kelas tidak seimbang, di antaranya pada level data dan memodifikasi algoritma metode klasifikasi standar. Selama tujuan klasifikasi adalah memprediksi suatu objek ke dalam suatu kelas, maka sangat dibutuhkan model klasifikasi yang menghasilkan prediksi dengan ketepatan tinggi pada amatan kelas minoritas. Bagaimanapun, metode klasifikasi standar cenderung menghasilkan performa yang baik pada kelas mayoritas karena mereka membangun model klasifikasi berdasarkan kesalahan pada keseluruhan data latih dimana tidak bergantung di antara dua kelas.

Dalam beberapa tahun terakhir, multi kelas tidak seimbang (*imbalanced multi-class*) mulai mendapatkan perhatian dari komunitas peneliti (Lango, 2019). Literatur mengenai *pattern recognition* banyak membahas teknik klasifikasi pola pada multi kelas, namun sebagian besar teknik tersebut diterapkan pada data yang seimbang (Ghanem *et al.*, 2010). Dalam menangani permasalahan multi kelas tidak seimbang, gugus data yang dihadapi memiliki konfigurasi yang berbeda, contohnya seperti kemungkinan untuk memiliki kelas mayoritas lebih dari satu dan sebaliknya dimana hanya terdiri dari satu kelas mayoritas dan beberapa kelas minoritas (Esteves, 2020). Beberapa metode standar tidak dapat diterapkan secara langsung pada kasus multi kelas tidak seimbang. Hal ini lah yang membuat permasalahan klasifikasi pada kasus multi kelas tidak seimbang menjadi lebih rumit. Metode yang digunakan harus mencoba untuk menangkap dan menggali lebih jauh hubungan antara beberapa kelas yang tidak seimbang serta siap untuk bekerja dengan ketidakseimbangan yang lebih ekstrim karena mereka lebih mungkin terjadi dalam gugus data multi kelas (Lango, 2019). Dalam mengatasi permasalahan klasifikasi multi kelas, salah satu metode yang populer untuk digunakan adalah metode dekomposisi. Metode tersebut diterapkan untuk membagi gugus data multi kelas menjadi multipel sub-kelas biner sehingga nantinya akan menghasilkan pengklasifikasi biner sebanyak sub-kelas.

Support vector machine (SVM) merupakan salah satu metode klasifikasi yang sering kali menjadi pilihan dalam mengatasi permasalahan ketidakseimbangan kelas. Hal ini disebabkan oleh SVM membangun model klasifikasi hanya bergantung pada subset amatan data latih (Núñez *et al.*, 2017). Selain itu, terdapat alasan bahwa SVM merupakan salah satu metode yang kini banyak mendapat perhatian sebagai *state of the art* dalam klasifikasi *pattern recognition* (Azis *et al.*, 2017). Meskipun begitu, seperti halnya metode klasifikasi yang lain, SVM meminimalisir *error* dari keseluruhan gugus data dalam membangun model klasifikasi sehingga terdapat bias terhadap kelas mayoritas ketika tingkat ketidakseimbangan cukup tinggi.

Penelitian ini fokus pada mengkaji dan membandingkan kinerja beberapa metode klasifikasi berbasis *support vector machine* dalam menangani permasalahan multi kelas tidak seimbang. Metode klasifikasi yang dipilih merupakan pengembangan dari metode klasifikasi standar *support vector machine* yang telah diusulkan berdasarkan beberapa penelitian sebelumnya. Dengan demikian hasil penelitian ini diharapkan dapat memberikan alternatif lain dalam mengatasi kasus klasifikasi untuk multi kelas tidak seimbang.

2. METODE PENELITIAN

2.1. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) merupakan salah satu metode klasifikasi yang mendapat banyak perhatian dalam *pattern recognition*. SVM dikembangkan oleh Boser, Guyon, dan Vapnik (Boser *et al.*, 1992). Tujuan utama dari metode SVM adalah memisahkan gugus data ke dalam dua kelas secara efektif dengan memaksimalkan jarak antara *hyperplane* atau pemisah dengan titik terdekat (*support vector*) pada setiap kelas sehingga akan diperoleh *hyperplane* yang optimal. Penentuan *hyperplane* bergantung kepada struktur data yang dimiliki, seperti apakah datanya terpisahkan secara linier atau nonlinier.

Tahapan yang dilakukan pada SVM ketika gugus datanya terpisahkan secara nonlinier adalah mentransformasikan data ke dimensi yang lebih tinggi menggunakan pemetaan nonlinier dan menemukan pemisah linier (*hyperplane*) pada ruang dimensi yang baru tersebut (Han dan Kamber, 2006), sehingga datanya dapat dipisahkan secara linier. Fungsi nonlinier yang berfungsi memetakan data amatan ke ruang dimensi yang lebih tinggi dinotasikan dengan ϕ . Misalkan fungsi nonlinier $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}_1), \phi_2(\mathbf{x}_2), \dots, \phi_3(\mathbf{x}_3))$ memetakan vektor \mathbf{x} di ruang input ke dalam ruang dimensi yang lebih tinggi, maka secara umum *hyperplane* yang optimal dalam ruang berdimensi tinggi didefinisikan sebagai berikut:

$$f(x, \alpha_i) = \sum_{i=1}^q \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b \quad \dots (1)$$

dengan \mathbf{x}_i adalah *support vector* yang berjumlah q . Berbeda dengan kasus gugus data yang terpisahkan linier secara sempurna, pada kasus ini vektor \mathbf{w} merupakan kombinasi linier dari *support vector* yang ada di ruang berdimensi tinggi (*feature space*).

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \phi(x_i) \quad \dots (2)$$

Persamaan di atas menunjukkan bahwa *hyperplane* yang dicari bergantung kepada perkalian antara $\phi(\mathbf{x}_i)$ dan $\phi(\mathbf{x}_j)$. Apabila perkalian tersebut dilakukan berulang kali untuk menemukan *hyperplane* yang optimal akan menimbulkan perhitungan yang berat dan mahal. *Kernel trick* membantu dalam mengatasi hal tersebut dimana memiliki fungsi sebagai berikut:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \quad \dots (3)$$

Kernel trick memberikan kemudahan dalam proses pencarian *support vector* tanpa tahu seperti apa wujud fungsi nonlinier ϕ . Oleh karena itu, dengan menerapkan *kernel trick* pada permasalahan fungsi optimasi *Lagrange multiplier* di ruang dimensi yang lebih tinggi diperoleh persamaan sebagai berikut:

$$L(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \left(\sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right) \quad \dots (4)$$

dengan kendala $\sum_{i=1}^m y_i \alpha_i = 0$ dan $0 \leq \alpha_i \leq C$. Nilai C merupakan besarnya pinalti akibat adanya kesalahan klasifikasi. Dengan cara yang sama maka formula untuk *hyperplane* optimal pada ruang berdimensi tinggi adalah sebagai berikut:

$$f(x, \alpha_i) = \sum_{i=1}^q \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b \quad \dots (5)$$

Fungsi kernel yang umum digunakan dalam SVM (Meyer, 2015) adalah sebagai berikut:

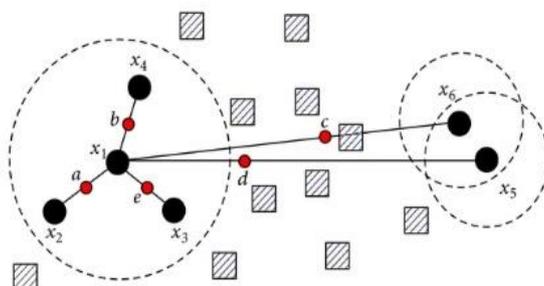
Tabel 1. Fungsi Kernel dalam *Support Vector Machine*

Jenis Kernel	Fungsi Kernel ^a	Keterangan
Polinomial	$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i' \cdot \mathbf{x}_j + r)^d$	d : derajat polinom r : parameter penyeimbang pengaruh orde tinggi dan rendah
Radial basis (RBF)	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \mathbf{x}_i - \mathbf{x}_j ^2)$	γ : parameter positif pengontrol radius
Sigmoid	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i' \cdot \mathbf{x}_j + r)$	γ : parameter penentu penskalaan data <i>input</i> r : parameter pengendali batas pemetaan

Penelitian ini membatasi fungsi kernel yang digunakan, yaitu radial basis (RBF).

2.2. SVM-SMOTE

Synthetic Minority Oversampling Technique (SMOTE) merupakan teknik *oversampling* yang diperkenalkan oleh Chawla *et al.* pada tahun 2002. SMOTE merupakan salah satu cara untuk menyeimbangkan jumlah amatan kelas dengan cara membangkitkan data sintetis di sekitar amatan kelas minoritas.



Gambar 1. Ilustrasi Kinerja SMOTE
(Hu dan Li, 2013)

Pada SMOTE, kelas minoritas akan ditambah dengan data buatan yang diyakini lebih baik dibandingkan dengan mereplikasi data yang ada sehingga setara dengan kelas mayoritas. Pembentukan data buatan didasarkan pada karakteristik tetangga terdekat (*k-nearest neighbor*). Tahapan dalam membangkitkan data buatan bagi peubah kontinu adalah sebagai berikut:

1. Tentukan k tetangga terdekat untuk setiap amatan pada kelas minoritas menggunakan jarak *Euclidean*. Berikut adalah formula jarak *Euclidean*

$$\Delta(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})'(\mathbf{x} - \mathbf{y})} \quad \dots (6)$$

2. Tentukan 1 dari k -tetangga terdekat tersebut secara acak.
3. Bangkitkan data buatan dengan rumus sebagai berikut:

$$x_{baru} = x_i + (\hat{x}_i - x_i) \times \delta. \quad \dots (7)$$

Keterangan:

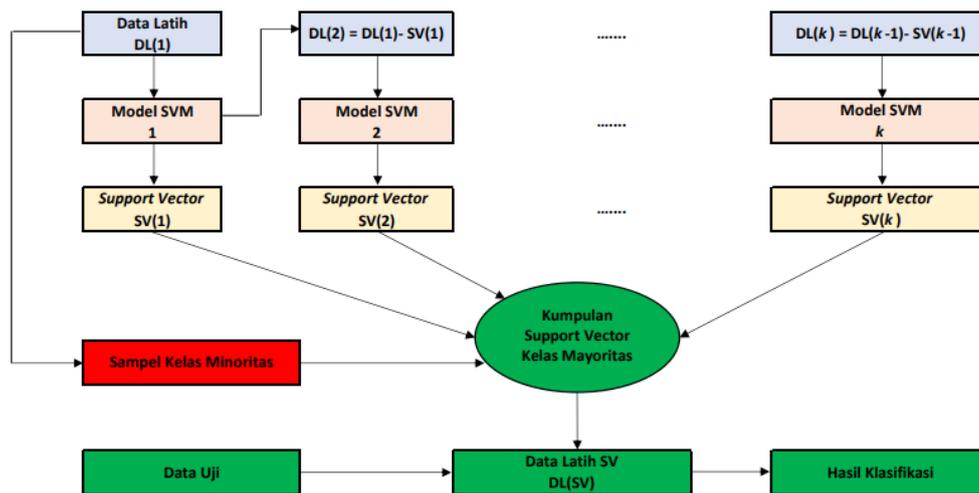
- x_{baru} : data buatan
- x_i : data pada kelas minoritas
- \hat{x}_i : 1 dari k tetangga terdekat yang terpilih
- δ : bilangan acak antara 0 dan 1

SVM-SMOTE merupakan penerapan *support vector machine* setelah SMOTE diterapkan. Dengan kondisi data yang lebih seimbang, maka *support vector machine* standar dapat digunakan.

2.3. Granular Support Vector Machines – Repetitive Undersampling (GSVM-RU)

GSVM-RU merupakan strategi *undersampling* berulang yang dapat menyeimbangkan kembali kumpulan data. Ide dasar GSVM-RU memaknai bahwa model SVM hanya membutuhkan *support vector* yang informatif untuk klasifikasi sehingga sampel lainnya dapat dihapus atau dibuang. Namun, untuk klasifikasi yang sangat tidak seimbang, kelas mayoritas cenderung mendorong *hyperplane* menuju kelas minoritas sehingga *support vector* yang dekat dengan batas yang dipelajari mungkin bukan yang paling informatif. *Undersampling* secara acak merupakan pendekatan umum untuk menyeimbangkan ulang gugus data untuk mencapai distribusi data yang lebih baik. Namun, teknik *undersampling* sering kali menimbulkan kehilangan informasi. GSVM-RU ditargetkan untuk langsung memanfaatkan SVM itu sendiri untuk proses

undersampling. *Support vector* yang diperoleh dari pemodelan data latih menggunakan SVM dianggap memiliki informasi yang penting atau disebut sebagai *granule*. Kemudian *support vector* yang berasal dari kelas mayoritas ini dihapus dari gugus data latih asli untuk menghasilkan gugus data latih yang lebih kecil (sub data latih) dimana SVM baru dimodelkan untuk mengekstrak grup *support vector* kelas mayoritas lainnya. Selanjutnya seluruh *support vector* dari kelas mayoritas yang dianggap mengandung informasi (*granule*) digabung dan membentuk data latih baru. Proses *undersampling* berulang ini dihentikan ketika *support vector* yang baru tidak dapat meningkatkan kinerja klasifikasi jika digabungkan dengan kumpulan data agregasi sebelumnya. Tahapan alur GSVM-RU menurut Tang et al. (2002) dapat dilihat pada Gambar 2 berikut ini.



Gambar 2. Tahapan GSVM-RU

Tahapan GSVM-RU adalah sebagai berikut:

1. Menentukan data latih ke-1 atau $DL(1)$.
2. Menerapkan *Support Vector Machine* (SVM) pada $DL(1)$.
3. Mengidentifikasi *support vector* yang dihasilkan berdasarkan $DL(1)$, beri notasi $SV(1)$.
4. Simpan $SV(1)$ sebagai gugus data baru (*pada bulatan warna hijau*).
5. Menentukan data latih ke-2 atau $DL(2)$, yaitu dengan mengurangi $DL(1)$ dengan amatan yang teridentifikasi sebagai *support vector*.
6. Menerapkan *Support Vector Machine* (SVM) pada $DL(2)$.
7. Mengidentifikasi *support vector* yang dihasilkan berdasarkan $DL(2)$, beri notasi $SV(2)$.
8. Lakukan kembali tahap 1 sampai 4 untuk $DL(3)$, $DL(4)$, dan seterusnya hingga *support vector* yang baru tidak dapat meningkatkan kinerja klasifikasi jika digabungkan dengan kumpulan data agregasi sebelumnya.
9. Membentuk data latih baru, yaitu gabungan dari amatan kelas minoritas dan *support vector* kelas mayoritas (dinotasikan $DL(SV)$).

Menurut Tang et al. (2002), GSVM-RU dapat meningkatkan kinerja klasifikasi dengan mengekstraksi sampel yang informatif yang penting untuk klasifikasi dan menghilangkan sejumlah besar sampel yang berlebihan atau bahkan bersifat *noise*.

2.4. Ukuran Performa Klasifikasi

Matriks evaluasi memiliki peranan penting dalam *machine learning*. Matriks yang dikenal dengan nama *confusion matrix* tersebut digunakan untuk mengevaluasi suatu algoritma pembelajaran (Phung et al. 2009). Setiap kolom pada matriks tersebut menunjukkan jumlah amatan pada kelas yang sesungguhnya, sedangkan baris pada matriks tersebut menunjukkan jumlah amatan pada kelas yang diprediksi.

Tabel 2. *Confusion Matrix* untuk *Multi-class Classification*

Prediksi	Aktual		
	1	2	3
1	<i>a</i>	<i>b</i>	<i>c</i>
2	<i>d</i>	<i>e</i>	<i>f</i>
3	<i>g</i>	<i>h</i>	<i>i</i>

Pada kasus *multiclass classification*, nilai *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN) tidak diperoleh langsung seperti kasus klasifikasi biner. Nilai-nilai tersebut diperoleh dengan mengasumsikan salah satu kelas menjadi kelas “positif” dan kelas lainnya menjadi kelas “negatif”. Misalkan, untuk menilai performa metode klasifikasi pada kelas 1, maka kelas 1 sebagai kelas “positif” sedangkan kelas 2 dan 3 menjadi kelas “negatif”.

Tabel 3. Contoh Pengolahan *Confusion Matrix* untuk Kelas 1

Kategori	Nilai
<i>True Positive</i> (TP)	<i>a</i>
<i>True Negatif</i> (TN)	<i>e + f + h + i</i>
<i>False Positive</i> (FP)	<i>b + c</i>
<i>False Negative</i> (FN)	<i>d + g</i>

True Positives (TP) adalah banyaknya contoh dari kelas positif dan diklasifikasikan ke dalam kelas positif (*correctly classified*). *False Positives* (FP) adalah banyaknya contoh dari kelas negatif namun diklasifikasikan ke dalam kelas positif (*incorrectly classified*). *False Negatives* (FN) adalah banyaknya contoh dari kelas positif namun diklasifikasikan ke dalam kelas negatif (*incorrectly classified*). *True Negative* (TN) adalah banyaknya contoh dari kelas negatif dan diklasifikasikan ke dalam kelas negatif (*correctly classified*). Dengan demikian, nilai TP, TN, FP, dan FN untuk kelas 1 (positif) secara berturut-turut disajikan pada Tabel 3.

Confusion matrix memuat beberapa informasi yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi, seperti akurasi, spesifisitas, sensitivitas atau *recall*, dan lainnya. Akurasi menunjukkan jumlah data yang diprediksi dengan benar terhadap jumlah keseluruhan data. Sensitivitas menunjukkan keandalan model atau metode klasifikasi untuk mendeteksi data kelas positif dengan benar, sedangkan spesifisitas merupakan keandalan model atau metode klasifikasi untuk mendeteksi data kelas negatif dengan benar. Ketiga ukuran tersebut digunakan dalam penelitian sebagai ukuran performa metode klasifikasi dan dapat diperoleh dengan cara sebagai berikut.

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN} \quad \dots (8)$$

$$\text{Sensitivitas} = \frac{TP}{TP+FN} \quad \dots (9)$$

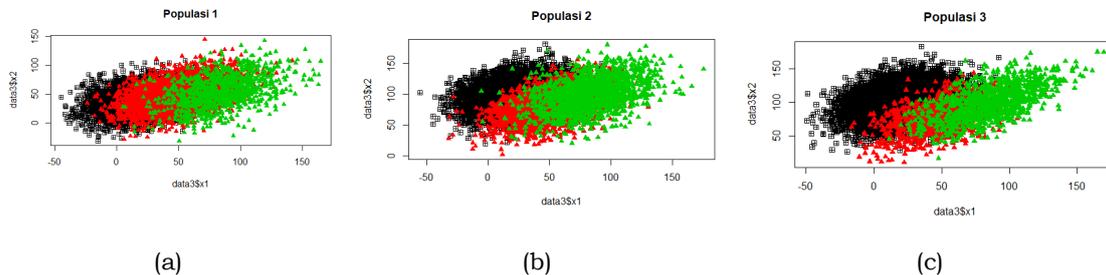
$$\text{Spesifisitas} = \frac{TN}{TN+FP} \quad \dots (10)$$

2.5. Data

Data yang digunakan pada penelitian ini merupakan data simulasi yang dibangkitkan menggunakan *software R*. Terdiri dari dua peubah prediktor yang bersifat kontinu serta peubah respon yang bersifat kategorik. Banyaknya kelas yang dibangun sebanyak 3 buah (1, 2, 3) dengan proporsi jumlah amatan yang dibuat berbeda. Data simulasi tersebut dibangkitkan menggunakan peubah acak multivariat normal dengan nilai μ dan Σ tertentu sehingga sebaran antara amatan kelas mayoritas dan minoritas dapat beririsan (disajikan pada Gambar 3). Jumlah data populasi diasumsikan sebanyak 10000 amatan sehingga komposisi jumlah amatan bagi setiap kelasnya adalah sebagai berikut.

Tabel 4. Proporsi Jumlah Amatan Kelas pada Data Simulasi

Data	Kelas		
	1	2	3
Data 1	45% (4500)	45% (4500)	10% (1000)
Data 2	60% (6000)	25% (2500)	15% (1500)
Data 3	80% (8000)	10% (1000)	10% (1000)



Gambar 3. Scatter Plot Data Simulasi Populasi: (a) Data 1, (b) Data 2, (c) Data 3

Sehubungan dengan kasus pada penelitian ini merupakan permasalahan multi kelas tidak seimbang (*imbalanced multi-class*), maka data simulasi juga dirancang berdasarkan banyaknya kelas minoritas pada suatu gugus data. Data simulasi pertama memiliki dua kelas mayoritas (kelas 1 dan 2) dan satu kelas minoritas (kelas 3). Data simulasi populasi pertama ditunjukkan oleh Gambar 3-a dimana amatan warna hitam dan merah merupakan kelas mayoritas. Data ke-2 menunjukkan terdapat satu kelas mayoritas (kelas 1) dan dua kelas minoritas dengan persentase yang berbeda (kelas 2 dan 3). Sebaran data pada populasi ke-2 tersebut dideskripsikan melalui Gambar 3-b dimana kelas mayoritas ditunjukkan oleh amatan warna hitam, sedangkan titik-titik warna merah dan hijau secara berturut-turut merupakan 25% dan 15% dari keseluruhan amatan. Data atau populasi ke-3 terdiri dari satu kelas mayoritas (kelas 1) dan dua kelas minoritas dengan persentase yang sama (kelas 2 dan 3). Sebaran data tersebut terlihat dari Gambar 3-c dimana amatan kelas minoritas digambarkan dengan warna merah dan hijau.

2.6. Langkah Kerja

Langkah kerja penelitian ini adalah sebagai berikut.

1. Membagi setiap gugus data simulasi menjadi data latih (80% atau 8000 amatan) dan uji (20% atau 2000 amatan). Pembagian tersebut dilakukan menggunakan teknik penarikan contoh acak sederhana.
2. Membangun model klasifikasi menggunakan prosedur dekomposisi *One-Versus-One* (OVO) dengan tahapan sebagai berikut.
 - a. Membangun data latih berdasarkan pasangan sub kelas yang mungkin dari 3 kelas yang ada, yaitu sebagai berikut.

Tabel 5. Pasangan Sub-kelas Berdasarkan Metode *One-Versus-One* (OVO)

Sub-kelas A	Sub-kelas B	Sub-kelas C
(1,2)	(1,3)	(2,3)

- b. Membangun model klasifikasi pada setiap sub-kelas menggunakan data latih.
 - 1) Menerapkan SVM standar pada data latih tanpa adanya tahap *preprocessing*.
 - 2) Menerapkan SVM-SMOTE dengan terlebih dahulu menerapkan teknik *oversampling* SMOTE. Setelah data latih menjadi seimbang, proses dilanjutkan dengan menerapkan SVM standar.
 - 3) Menerapkan GSVM-RU sesuai dengan tahapan yang dijelaskan pada Gambar 2.
- c. Tahapan b diulang sebanyak 50 kali menggunakan metode *bootstrap*.
- 3. Melakukan prediksi terhadap data uji berdasarkan model klasifikasi yang dihasilkan pada tahap 2 menggunakan prosedur dekomposisi *One-Versus-One* (OVO).
 - a. Melakukan prediksi sebanyak 50 kali untuk setiap data uji sehingga diperoleh *layout* tabel hasil klasifikasi bagi setiap sub-kelas sebagai berikut.

Tabel 6. *Layout* Hasil Prediksi *Bootstrap*

Data Uji ke-	SVM				SVM-SMOTE				GSVM-RU			
	<i>Bootstrap</i> ke-											
	1	2	...	50	1	2	...	50	1	2	...	50
1												
2												
3												
⋮												
2000												

- b. Menentukan hasil prediksi pada setiap sub-kelas bagi masing-masing data uji berdasarkan nilai modus dari hasil *bootstrap*. Lakukan proses ini untuk model SVM standar, SVM-SMOTE, dan GSVM-RU sehingga diperoleh *layout* hasil klasifikasi sebagai berikut.

Tabel 7. *Layout* Hasil Prediksi pada Setiap *Subset*

Kelas Biner (Sub-kelas)	Data Uji	SVM	SVM-SMOTE	GSVM-RU
A (1,2)	1
	2
	⋮
	2000
B (1,3)	1
	2
	⋮
	2000
C (2,3)	1
	2
	⋮
	2000

- c. Menentukan hasil prediksi akhir setiap data uji dengan cara mencari modus dari hasil prediksi ketiga sub-kelas. *Layout* hasil prediksi akhir ditampilkan sebagai berikut.

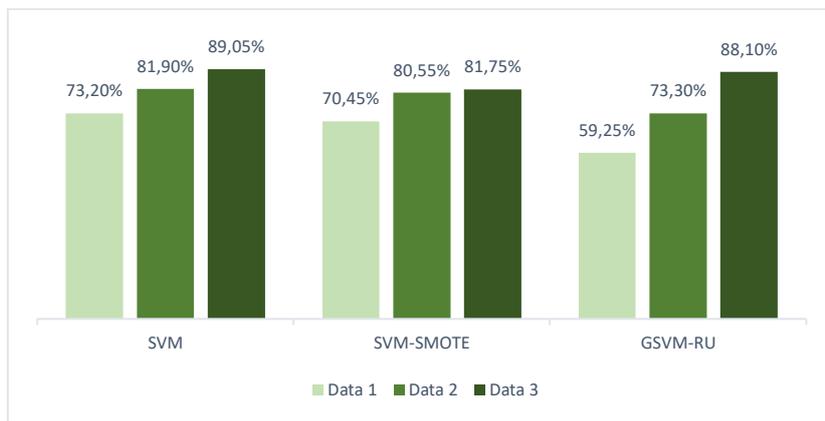
Tabel 8. *Layout* Hasil Prediksi Akhir Berdasarkan Metode Dekomposisi

Data Uji ke-	SVM			Hasil Prediksi Akhir (SVM)	SVM-SMOTE			Hasil Prediksi Akhir (SVM-SMOTE)	GSVM-RU			Hasil Prediksi Akhir (GSVM-RU)
	1 vs 2	1 vs 3	2 vs 3		1 vs 2	1 vs 3	2 vs 3		1 vs 2	1 vs 3	2 vs 3	
	1											
2												
3												
⋮												
2000												

4. Melakukan evaluasi performa SVM standar, SVM-SMOTE, dan GSVM-RU berdasarkan nilai akurasi, sensitivitas, dan spesifisitas.
5. Membandingkan performa model klasifikasi SVM standar, SVM-SMOTE, dan GSVM-RU dalam menangani multi kelas tidak seimbang.

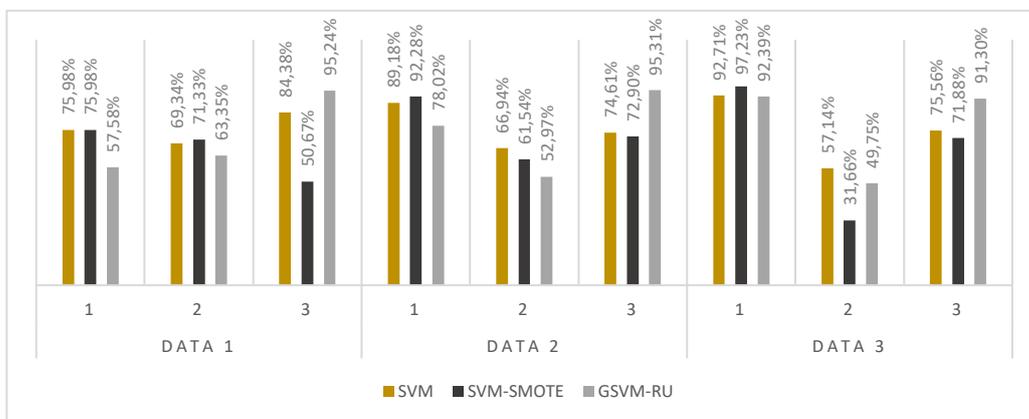
3. HASIL DAN PEMBAHASAN

Gambar 4 menunjukkan nilai akurasi seluruh metode klasifikasi pada setiap jenis data simulasi. Hasil yang diperoleh menunjukkan bahwa nilai akurasi meningkat seiring penambahan jumlah amatan kelas 1 pada setiap data simulasi. Hal tersebut terjadi pada ketiga metode klasifikasi, yaitu SVM standar, SVM-SMOTE, dan GSVM-RU. Di antara ketiga metode klasifikasi tersebut, metode SVM standar memiliki tingkat akurasi tertinggi, sedangkan GSVM-RU memiliki tingkat akurasi terendah.



Gambar 4. Performa Metode Klasifikasi Berdasarkan Akurasi

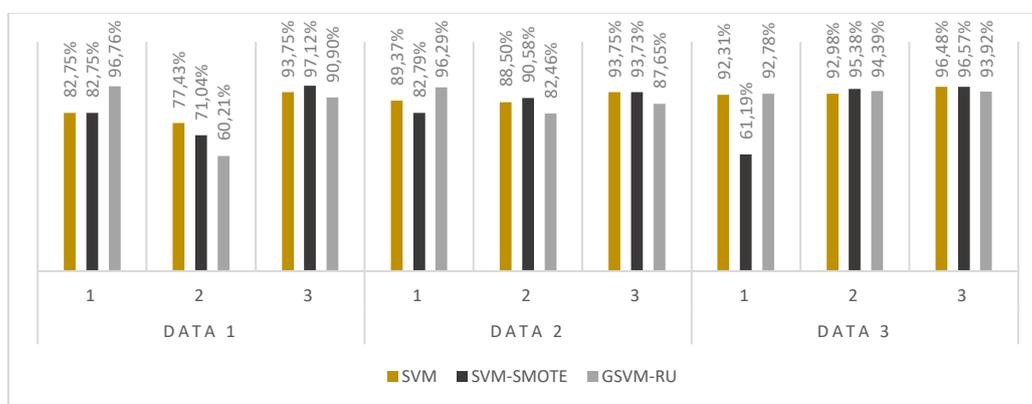
Perbandingan performa metode klasifikasi berdasarkan nilai sensitivitas dapat dilihat pada Gambar 5. Metode SVM dan SVM-SMOTE menunjukkan performa yang relatif konsisten untuk kelas 1. Hal tersebut dapat terlihat dari tingkat sensitivitas yang berdekatan di antara kedua metode tersebut, sedangkan tingkat sensitivitas tertinggi pada seluruh data simulasi terjadi di kelas 3 yaitu yang memiliki jumlah amatan terkecil dimana GSVM-RU diterapkan.



Gambar 5. Performa Metode Klasifikasi Berdasarkan Sensitivitas

Gambar 6 menunjukkan performa metode klasifikasi berdasarkan ukuran spesifisitas. Secara umum SVM standar, SVM-SMOTE, dan GSVM-RU menunjukkan tingkat spesifisitas yang relatif

konsisten tinggi. Pada kelas 1, GSVM-RU menunjukkan spesifisitas tertinggi dibandingkan dengan metode klasifikasi lainnya, yaitu berkisar 96,29-92,78 persen. Pada kelas 2, tingkat spesifisitas yang ditunjukkan oleh ketiga metode klasifikasi relatif sama untuk seluruh jenis data simulasi, kecuali pada data 1 dimana lebih rendah dibandingkan pada data simulasi lainnya. Pada kelas 3, metode SVM dan SVM-SMOTE menunjukkan spesifisitas yang berdekatan dan lebih tinggi dibandingkan dengan hasil yang diperoleh berdasarkan metode GSVM-RU, meskipun secara umum nilainya masih baik.



Gambar 6. Performa Metode Klasifikasi Berdasarkan Spesifisitas

Berdasarkan penjelasan sebelumnya, SVM dan SVM-SMOTE memiliki performa yang sama baiknya. Hal ini dapat dilihat dari tingkat sensitivitas kedua model tersebut yang secara umum hampir sama. Model SVM dan SVM-SMOTE memiliki tingkat sensitivitas yang baik pada kelas 1 dimana jika melihat data aktual simulasi, kelas tersebut memiliki jumlah amatan yang minimal sama atau lebih banyak dibandingkan kelas lainnya. Hal yang berbeda ditunjukkan oleh model GSVM-RU yang menunjukkan sensitivitas terbaik pada kelas 3 yang memiliki jumlah amatan paling sedikit dibandingkan kelas lainnya. Hal ini menunjukkan bahwa pada kasus multi kelas tidak seimbang (*imbalanced multi-class*), model GSVM-RU memiliki sensitivitas yang lebih baik dibandingkan SVM dan SVM-SMOTE dalam mendeteksi data berlabel kelas minoritas.

Spesifisitas merupakan keandalan model klasifikasi untuk mendeteksi data kelas negatif dengan benar. Dengan demikian, spesifisitas untuk kelas 1 menunjukkan keandalan model untuk mendeteksi kelas lainnya, yaitu kelas 2 dan 3 secara benar. Dengan ide yang sama, hal tersebut berlaku pula untuk kelas 2 dan 3. Berdasarkan tingkat spesifisitas yang diperoleh pada penelitian ini, model GSVM-RU memiliki performa yang baik pada kelas 1 di seluruh data simulasi. Data simulasi yang dibangkitkan menunjukkan bahwa kelas 1 merupakan jumlah amatan yang lebih banyak dibandingkan kelas lainnya secara umum sehingga kelas lainnya memiliki jumlah amatan yang lebih sedikit. Hal ini menunjukkan bahwa GSVM-RU memiliki performa yang konsisten dalam mendeteksi kelas minoritas. Berdasarkan Gambar 5, model klasifikasi SVM standar dan SVM-SMOTE memiliki tingkat spesifisitas yang hampir seluruhnya sama. Kedua metode klasifikasi tersebut memiliki performa yang baik pada kelas 3 di seluruh jenis data simulasi. Dengan demikian SVM standar dan SVM-SMOTE mampu mendeteksi selain kelas 3 dengan baik dimana jumlah amatannya relatif lebih banyak. Hal ini mendukung hasil yang diperoleh berdasarkan tingkat sensitivitas kedua model klasifikasi tersebut.

4. SIMPULAN DAN SARAN

Berdasarkan tujuan penelitian, maka berikut ini merupakan kesimpulan berdasarkan hasil analisis yang diperoleh.

- Seluruh model klasifikasi (SVM standar, SVM-SMOTE, dan GSVM-RU) menunjukkan tingkat akurasi tertinggi pada data simulasi 3 dimana perbandingan persentase antara jumlah amatan kelas mayoritas dan minoritasnya paling tinggi.
- Berdasarkan kriteria sensitivitas dan spesifisitas, model klasifikasi SVM standar dan SVM SMOTE memberikan performa yang sama baiknya pada kelas mayoritas, sedangkan model klasifikasi GSVM-RU memiliki performa yang baik dalam mendeteksi kelas minoritas.

Berdasarkan hasil penelitian, pembahasan, serta kesimpulan, maka beberapa hal berikut ini dapat dijadikan sebagai bahan perbaikan penelitian ke depannya.

- a. Data penelitian dapat ditingkatkan lagi dimensinya, misalnya dengan melibatkan lebih dari dua variabel dengan skala pengukuran yang beragam sehingga kompleksitas masalah yang muncul dapat memberikan ruang metode analisis tambahan untuk dilibatkan.
- b. Fungsi kernel pada metode *support vector machine* dapat dilakukan berdasarkan kajian yang tepat.
- c. Metode dekomposisi *one-versus-all* dapat dicoba untuk diterapkan dan dibandingkan kinerjanya dengan metode dekomposisi *one-versus-one*.

DAFTAR PUSTAKA

- Azis, A. I. S., Suhartono, V., & Himawan, H. (2017). Model Multi-class SVM Menggunakan Strategi 1V1 untuk Klasifikasi Wall-Following Robot Navigation Data. *Jurnal Teknologi Informasi*, 13(2).
- Boser, B. E., Guyon, I.M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of Fifth Annual Workshop on Computational Learning Theory*, 144-152. <https://doi.org/10.1145/130385.130401>.
- Chawla, N. V., Bowyer, K. W., Hall, L.O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357. <https://doi.org/10.1613/jair.953>.
- Esteves, V. M. S. (2020). *Techniques to deal with imbalanced data in multi-class problems: A review of existing methods*. Faculdade de Engenharia da Universidade do Porto.
- Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F. (2011). A review an ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems*. 42:463-484.
- Ghanem, A. S., Venkatesh, S., & West, G. (2010). Multi-class pattern classification in imbalanced data. *Proceedings: 20th International Conference on Pattern Recognition, IEEE, Los Alamitos, Calif.*, 2881-2884. <https://doi.org/10.1109/ICPR.2010.706>.
- Han, J. & Kamber, M. (2006). *Data Mining: Concepts and Techniques, 2nd ed.* San Fransisco: Morgan Kaufmann.
- He H, Garcia EA. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*. 21:1263-1284.
- Hu F, Li H. (2013). A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRS Boundary-SMOTE. *Mathematical Problems in Engineering*. 2013(694809): 1-10.
- Lango, M. (2019). Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study. *Sciendo*, 44(2). <https://doi.org/10.2478/fods-2019-0009>.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C. C., & Lin, C. C. (2020, February 1). Package 'e1071'. <https://CRAN.R-project.org/package=e1071>.
- Núñez, H., Gonzalez-Abril, L. & Angulo, C. (2017). Improving SVM Classification on Imbalanced Datasets by Introducing a New Bias. *Journal of Classification*, 34(3):427-443. <https://doi.org/10.1007/s00357-017-9242-x>
- Phung SL, Bouzerdoum A, Nguyen GH. (2009). Learning pattern classification tasks with imbalanced data sets. Di dalam: P. Yin, editor. *Pattern Recognition*, 193-208. Vukovar, Croatia: In-Teh.
- Y. Tang, Y. -Q. Zhang, N. V. Chawla and S. Krasser. (2009). SVMs Modeling for Highly Imbalanced Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 39(1):281-288. doi: 10.1109/TSMCB.2008.2002909.