

PENDEKATAN *RESAMPLING* DATA UNTUK MENANGANI MASALAH KETIDAKSEIMBANGAN KELAS

Yosua Alberth Sir¹ dan Agus H. H. Soepranoto²

¹Pranata Komputer, Universitas Nusa Cendana, Jl. Adisucipto, Penfui, Kupang, Indonesia

Email: yosuasir@staf.undana.ac.id

Email: hendrosoepranoto@gmail.com

ABSTRAK

Masalah ketidakseimbangan kelas dalam bidang pembelajaran mesin adalah masalah yang muncul karena adanya perbedaan jumlah *instance* yang signifikan antara kelas minoritas dengan kelas mayoritas. Perbedaan rasio *instance* ini membuat *classifier* mengambil keputusan yang keliru pada saat klasifikasi yaitu cenderung lebih memilih kelas mayoritas serta mengabaikan kelas minoritas. Untuk menangani masalah tersebut kami menggunakan pendekatan *resampling* data. Pendekatan *resampling* yang dimaksud adalah penggunaan 6 jenis teknik *resampling* data yang populer, yaitu: (i) *random oversampling* (ROS), (ii) *random undersampling* (RUS), (iii) *synthetic minority oversampling technique* (SMOTE), (iv) *adaptive synthetic sampling* (ADASYN), (v) SMOTETomek, dan (vi) SMOTEENN untuk membuat rasio jumlah *instance* dari 15 jenis *dataset* menjadi seimbang. Selanjutnya *dataset* yang sudah seimbang diklasifikasi menggunakan algoritma *random forest*. Metrik yang digunakan sebagai alat ukur kinerja adalah *geometric mean* (G-Mean). Untuk membandingkan kinerja dari 6 jenis teknik *resampling* data maka nilai-nilai G-Mean ini diuji menggunakan uji statistik nonparametrik *Friedman*, dan jika hipotesis nol ditolak maka dilanjutkan dengan melakukan uji statistik *Post Hoc Nemenyi*. Berdasarkan pada nilai *mean of ranks*, teknik *resampling* yang paling baik adalah SMOTEENN (1,700), ADASYN (2,767), RUS (3,333), SMOTETomek (3,867), SMOTE (4,000), ROS (5,333).

Kata kunci: *resampling* data, ketidakseimbangan kelas, uji *Friedman*, uji *Post Hoc Nemenyi*

ABSTRACT

Imbalanced class problem (machine learning) is a problem that arises because of the significant difference in the number of instances between the minority class and the majority class. Imbalanced class ratio makes the classifier do the wrong decision when classifying, which tends to prefer the majority class and ignore the minority class. To tackle this problem, we use a data resampling approach that use 6 types of popular data resampling techniques, such as: (i) random oversampling (ROS), (ii) random undersampling (RUS), (iii) synthetic minority oversampling technique (SMOTE), (iv) adaptive synthetic sampling (ADASYN), (v) SMOTETomek, and (vi) SMOTEENN to balance the ratio of the number of instances of 15 types of datasets. Furthermore, this balanced dataset is classified using a random forest classifier. The metric used as a performance measurement tool is the geometric mean (G-Mean). To compare the performance of the 6 types of data resampling techniques, these G-Mean values were tested using Friedman's nonparametric statistical test, and if the null hypothesis was rejected, it was continued with Nemenyi's Post Hoc statistical test. Based on mean of ranks values, the best resampling technique is SMOTEENN (1.700), ADASYN (2.767), RUS (3.333), SMOTETomek (3.867), SMOTE (4.000), ROS (5.333).

Keywords: resampling data, imbalanced class, Friedman test, Post Hoc Nemenyi test

1. PENDAHULUAN

Masalah ketidakseimbangan kelas dalam bidang pembelajaran mesin adalah adanya perbedaan jumlah *instance* yang signifikan antara kelas minoritas dengan kelas mayoritas. Efek negatif dari adanya ketidakseimbangan kelas adalah *classifier* akan mengambil keputusan yang keliru pada saat klasifikasi karena lebih memilih kelas mayoritas daripada kelas minoritas. Sebenarnya, semua *dataset* yang memiliki distribusi kelas yang tidak sama termasuk dalam kategori ketidakseimbangan kelas namun yang dianggap sebagai masalah ketidakseimbangan kelas hanya jika terdapat rasio ketidakseimbangan yang signifikan pada setiap kelasnya [1]. Menurut [2], rasio ketidakseimbangan minimal bagi *dataset* yang dianggap sebagai masalah adalah lebih besar atau sama dengan 1:4.

Ada dua pendekatan dalam menangani masalah ketidakseimbangan kelas, yaitu pendekatan pada level algoritmik dan pendekatan pada level data [3], [4]. Pendekatan level algoritmik lebih difokuskan pada perbaikan algoritma *classifier* tanpa mengubah distribusi kelas (contohnya algoritma *cost sensitive*)

sedangkan pada level data difokuskan pada penggunaan berbagai macam teknik *resampling* data untuk membuat distribusi kelas menjadi seimbang. Di antara dua pendekatan tersebut, pendekatan level data (*resampling* data) terbukti memiliki kinerja lebih baik daripada pendekatan level algoritmik. Teknik *resampling* data dikelompokkan dalam tiga jenis [5], yaitu: (i) jenis *oversampling* (*random oversampling* (ROS), *synthetic minority oversampling technique* (SMOTE), *adaptive synthetic sampling* (ADASYN), dan *borderline-SMOTE*), (ii) jenis *undersampling* (*random undersampling* (RUS), *tomek links*, *edited nearest neighbors rule* (ENN)), (iii) kombinasi dari keduanya (SMOTETomek dan SMOTEENN). Untuk mengetahui teknik *resampling* mana yang terbaik dalam menangani masalah ketidakseimbangan kelas maka perlu diuji coba pada sejumlah *dataset* yang memiliki rasio ketidakseimbangan kelas yang bervariasi.

Pada penelitian ini, kami membandingkan kinerja dari 6 teknik *resampling* data saat menyeimbangkan 15 *dataset* yang memiliki rasio ketidakseimbangan yang bervariasi. Untuk mengukur kinerja dari setiap teknik *resampling* data maka *geometric mean* (G-Mean) dipilih sebagai metrik pengukurannya. Metrik G-Mean dipilih karena merupakan salah satu metrik terbaik khususnya saat menangani masalah ketidakseimbangan kelas [6]. Selanjutnya, lima belas *dataset* yang sudah diseimbangkan diklasifikasi menggunakan algoritma *random forest*. Untuk mengetahui teknik *resampling* mana yang terbaik maka nilai G-Mean hasil klasifikasi dibandingkan satu sama lainnya menggunakan uji statistik *Friedman* [7] dan jika hipotesis nol ditolak akan dilanjutkan dengan menggunakan uji statistik *Post Hoc Nemenyi* [8].

2. MATERI DAN METODE

Bagian ini dimulai dengan penjelasan tentang metode, kemudian dilanjutkan dengan *dataset*, teknik *resampling* data yang digunakan, algoritma klasifikasi *random forest* dan terakhir membahas mengenai uji statistik yang digunakan untuk membandingkan kinerja model.

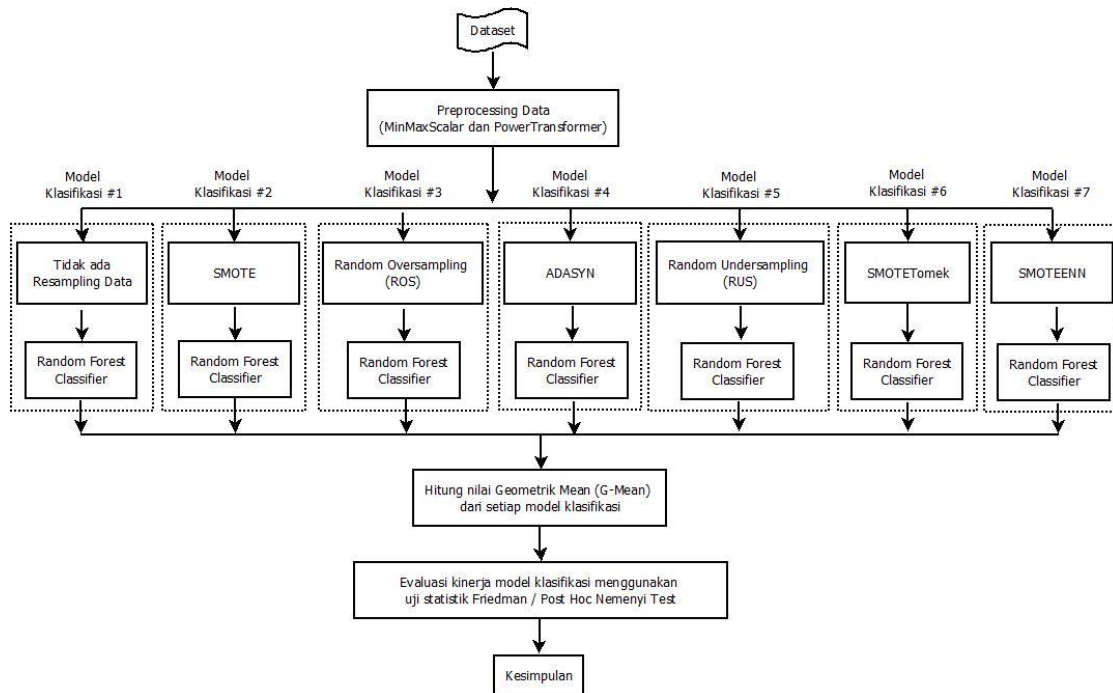
Metode

Terdapat dua *tool* yang digunakan sebagai alat bantu, yaitu: (i) pustaka *imbalanced-learn* versi 0.9.0 yang digunakan untuk *resampling* data dan klasifikasi. Pustaka ini ditulis dengan bahasa pemrograman *python* serta dapat diunduh pada laman <https://github.com/scikit-learn-contrib/imbalanced-learn>, (ii) *Addin XLSTAT* yang digunakan untuk melakukan uji statistik nonparametrik *Friedman* dan *Post Hoc Nemenyi*. *Tool* ini dapat diunduh pada laman: <https://www.xlstat.com/en/download>. Gambaran umum mengenai tahapan penelitian tampak pada gambar 1, berikut ini adalah detail tahapan penelitian:

1. Membangun 7 model klasifikasi. Pada saat eksekusi, ke-7 model tersebut diklasifikasi menggunakan algoritma *random forest* dan pada bagian *cross validation* digunakan metode *RepeatedStratifiedKfold*.
 - a. Model #1: 15 *dataset* diklasifikasi menggunakan *random forest*. Pada model ini tidak ada *resampling* data, semua *dataset* dibiarkan tidak seimbang.
 - b. Model #2: 15 *dataset* di-*resampling* menggunakan SMOTE. Setelah distribusi data menjadi seimbang kemudian diklasifikasi menggunakan *random forest*.
 - c. Model #3: 15 *dataset* di-*resampling* menggunakan ROS. Setelah distribusi data menjadi seimbang kemudian diklasifikasi menggunakan *random forest*.
 - d. Model #4: 15 *dataset* di-*resampling* menggunakan ADASYN. Setelah distribusi data menjadi seimbang kemudian diklasifikasi menggunakan *random forest*.
 - e. Model #5: 15 *dataset* di-*resampling* menggunakan RUS. Setelah distribusi data menjadi seimbang kemudian diklasifikasi menggunakan *random forest*.
 - f. Model #6: 15 *dataset* di-*resampling* menggunakan SMOTETomek. Setelah distribusi data menjadi seimbang kemudian diklasifikasi menggunakan *random forest*.
 - g. Model #7: 15 *dataset* di-*resampling* menggunakan SMOTEENN. Setelah distribusi data menjadi seimbang kemudian diklasifikasi menggunakan *random forest*.
2. Melakukan evaluasi kinerja setiap model klasifikasi dengan cara menggunakan nilai G-Mean dari masing-masing model. Selanjutnya nilai G-Mean ini diuji menggunakan uji statistik *Friedman* dan dilanjutkan dengan uji *Post Hoc Nemenyi* untuk mengetahui model klasifikasi mana yang terbaik.

Dataset

Dataset yang digunakan berjumlah 15 *dataset*, di mana semuanya bersifat *binary class* dan memiliki distribusi kelas yang tidak seimbang dengan *imbalanced ratio* (IR) sebesar 7. Semua *dataset* diambil pada laman <https://sci2s.ugr.es/keel/imbalanced.php?order=ir#sub50>. Detail *dataset* penelitian tampak pada tabel 1.



Gambar 1. Tahapan penelitian

Tabel 1. *Dataset* penelitian

No	<i>Dataset</i>	IR	Jumlah Instance
1	paw02a-800-7-70-BI	7	Kelas 0: 100, Kelas 1: 700
2	paw02a-800-7-60-BI	7	Kelas 0: 100, Kelas 1: 700
3	paw02a-800-7-50-BI	7	Kelas 0: 100, Kelas 1: 700
4	paw02a-800-7-30-BI	7	Kelas 0: 100, Kelas 1: 700
5	paw02a-800-7-0-BI	7	Kelas 0: 100, Kelas 1: 700
6	04clover5z-800-7-70-BI	7	Kelas 0: 100, Kelas 1: 700
7	04clover5z-800-7-60-BI	7	Kelas 0: 100, Kelas 1: 700
8	04clover5z-800-7-50-BI	7	Kelas 0: 100, Kelas 1: 700
9	04clover5z-800-7-30-BI	7	Kelas 0: 100, Kelas 1: 700
10	04clover5z-800-7-0-BI	7	Kelas 0: 100, Kelas 1: 700
11	03subcl5-800-7-70-BI	7	Kelas 0: 100, Kelas 1: 700
12	03subcl5-800-7-60-BI	7	Kelas 0: 100, Kelas 1: 700
13	03subcl5-800-7-50-BI	7	Kelas 0: 100, Kelas 1: 700
14	03subcl5-800-7-30-BI	7	Kelas 0: 100, Kelas 1: 700
15	03subcl5-800-7-0-BI	7	Kelas 0: 100, Kelas 1: 700

Resampling data

Kami memilih 3 teknik *oversampling* (ROS, ADASYN, SMOTE), 1 teknik *undersampling* (RUS) dan 2 teknik *over-under sampling* (SMOTEENN, SMOTETomek) untuk penelitian ini. Metode-metode tersebut dipilih karena paling sering digunakan di kelasnya [5]. Tabel 2 menjelaskan tentang teknik *resampling* data yang digunakan dalam penelitian ini.

Algoritma Klasifikasi: Random Forest

Random forest adalah *classifier* yang termasuk dalam golongan *ensemble*. Cara kerja algoritma *random forest* sebagai berikut: anggap setiap *classifier* dalam *ensemble* berbentuk sebuah *decision tree* (pohon keputusan) maka dapat dikatakan bahwa kumpulan *classifier* tersebut adalah *forest* (hutan). Dalam *forest* tersebut dibuat banyak *decision tree*, kemudian anggap sebuah *test data* dimasukkan kedalam *forest*. Setiap *decision tree* dalam *forest* akan memberi keputusan dan keputusan akhir (keputusan dari *forest*) adalah merupakan keputusan mayoritas dari setiap *decision tree* dalam *forest* [9]. Untuk memberikan pemahaman mengenai cara kerja *random forest* maka diberikan analogi berikut: anggap perwakilan pemerintah bertanya kepada 300 orang anggota Dewan Perwakilan Rakyat (DPR) mengenai apakah ibukota

negara harus pindah ke Kalimantan Timur atau tetap di Jakarta. Setiap anggota DPR kemudian memberikan keputusannya dan keputusan-keputusan tersebut bisa saja sama atau berbeda. Keputusan akhir dari DPR adalah merupakan keputusan mayoritas dari anggota DPR. Pada kasus ini, DPR merepresentasikan *forest*, anggota DPR merepresentasikan *decision tree* dan pertanyaan pemerintah merepresentasikan *test data*.

Tabel 2. Teknik *resampling* data

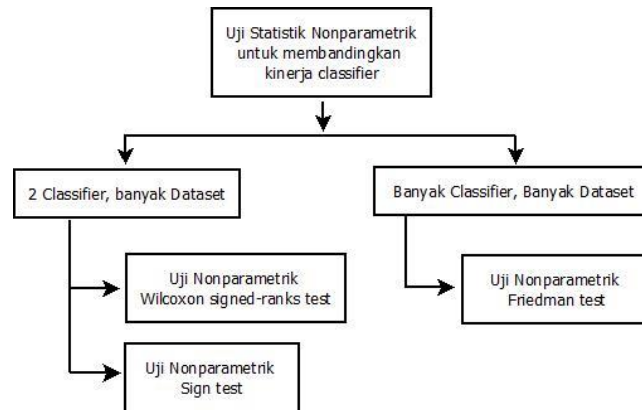
Metode	Jenis <i>Resampling</i>	Cara Kerja
ROS	<i>Oversampling</i>	ROS bekerja dengan cara memilih <i>instance</i> dalam kelas minoritas secara acak kemudian melakukan duplikasi berulang kali sampai jumlah <i>instance</i> pada kelas minoritas dan mayoritas menjadi seimbang. Pendekatan ini memiliki kelemahan karena menghasilkan banyak duplikasi <i>instance</i> yang akhirnya dapat menimbulkan masalah <i>overfitting</i> .
SMOTE	<i>Oversampling</i>	Pendekatan lain yang cenderung lebih cerdas daripada ROS adalah SMOTE yang bekerja dengan cara mengambil secara acak tetangga terdekat sebanyak <i>k</i> dari setiap <i>instance</i> dalam kelas minoritas kemudian membuat <i>instance</i> baru (sintetis) antara <i>instance</i> tersebut dengan tetangga terdekat <i>k</i> yang dipilih secara acak. Dengan pendekatan SMOTE maka dapat dipastikan tidak terjadi masalah duplikasi data sehingga lebih kebal terhadap masalah <i>overfitting</i> .
ADASYN	<i>Oversampling</i>	ADASYN bekerja dengan cara menggunakan bobot distribusi untuk <i>instance</i> pada kelas minoritas berdasarkan pada tingkat kesulitan pembelajaran data oleh model, di mana <i>instance</i> baru (sintetis) dihasilkan dari kelas minoritas yang susah untuk belajar dibandingkan dengan data minoritas yang lebih mudah untuk belajar.
RUS	<i>Undersampling</i>	RUS bekerja dengan cara memilih secara acak <i>instance-instance</i> pada kelas mayoritas kemudian menghapusnya. Proses ini dilakukan berulang kali sampai jumlah <i>instance</i> dalam kelas minoritas sama dengan kelas mayoritas. Pendekatan ini memiliki kelemahan karena membuat banyak <i>instance</i> penting terhapus sehingga dapat mempengaruhi kinerja <i>classifier</i> .
SMOTETomek	<i>Over-Under Sampling</i>	SMOTETomek adalah kombinasi dari SMOTE dan Tomek Link yang termasuk dalam kategori <i>over-under sampling</i> dimana teknik <i>oversampling</i> menggunakan SMOTE dan teknik <i>undersampling</i> adalah Tomek Link.
SMOTEENN	<i>Over-Under Sampling</i>	SMOTEENN adalah kombinasi antara SMOTE dan ENN dimana yang berperan sebagai <i>oversampling</i> adalah SMOTE sedangkan <i>undersampling</i> adalah ENN.

Uji Statistik Nonparametrik

Uji statistik terdiri dari dua bagian utama: (i) uji statistik parametrik, (ii) uji statistik nonparametrik. Uji statistik parametrik mewajibkan sebaran parameter terdistribusi normal atau homogen sedangkan uji statistik nonparametrik tidak mensyaratkan sebaran parameter populasi berbentuk distribusi normal. Di antara dua jenis uji statistik tersebut, uji statistik nonparametrik (gambar 2) lebih cocok digunakan untuk membandingkan kinerja algoritma-algoritma klasifikasi daripada menggunakan uji statistik parametrik (*paired t-test* dan ANOVA) [7].

Gambar 2 menunjukkan jenis-jenis uji statistik nonparametrik yang dapat digunakan untuk membandingkan kinerja *classifier*. Pada bagian 2 *classifier*, banyak *dataset*, uji statistik yang dapat digunakan adalah *Wilcoxon signed-ranks test* [10] dan *Sign test*. Sedangkan pada bagian banyak *classifier*, banyak *dataset*, uji statistik yang digunakan adalah uji nonparametrik *Friedman*.

Pada penelitian ini kami menggunakan uji nonparametrik *Friedman* untuk mendeteksi apakah ada perbedaan yang signifikan di antara teknik *resampling* data yang digunakan. Jika ada perbedaan maka dilanjutkan dengan uji statistik *Post Hoc Nemenyi* [8], [11] untuk membandingkan kinerja dari teknik-teknik *resampling* yang digunakan.



Gambar 2. Uji statistik nonparametrik untuk membandingkan kinerja classifier

3. HASIL DAN PEMBAHASAN

Pada tahap ini model klasifikasi #1 sampai model klasifikasi #7 seperti yang ditampilkan pada gambar 1 diklasifikasi. Parameter yang digunakan saat klasifikasi dapat dilihat pada tabel 3 sedangkan tabel 4 menunjukkan hasil klasifikasi berbentuk nilai nilai G-Mean dari 7 model klasifikasi. Angka-angka yang dicetak tebal pada tabel 4 menunjukkan nilai G-Mean tertinggi.

Tabel 3. Parameter yang digunakan saat klasifikasi

Metode	Parameter yang digunakan
ROS	Tidak ada
SMOTE	K_neighbours = 5
ADASYN	Tidak ada
RUS	Tidak ada
SMOTETomek	K (smote) = 5
SMOTEENN	K (smote) = 5, K(ENN) = 3
Random Forest Classifier	criterion = entropy, max_depth = 10, n_estimator = 100

Tabel 4. Nilai G-Means hasil klasifikasi

Dataset	Tanpa Sampling (Mdl#1)	ROS (Mdl#2)	SMOTE (Mdl#3)	ADASYN (Mdl#4)	RUS (Mdl#5)	SMOTE Tomek (Mdl#6)	SMOTE ENN (Mdl#7)
paw02a-800-7-70-BI	0,335	0,804	0,822	0,826	0,828	0,827	0,839
paw02a-800-7-60-BI	0,483	0,797	0,807	0,823	0,815	0,809	0,827
paw02a-800-7-50-BI	0,636	0,822	0,836	0,842	0,844	0,840	0,841
paw02a-800-7-30-BI	0,781	0,837	0,847	0,866	0,854	0,844	0,855
paw02a-800-7-0-BI	0,937	0,950	0,953	0,952	0,947	0,951	0,957
04clover5z-800-7-70-BI	0,256	0,753	0,786	0,805	0,777	0,796	0,807
04clover5z-800-7-60-BI	0,329	0,781	0,805	0,813	0,804	0,811	0,814
04clover5z-800-7-50-BI	0,376	0,791	0,806	0,809	0,793	0,808	0,810
04clover5z-800-7-30-BI	0,563	0,822	0,830	0,832	0,839	0,824	0,842
04clover5z-800-7-0-BI	0,737	0,906	0,915	0,906	0,889	0,916	0,917
03subcl5-800-7-70-BI	0,201	0,759	0,768	0,781	0,795	0,776	0,797
03subcl5-800-7-60-BI	0,321	0,762	0,781	0,800	0,801	0,799	0,800
03subcl5-800-7-50-BI	0,432	0,784	0,796	0,799	0,794	0,795	0,797
03subcl5-800-7-30-BI	0,668	0,824	0,824	0,823	0,825	0,820	0,820
03subcl5-800-7-0-BI	0,882	0,943	0,938	0,936	0,940	0,938	0,944

Ada dua skenario uji coba, yaitu: (i) uji coba 1 digunakan untuk mengetahui apakah model tanpa *resampling* memiliki kinerja yang sama dengan model menggunakan *resampling*, (ii) uji coba 2 difokuskan untuk membandingkan kinerja antar teknik-teknik *resampling* yang digunakan.

Uji coba 1: Model tanpa *resampling* Vs Model menggunakan *resampling*

Uji coba pertama dilakukan dengan tujuan untuk mengetahui apakah model yang tidak menggunakan *resampling* memiliki kinerja yang sama dengan model yang menggunakan *resampling*. Untuk keperluan tersebut digunakan uji statistik *Wilcoxon Signed-Rank* dan hasil uji ada di tabel 5.

Tabel 5. Hasil uji *Wilcoxon Signed-Rank*

<i>Variable Test</i>	<i>R</i> ⁺	<i>R</i> ⁻	<i>p-value</i>
Tanpa Resampling Vs ROS	0	120	0,00064
Tanpa Resampling vs SMOTE	0	120	0,00064
Tanpa Resampling vs ADASYN	0	120	0,00064
Tanpa Resampling vs RUS	0	120	0,00064
Tanpa Resampling vs SMOTETomek	0	120	0,00064
Tanpa Resampling vs SMOTEENN	0	120	0,00064

Hasil uji *Wilcoxon Signed-Rank* menunjukkan bahwa model “*Tanpa Resampling*” memiliki kinerja paling buruk dibandingkan dengan semua model-model lain yang menggunakan *resampling*. Hal tersebut dibuktikan dengan nilai *R*⁺ = 0 dan nilai *R*⁻ = 120. Nilai *R*⁺ adalah peringkat positif sehingga dapat dikatakan selama dilakukan proses perbandingan, model “*Tanpa Resampling*” tidak pernah menang terhadap model lainnya atau jumlah kemenangannya nol. Nilai *R*⁻ adalah peringkat negatif yang artinya selama dilakukan proses perbandingan, model “*Tanpa Resampling*” selalu kalah dari model lainnya dengan jumlah kekalahan sebesar 120. Selain nilai *R*⁺ dan *R*⁻, ada juga nilai *p-value* yang lebih rendah dari 0,01. Ini berarti model “*Tanpa Resampling*” adalah model paling buruk jika dibandingkan dengan model menggunakan *resampling* dengan level signifikan sebesar $\alpha = 0,01$ (sangat signifikan).

Uji coba 2: Perbandingan kinerja model *resampling*

Uji coba kedua bertujuan untuk membandingkan kinerja dari 6 teknik *resampling* data saat menangani masalah ketidakseimbangan kelas. Pada uji coba 2, kami tidak menggunakan model “*Tanpa Resampling*” karena tidak relevan dengan tujuan uji coba. Untuk membandingkan kinerja model *resampling* digunakan uji statistik *Friedman*. Hipotesis yang digunakan pada uji *Friedman* adalah:

Hipotesis **H₀** = Kinerja model#2 = model#3 = model#4 = model#5 = model#6 = model#7

Hipotesis **H₁** = Minimal ada satu model yang memiliki kinerja berbeda dengan model lainnya

Langkah uji *Friedman* sebagai berikut: mula-mula data pada tabel 4 dikonversi ke dalam bentuk peringkat dengan cara mengurutkan nilai G-Mean pada setiap baris kemudian nilai G-Mean terbesar diberi peringkat 1, terbesar kedua diberi peringkat 2, dan seterusnya. Apabila pada saat mengurutkan, ditemukan ada urutan yang sama maka peringkatnya ditentukan oleh nilai rata-ratanya. Contoh, pada tabel 4, baris ke-15, kolom 4 (bernilai 0,938) dan kolom 7 (bernilai 0,938). Kedua data tersebut memiliki nilai yang sama dan posisinya ada di urutan ke-4 dan ke-5 (urutan ke-3=0,940) maka nilai peringkatnya adalah $(4+5)/2=4,5$. Proses pemeringkatan tersebut dilakukan sampai baris terakhir. Hasil lengkap dari pemeringkatan data dapat dilihat pada tabel 6.

Tabel 6. Peringkat G-Means

<i>Dataset</i>	ROS (Mdl#2)	SMOTE (Mdl#3)	ADASYN (Mdl#4)	RUS (Mdl#5)	SMOTE Tomek (Mdl#6)	SMOTE ENN (Mdl#7)
paw02a-800-7-70-BI	6	5	4	2	3	1
paw02a-800-7-60-BI	6	5	2	3	4	1
paw02a-800-7-50-BI	6	5	2	1	4	3
paw02a-800-7-30-BI	6	4	1	3	5	2
paw02a-800-7-0-BI	5	2	3	6	4	1
04clover5z-800-7-70-BI	6	4	2	5	3	1
04clover5z-800-7-60-BI	6	4	2	5	3	1
04clover5z-800-7-50-BI	6	4	2	5	3	1
04clover5z-800-7-30-BI	6	4	3	2	5	1
04clover5z-800-7-0-BI	4,5	3	4,5	6	2	1
03subcl5-800-7-70-BI	6	5	3	2	4	1
03subcl5-800-7-60-BI	6	5	2,4	1	4	2,5
03subcl5-800-7-50-BI	6	3	1	5	4	2
03subcl5-800-7-30-BI	2,5	2,5	4	1	5,5	5,5
03subcl5-800-7-0-BI	2	4,5	6	3	4,5	1

Langkah berikutnya adalah melakukan uji *Friedman* terhadap data pada tabel 6. Hasil uji *Friedman* dapat dilihat pada tabel 7 di mana interpretasinya adalah: oleh karena nilai *p-value* lebih rendah daripada nilai $\alpha=0,05$, maka hipotesis **H₀** ditolak. Langkah selanjutnya adalah melakukan uji *Post Hoc Nemenyi*. Hasil uji *Post Hoc Nemenyi* dapat dilihat pada tabel 8 dan tabel 9. Tabel 8 menunjukkan nilai-nilai *p-value*

yang lebih kecil dari $\alpha = 0,05$ dicetak lebih tebal tetapi jika lebih besar atau sama dengan nilai α tidak dicetak tebal. Nilai p -value yang dicetak lebih tebal menunjukkan bahwa ada perbedaan yang signifikan di antara model pada baris dan kolom yang bersesuaian.

Tabel 7. Hasil uji statistik *Friedman*

Q (Observed value)	32,610
Q (Critical value)	11,070
DF	5
p -value (one-tailed)	<0,0001
α	0,050

Tabel 8. Perbandingan *pairwise* (p -values)

	ROS	SMOTE	ADASYN	RUS	SMOTE Tomek	SMOTE ENN
ROS	1	0,373	0,003	0,043	0,267	< 0,0001
SMOTE	0,373	1	0,464	0,925	1,000	0,011
ADASYN	0,003	0,464	1	0,962	0,593	0,625
RUS	0,043	0,925	0,962	1	0,971	0,164
SMOTE Tomek	0,267	1,000	0,593	0,971	1	0,021
SMOTE ENN	< 0,0001	0,011	0,625	0,164	0,021	1

Tabel 9. Perbandingan *multiple pairwise* menggunakan *Post Hoc Nemenyi / Two-tailed*

Sample	Frequency	Sum of ranks	Mean of ranks
SMOTE ENN	15	25,500	1,700
ADASYN	15	41,500	2,767
RUS	15	50,000	3,333
SMOTE Tomek	15	58,000	3,867
SMOTE	15	60,000	4,000
ROS	15	80,000	5,333

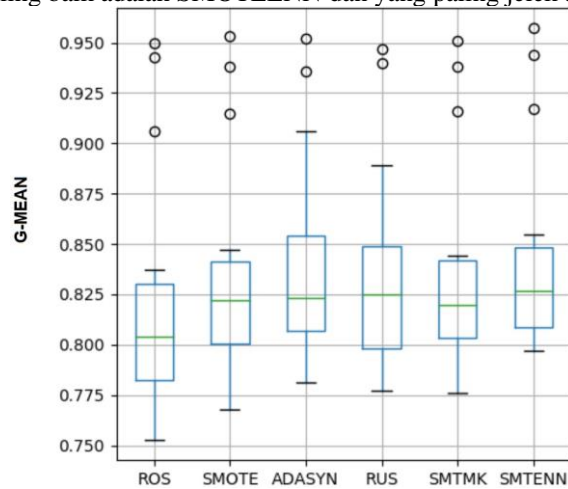
Selain menggunakan uji statistik, kami juga menggunakan diagram *boxplots* (gambar 3) untuk menganalisis hasil uji coba. Analisis *boxplot* sebagai berikut:

1. Nilai median

Teknik *resampling* SMOTEENN, ADASYN, dan RUS cenderung memiliki nilai median yang sama dibandingkan dengan teknik *resampling* lainnya. Hal ini ditunjukkan dari posisi garis tengah dalam *boxplot* yang cenderung sama. Nilai median yang tinggi berarti ketiga teknik *resampling* tersebut cenderung memiliki kinerja yang lebih baik dari teknik *resampling* lainnya. Teknik *resampling* ROS memiliki nilai median yang paling rendah dari semua *resampling* yang ada sehingga dapat dikatakan teknik *resampling* ROS adalah teknik *resampling* terburuk dari lainnya.

2. Ketahanan yang baik

Ketahanan yang baik artinya memiliki distribusi data yang simetris, garis median ada di tengah, dan *whisker* atas dan *whisker* bawah memiliki panjang yang sama. Teknik *resampling* yang memiliki ketahanan yang paling baik adalah SMOTEENN dan yang paling jelek adalah ADASYN.



Gambar 3. *Boxplot* dari nilai G-Means

4. KESIMPULAN DAN SARAN

Kami telah melakukan dua kali uji coba untuk menangani masalah ketidakseimbangan kelas. Percobaan pertama untuk mengetahui apakah pendekatan *resampling* lebih baik daripada tanpa *resampling*. Percobaan kedua untuk membandingkan kinerja dari teknik-teknik *resampling* yang ada. Berdasarkan hasil uji coba (tabel 9) dan hasil analisis *boxplot* diperoleh hasil sebagai berikut:

1. Penggunaan teknik *resampling* data yaitu ROS, SMOTE, ADASYN, RUS, SMOTETomek, dan SMOTEENN dapat mengatasi masalah ketidakseimbangan kelas. Pernyataan tersebut dibuktikan dari hasil uji *Wilcoxon Signed-Rank* pada tabel 5. Hasil perbandingan antara model “*Tanpa Resampling*” terhadap model ROS, SMOTE, ADASYN, RUS, SMOTETomek, dan SMOTEENN selalu menghasilkan kinerja yang buruk yang ditandai dengan nilai $R+ = 0$ dan nilai $R- = 136$.
2. Nilai *mean of ranks* pada tabel 9 menunjukkan bahwa peringkat terbaik ditunjukkan oleh nilai *mean of ranks* yang paling kecil sehingga jika diurutkan teknik *resampling* mulai dari yang terbaik sampai terburuk adalah SMOTEENN (1,700), ADASYN (2,767), RUS (3,333), SMOTETomek (3,867), SMOTE (4,000), dan ROS (5,333).

DAFTAR PUSTAKA

- [1] A. F. Hilario, S. G. López, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, “Learning from Imbalanced Data Sets,” *Springer*, p. 19, 2018, doi: [10.1007/978-3-319-98074-4](https://doi.org/10.1007/978-3-319-98074-4).
- [2] B. Krawczyk, “Learning from Imbalanced Data: Open Challenges and Future Directions,” *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016, doi: [10.1007/s13748-016-0094-0](https://doi.org/10.1007/s13748-016-0094-0).
- [3] V. Ganganwar, “An overview of classification algorithms for imbalanced datasets,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 42–47, 2012.
- [4] M. Fatourehchi, R. K. Ward, S. G. Mason, J. Huggins, A. Schloegl, and G. E. Birch, “Comparison of evaluation metrics in classification applications with imbalanced datasets,” in *2008 seventh international conference on machine learning and applications*, 2008, pp. 777–782. doi: [10.1109/ICMLA.2008.34](https://doi.org/10.1109/ICMLA.2008.34).
- [5] J. Brownlee, *Imbalanced classification with python: Better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery, 2020.
- [6] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics,” *Information sciences*, vol. 250, pp. 113–141, 2013, doi: [10.1016/j.ins.2013.07.007](https://doi.org/10.1016/j.ins.2013.07.007).
- [7] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [8] A. Saifudin and R. S. Wahono, “Penerapan teknik ensemble untuk menangani ketidakseimbangan kelas pada prediksi cacat software,” *IlmuKomputer. com Journal of Software Engineering*, vol. 1, no. 1, pp. 28–37, 2015.
- [9] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*, 3rd ed. Amsterdam: Morgan Kaufmann, 2011.
- [10] F. Grina, Z. Elouedi, and E. Lefevre, “A preprocessing approach for class-imbalanced data using SMOTE and belief function theory,” *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 3–11, 2020, doi: [10.1007/978-3-030-62365-4_1](https://doi.org/10.1007/978-3-030-62365-4_1).
- [11] A. Saifudin and R. S. Wahono, “Pendekatan Level Data untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software,” *IlmuKomputer. com Journal of Software Engineering*, vol. 1, no. 2, pp. 76–85, 2015.