MWMOTE DALAM MENGATASI KETIDAKSEIMBANGAN KELAS PADA PREDIKSI CHURN MENGGUNAKAN KLASIFIKASI C4.5

Muhamad Syarif, Wahyu Nugraha
Progam Studi Sistem Informasi, Fakultas Teknik dan Informatika
Universitas Bina Sarana Informatika, Jalan Abdul Rahman Saleh No.18, Pontianak
muhamad.mdx@bsi.ac.id, wahyu.whn@bsi.ac.id

ABSTRAK

Churn prediction merupakan metode untuk memprediksi potensi pelanggan yang akan melakukan *churn*. Penerapan klasifikasi pada *churn prediction* dapat menjadikan hasil yang lebih akurat. Namun karakteristik *dataset churn* yaitu memiliki tingkat *imbalance* yang besar, dikarenakan pelanggan yang *churn* lebih sedikit dibandingkan pelanggan yang tidak melakukan *churn*. Perlu melakukan pendekatan pada level data untuk mengatasi *imbalance* yaitu dengan teknik *resampling*. Penerapan *resampling* pada data yang memiliki *imbalance*, untuk mengurangi *imbalance* data maka perlu melakukan *resampling* dan klasifikasi data, salah satunya dengan menggunakan metode *oversampling* ataupun *undersampling*. Penggunaan resampling dapat menyeimbangkan data antara minoritas dan mayoritas. *Majority Weighted Minority Oversampling Technique* atau biasa disingkat MWMOTE merupakan salah satu metode *oversampling*. MWMOTE terbukti lebih baik dan sebanding dari hasil pengujian menggunakan 24 *dataset*. Hasil penelitian ini yaitu menghasilkan perbandingan nilai antara penggunaan MWMOTE dan tanpa MWMOTE, hasil perbandingan dapat dilihat pada hasil grafik perbandingan nilai *sensitivity*, *specificity* dan AUC. Hasilnya menunjukan peningkatan yang signifikan pada nilai *sensitifity* dan nilai AUC. Sehingga metode MWMOTE membuktikan menghasilkan nilai yang lebih baik pada penggujian yang menerapkan algoritma klasifikasi C4.5 *dataset churn prediction*.

Kata kunci: *Oversampling, MWMOTE, Churn Prediction, Imbalance, Algoritma C4.5.*

1. PENDAHULUAN

Perusahaan jasa khususnya bergerak dibidang telekomunikasi ketika kehilangan pelanggan merupakan hal merugikan dan menguntungkan bagi pesaing, hal ini dikenal dengan istilah *churn* pelanggan. *Churn* pelanggan merupakan kejadian kehilangan pelanggan secara berkala pada sebuah organisasi [1]. Banyaknya saingan dan gencarnya strategi untuk mendapatkan pelanggan baru menjadikan pelanggan mudah untuk beralih dari langganan lama ke perusahaan baru, hal tersebut karena liberalisasi pasar yang membuka persaingan, penawaran layanan baru dan pemanfaatan teknologi baru. *Churn* pelanggan menyebabkan kerugian dan menjadi masalah yang serius [2].

Mengurangi *churn* pelanggan adalah salah satu tujuan jangka panjang dari perusahaan jasa telekomunikasi manapun. Apalagi sekarang persaingan pasar semakin intens, biaya untuk menyerap pelanggan baru meningkat karena memerlukan biaya lebih di berbagai aspek untuk mendapatkan pelanggan baru seperti biaya promosi atau penggunaan teknologi baru agar dapat memberikan penawaran yang lebih baik dibandingkan kompetitor, tentunya biaya yang dikeluarkan dapat meningkat hingga 6 kali lebih tinggi jika dibandingkan dengan mempertahankan pelanggan lama. Tingkat keberhasilan pemasaran produk ke pelanggan baru adalah 15%, sedangkan untuk pelanggan yang ada adalah 50% [3].

Churn prediction merupakan rangkaian proses untuk memprediksi pelanggan yang berpotensi melakukan *churn*. Untuk melakukan prediksi maka perlu mengimplementasikan metode klasifikasi, sehingnga dapat model prediksi *churn* yang akurat. Namun untuk memaksimalkan hasil klasifikasi atau menghindari ketidak akuratan maka data harus diseimbangkan, karena sifat data churn selalu bersifat imbalance (tidak seimbang antara pelanggan *churn* dan pelanggan yang loyal). Serta karakteristik dari dataset *churn* adalah tingkat *imbalance* yang besar.

Klasifikasi data antara minoritas dan mayoritas perlu dilakukan *resampling* agar kinerja algoritma belajar (*learning algorithm*) dan distribusi kelas menjadi relatif seimbang [4]. Ketepatan dalam penentuan parameter tidak juga perlu diperhatikan saat mengevaluasi kinerja *dataset* yang tidak seimbang, karena jika data yang tidak seimbang akan menghasilkan prediksi yang menghasilkan kelas mayoritas [5]. Untuk mengatasi ketidakseimbangan terbagi menjadi tiga pendekatan, yaitu pendekatan level data, pendekatan level algoritmik, dan pendekatan gabungan atau memasangkan (*ensemble*).

Metode yang paling umum untuk menyelesaikan permasalahan *imbalance data* yaitu dengan menerapkan *sampling*, tingkat *imbalance* akan semakin kecil sehingga klasifikasi dapat dilakukan dengan tepat.

Tedapat tiga metode *sampling*, yaitu metode *oversampling*, metode *undersampling*, dan metode hibirda. Masing-masing metode memiliki perbedaan dalam menyeimbangkan data. *Oversampling* menyeimbangkan data dengan meningkatkan jumlah data kelas minoritas, Sedangkan metode

undersampling menyeimbangkan dengan cara menurunkan jumlah data kelas mayoritas, serta hibrida yaitu gabungan dari kedua metode *sampling* [6].

MWMOTE (*Majority Weighted Minority Oversampling Technique*) adalah salah satu metode *oversampling*, cara kerja algoritma ini yaitu dengan mengidentifikasi kelas minoritas yang sulit untuk dipelajari dan memberikannya bobot berdasarkan jarak Euclidean dari kelas mayoritas terdekatnya, data sintetis yang dihasilkan didapat dari kelas minoritas yang telah diberikan bobot dengan menggunakan pendekatan *clustering* [7].

Berdasarkan pemaparan tersebut maka penelitian ini dilakukan dengan tujuan untuk mengetahui hasil dari penerapan MWMOTE sebagai metode *resampling* untuk mengatasi ketidakseimbangan kelas pada data *churn* pelanggan. Penelitian ini akan mengetahui hasil pengukuran kinerja pada pendekatan level data, dengan menerapkan algoritma klasifikasi C4.5 dan menghitung nilai *confuxion matrix*.

2. TINJAUAN PUSTAKA

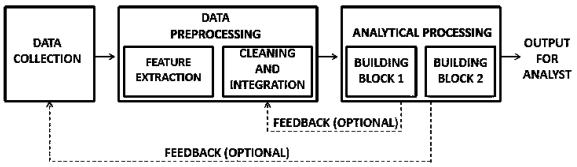
Penelitian ini tentunya memerlukan tinjauan pustaka, guna menjelaskan definisi dan deskripsi dari metode yang digunakan. Tinjauan pustaka menjabarkan informasi yang relevan dan dibutuhkan agar penelitian dapat dipahami.

2.1. *Data Mining*

Data mining merupakan rangkaian proses pemisahan informasi pada sebuah penyimpanan basis data. Tujuannya untuk mencari informasi tambahan yang belum diperoleh dari basis data, mencarian model data, dan mendapatkan informasi yang dapat dimanfaatkan untuk suatu kepentingan [8]. *Data mining* juga dapat artikan sebagai suatu proses menciptakan pola dan kecenderungan dari banyaknya informasi yang tersimpan, dari data yang tersimpan dan dengan memanfaatkan teknik pengenalan pola [9].

Proses *data mining* mengandung banyak fase seperti *data cleaning*, *feature extraction*, dan *algorithmic design*. Alur kerja dari *data mining* yang khas mengandung tahapan berikut:

1. *Data collection*, Tahap ini memerlukan penggunaan perangkat keras khusus seperti perangkat sensor, tenaga kerja manual, seperti pengumpulan data survei, atau perangkat lunak seperti mesin pencarian dokumen berbasis website. Tahap pengumpulan data sangat penting pada proses *data mining* karena pemilihan data yang baik dapat mempengaruhi proses *data mining* secara signifikan.
2. *Feature extraction and data cleaning*, Data yang dikumpulkan seringkali dalam bentuk yang tidak dapat langsung diolah. Misalnya, data dapat dikodekan kebentuk *log*. Dalam banyak kasus, berbagai jenis data dapat dicampur dalam bentuk dokumen bebas. Untuk membuat data sesuai untuk diproses, penting untuk mengubahnya menjadi format yang sesuai terhadap algoritma *data mining*, seperti *multidimensional*, *time series*, atau format *semistructured*. Perubahan format perlu dilakukan sehingga menghasilkan fitur yang relevan untuk proses penambangan. Fase *feature extraction* sering dilakukan bersamaan dengan fase *data cleaning*.
3. *Analytical processing and algorithms*: Bagian terakhir dari proses *data mining* adalah merancang metode analisis yang efektif.



Gambar 1. Proses pengolahan *data mining* [10]

2.2. Klasifikasi

Metode klasifikasi diterapkan untuk pembelajaran fungsi-fungsi berbeda dalam proses memetakan data terpilih kedalam salah satu dari kelompok kelas yang telah ditetapkan sebelumnya. Proses klasifikasi didasarkan pada empat komponen [10]:

1. Kelas (*class*)
   Kelas merupakan variabel dependen dari model yang merupakan kategori variabel yang mewakili label-label yang diletakkan pada obyek setelah klasifikasi.
2. Prediktor (*predictors*)
   Prediktor merupakan variabel independen dari model yang diwakili oleh karakteristik atau atribut dari data yang diklasifikasikan berdasarkan klasifikasi data.
3. *Dataset* pelatihan (*training dataset*)
   Training dataset merupakan *dataset* yang berisi dua komponen nilai yang digunakan dalam proses pelatihan untuk mengenali model yang sesuai berdasarkan prediktor yang ada.
4. *Dataset* pengujian (*testing dataset*)
   Testing datasate merupakan *dataset* baru yang akan diklasifikasikan sebagai model sehingga dapat dievaluasi hasil akurasi klasifikasi tersebut.

2.3. *Decision Tree*

Decision tree (pohon keputusan) merupakan salah satu metode klasifikasi yang paling populer dan banyak digunakan. Karena klasifikasi dapat dibangun dengan relatif cepat, dan hasil dari model yang dibangun mudah untuk dipahami.

Pohon keputusan menerapkan aturan-aturan klasifikasi tertentu [9]. *Decision tree* terdiri dari pohon (*tree*), yang merupakan sebuah struktur data yang terdiri dari simpul (*node*) dan rusuk (*edge*). *Node* pada sebuah pohon dibedakan menjadi tiga, yaitu simpul akar (*root*/*node*), simpul percabangan atau internal (*branch*/*internal node*) dan simpul daun (*leaf node*) [11]. Salah satu algoritma *decision tree* adalah algoritma C4.5.

2.4. Algoritma C4.5

Algoritma C4.5 adalah algoritma yang didapat dari hasil pengembangan ID3, metode ini bisa mengatasi *missing data*, dan bisa mengatasi data kontinyu serta *pruning* [9]. Algoritma C4.5 dapat membentuk pohon keputusan dari data pelatihan (*data training*), isi *data training* berupa *record-record* (*tupel*) dalam basis data. Setiap kasus berisikan nilai dari atribut-atribut untuk sebuah kelas. Setiap atribut dapat berisi data diskret atau kontinyu (numerik) [11].

Terdapat beberapan tahapan untuk membuat sebuah *decision tree* pada saat mengimplementasikan algoritma C4.5 yaitu:

1. Mempersiapkan *data training*, data ini dapat diambil dari data histori yang telah terjadi sebelumnya dan sudah dikelompokan dalam bentuk kelas-kelas tertentu.
2. Menentukan akar pohon dengan cara menghitung nilai *gain* yang tertinggi dari masing-masing atribut atau berdasarkan nilai *index entropy* terendah. Sebelumnya dihitung terlebih dahulu nilai *index entropy*, dengan rumus:
   $Entropy\ (i) = -\sum_{j=1}^{m} f(i,j).\log 2\ f[(i,j)]$
   Keterangan:
   i = himpunan kasus
   m = jumlah partisi i
   f(i,j) = proposi j terhadap i
3. Hitung nilai *gain* dengan rumus:
   $$Entropy\ split = -\sum_{i=1}^{p} \frac{n1}{n}\ .\ IE\ (i)$$
   Keterangan:
   p = jumlah partisi atribut
   ni = proporsi ni terhadap i
   n = jumlah kasus dalam bentuk n
4. Ulangi langkah ke-2 hingga semua *record* terpartisi. Proses partisi pohon keputusan akan berhenti disaat:
   a. Semua tupel dalam *record* dalam simpul m mendapat kelas yang sama
   b. Tidak ada atribut dalam *record* yang dipartisi lagi
   c. Tidak ada *record* didalam cabang yang kosong.

2.5. *Confuxion Matrix*

Confuxion matrix merupakan representasi dan kondisi sebenarnya dari data yang dihasilkan berdasarkan perhitungan algoritma *machine learning*. *Confusion matrix* menghasilkan nilai *accuracy*, *precision*, dan *recall* [9].

Confusion matrix memberikan keputusan yang diperoleh dalam *traning* dan *testing*, *confusion matrix* memberikan penilaian *performance* klasifikasi berdasarkan objek dengan benar atau salah.

Tabel 1. *Confusion Matrix*

Predicted	*Positive* (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Keterangan:

TP : *True Positive*
FP : *False Positive*
FN : *False Negative*
TN : *True Negative*

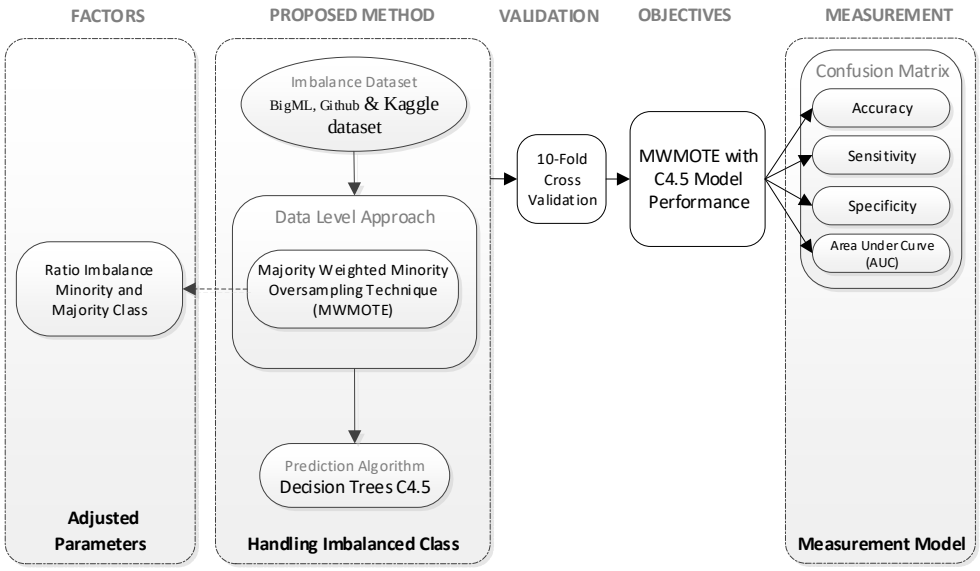
Accuracy : $\frac{(TP + TN)}{(TP+FP+FN+TN)}$

Precision (Specificity) : $\frac{TP}{(TP+FP)}$

Recall (Sensitivity) : $\frac{TP}{(TP+FN)}$

3. METODE PENELITIAN

Berikut gambaran tahapan penelitian yang dilakukan, nilai yang digunakan pada penelitian ini terdiri dari 4 nilai, yaitu *accuracy*, *sensitivity*, *specificity*, dan *area under curve* (AUC).



Gambar 2. Kerangka Pemikiran Penelitian

3.1. *Dataset*

Dataset yang digunakan pada penelitian ini menggunakan data sekunder. Data sekunder merupakan data yang diperoleh tidak secara langsung dari objek penelitian, namun biasanya berasal dari data yang telah dikumpulkan sebelumnya oleh pihak lain . Data sekunder umumnya memiliki bukti berupa catatan atau laporan historis dalam arsip (*data documenter*) yang dipublikasikan maupun yang tidak dipublikasikan.

Dataset yang digunakan pada penelitian ini sebagai berikut:

1. *Churn in the telecom industry*, *dataset* ini diambil dari https://bigml.com/dashboard/dataset/5a7836c5eba31d3437000126 memiliki 20 *attribute* dan 3333 *instance*, terdiri dari 2850 data *nonchurner* dan 483 pelanggan yang *churn*, dengan tingkat *imbalance* 85% : 15%.
2. *Customer churn*, *dataset* ini diambil dari https://github.com/navdeep-G/customer-churn memiliki 21 *attribute* dan 7032 *instance*, terdiri dari 5163 data *nonchurner* dan 1869 pelanggan yang *churn*, dengan tingkat *imbalance* 73% : 27%.
3. *Churn*, *dataset* ini diambil dari https://www.kaggle.com/filemide/churns/data memiliki 16 *attribute* dan 1397 *instance*, terdiri dari 1196 data *nonchurner* dan 201 pelanggan yang *churn*, dengan tingkat *imbalance* 86% : 14%.

3.2. Pengolahan Data Awal

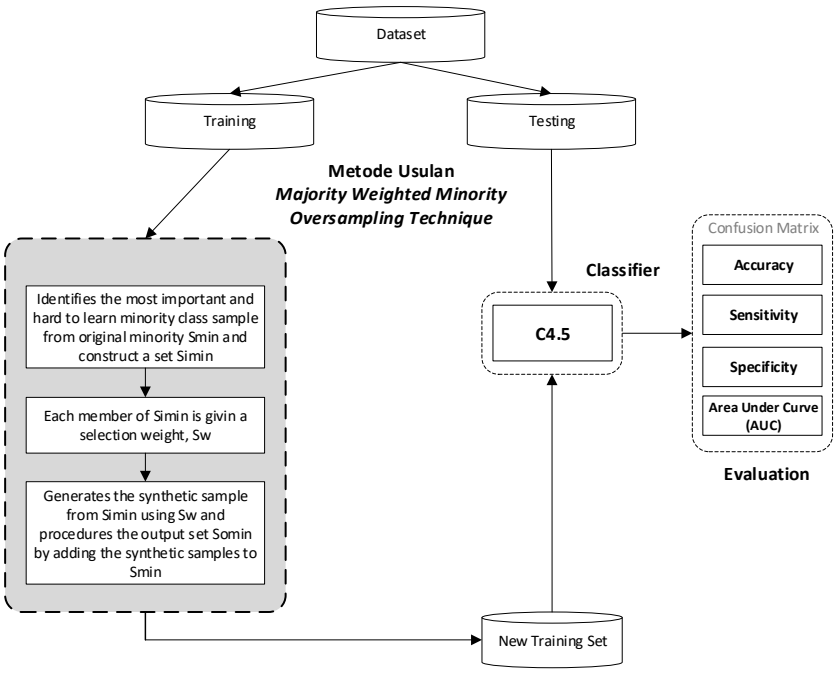
Dataset yang digunakan pada penelitian ini telah melewati *data preparation*, seperti proses validasi (membersihkan data yang bersifat *missing)* data yang bersifat *missing* akan di olah berdasarkan nilai rata-rata secara acak, proses *transformation* (merubah tipe data *text* menjadi *numeric*) data yang berisikan *text* harus dirubah dalam bentuk *numeric* agar dapat diklasifikasikan.

Sebelum melakukan klasifikasi maka pembagian data perlu dilakukan untuk tahap pembelajaran (*training*). Proses klasifikasi dibagi menjadi dua *data training* dan *data testing*, *dataset* tersebut dibagi masing-masing sebesar 80% dan 20%.

3.3. Metode yang diusulakan

Untuk mengatasi permasalahan ketidakseimbangan kelas khusunya pada perbaikan dan kelemahan *oversampling* penelitian ini menerapkan MWMOTE yang merupakan metode *oversampling* sintetis. Klasifikasi yang digunakan pada penelitian ini menggunakan algoritma C4.5. pembagian *dataset* dibagi menjadi dua yaitu *data training* dan *data testing* dengan besaran 80% dan 20%. Model evaluasi menggunakan tabel *confusion matrix*. nilai yang diukur adalah *accuracy*, *sensitifity*, *speciticity* dan AUC.

Penelitian ini menggunakan metode MWMOTE untuk menangani permasalahan pada level data dengan tujuan agar hasil algoritma klasifikasi C4.5 meningkat. Cara kerja algoritma ini yaitu dengan mengidentifikasi kelas minoritas yang sulit untuk dipelajari dan memberikannya bobot berdasarkan jarak *Euclidean* dari kelas mayoritas terdekatnya, data sintetis yang dihasilkan didapat dari kelas minoritas yang telah diberikan bobot dengan menggunakan pendekatan *clustering.*



Gambar 3. Metode Usulan

Metode MWMOTE memiliki 3 tahapan penting dalam memperbaiki permasalahan *imbalance class*. Tahap pertama MWMOTE mengindentifikasi sampel kelas minoritas yang paling penting dari *set* minoritas yang asli, S_{min} dibuat kedalam *set* oleh sampel yang teridentifikasi, tahap kedua setiap anggota S_{min} diberikan bobot sesuai data, pada tahap ketiga MWMOTE menghasilkan sampel sintetis dari S_{min} menggunakan S_{ws} dan menghasilkan *output* S_{omin} dengan menambahkan sampel sintetis ke dalam S_{min}. Pada langkah 1 sampai 6 dan 7 sampai 9 merupakan fase pertama dan fase kedua, langkah 10 sampai 13 merupakan fase ketiga. Berikut ini tahap dan *psedocode* metode MWMOTE:

Tabel 2. *Pseudocode* MWMOTE

Algorithm 1. MWMOTE *(S_{maj}; S_{min};N; k1; k2; k3).*
Input:
S_{maj}: Set of majority class samples
S_{min}: Set of minority class samples
N: Number of synthetic samples to be generated
k1: Number of neighbors used for predicting noisy minority class samples
k2: Number of majority neighbors used for constructing informative minority set
k3: Number of minority neighbors used for constructing informative minority set
Procedure Begin
1) For each minority example $\chi_i \in S_{min}$, compute the nearest neighbor set, $NN(\chi_i)$. $NN(\chi_i)$ consists of the nearest k1 neighbors of χ_i according to Euclidean distance.
2) Construct the filtered minority set, S_{minf} by removing those minority class samples which have no minority example in their neighborhood:
$S_{minf} = S_{min} - \{ \chi_i \in S_{min}$: $NN(\chi_i)$ contains no minority example$\}$
3) For each $\chi_i \in S_{minf}$, compute the nearest majority set, $Nmaj(\chi_i)$. $Nmaj(\chi_i)$ consists of the nearest k2 majority samples from xi according to euclidean distance.
4) Find the borderline majority set, S_{bmaj}, as the union of all $N_{maj}(\chi_i)$, i.e.,
$S_{bmaj} = \cap \chi_i \in S_{minf}\ N_{maj}(\chi_i)$
5) For each majority example $y_i \in S_{bmaj}$, compute the nearest minority set, $N_{min}(y_i)$. $N_{min}(y_i)$ consists of the nearest k3 minority examples from y_i according to euclidean distance.
6) Find the informative minority set, S_{imin}, as the union of all $N_{min}(y_i)_s$, i.e.,
$S_{imin} = \cap \chi_i \in Sb_{maj}\ N_{min}(\chi_i)$
7) For each $y_i \cap \chi_i \in Sb_{maj}$ Sbmaj and for each $\chi_i \in S_{imin}$, compute the information weight, $I_w(y_i; \chi_i)$.
8) For each $\chi_i \in S_{imin}$, compute the selection weight $S_w(\chi_i)$ as
$S_w(\chi_i) = \sum_{y_i \in} S_{bmaj} I_w(y_i; \chi_i)$
9) Convert each $S_w(\chi_i)$ into selection probability $S_p(\chi_i)$ according to
$Sp(\chi_i) = Sw(\chi_i) / \sum Z_{i \in Simin} Sw(z_i)$
10) Find the clusters of S_{min}. Let, M clusters are formed which are L1; L2; . . . ; LM.
11) Initialize the set, $S_{omin} = S_{min}$.
12) Do for j = 1 . . . N.
a) Select a sample x from Simin according to probability distribution $\{S_p(\chi_i)\}$. Let, x is a member of the cluster L_k; 1 < k < M.
b) Select another sample y, at random, from the members of the cluster L_k.
c) Generate one synthetic data, s, according to s = $\chi + \alpha \times (y - r)$, where α is a random number in the range[0,1].
d) Add s to S_{omin}: $S_{omin} = S_{omin} \{s\}$.
13) End Loop
End
Output: The oversampled minority set, S_{omin}.

3.4. Eksperimen dan Pengujian Metode

Pada tahap eksperimen dan pengujian model yaitu dilakukan dengan cara menghitung dan mendapatkan *rule* yang ada pada algoritma *sampling* yang digunakan. Eksperimen pada penelitian ini menggunakan dua *study*. Pertama menguji model klasifikasi C4.5 tanpa melalui proses *resampling* pada *data training*. Sedangkan yang kedua adalah tahap menguji model klasifikasi C4.5 melalui proses *resampling* menggunakan MWMOTE.

3.5. Evaluasi Model

Evaluasi dan validasi kinerja model pada eksperimen ini menggunakan *confution matrix*. Nilai yang diukur diantaranya *accuracy*, *sensitifity*, *speciticity* dan AUC. *Confution matrix* didapat dari model validasi dengan nilai 10 – *fold cross validation*.

Nilai AUC dapat dijadikan ukuran untuk melihat model yang terbentuk. Kurva ROC akan digunakan untuk menemukan nilai AUC, dimana nilai AUC digunakan untuk menentukan keakuratan akurasi secara klasifikasi. Tingkat akuransi nilai AUC dalam klasifikasi *data mining* dibagi menjadi lima kelompok, yaitu:

1. 0.90 - 1.00 = klasifikasi sangat baik (*excellent classification*)
2. 0.80 - 0.90 = klasifikasi baik (*good classification*)
3. 0.70 - 0.80 = klasifikasi cukup (*fair classification*)
4. 0.60 - 0.70 = klasifikasi buruk (*poor classification*)
5. 0.50 - 0.60 = klasifikasi salah (*failure*)

4. HASIL DAN PEMBAHASAN

4.1. Hasil Eksperimen

Hasil pengujian dari model yang diusulkan, pengujian dilakukan dengan menggunakan tiga *dataset churn* yang berbeda dengan jumlah kelas minoritas dan mayoritas yang tidak seimbang. Pada penelitian ini eksperimen menggunakan spesifikasi

software dan *hardware* sebagai alat bantu dalam pembuatan model penelitian. Berikut adalah spesifikasi lengkap mengenai *software* dan *hardware* yang digunakan.

Tabel 3. Spesifikasi *Hardware* dan *Software*

Processor	Intel® Core™ i3 – 4005U 1.70 GHz
Memory	4GB
Hardisk	500 GB
Sistem Operasi	Microsoft Windows 7
Aplikasi	RStudio 1.1.3

4.2. Hasil Pengujian Metode Usulan

Pengujian model dilakukan menggunakan tiga *dataset churn*. Pengujian untuk mengatasi ketidakseimbangan kelas dilakukan dengan cara meningkatkan kelas minoritas (*oversampling*) menggunakan teknik MWMOTE.

Eksperimen yang dilakukan harus secara adil dalam menentukan rasio *resampling* yang optimal padasemua pengujian. Untuk itu *data training* dan *testing* dengan besaran 80% *data training* dan 20% *data testing*. Pada bagian ini akan menampilkan hasil pengujian dari metode usulan serta menjadi factor penentu hasil kinerja model.

1. *Dataset Churn* BigML

   Dataset ini terdiri dari 20 *variabel* dan 3.333 data sampel persentase ketidakseimbangan kelas adalah 85% *negative class* dan 15% *positive class*.

```
$ State                 : Factor w/ 51 levels "AK","AL","AR",..: 17 36 32 36 37 2 20 25 19 50 ...
$ Account.length        : int  128 107 137 84 75 118 121 147 117 141 ...
$ Area.code             : int  415 415 415 408 415 510 510 415 408 415 ...
$ International.plan    : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 2 1 2 1 2 ...
$ Voice.mail.plan       : Factor w/ 2 levels "No","Yes": 2 2 1 1 1 1 2 1 1 2 ...
$ Number.vmail.messages : int  25 26 0 0 0 0 24 0 0 37 ...
$ Total.day.minutes     : num  265 162 243 299 167 ...
$ Total.day.calls       : int  110 123 114 71 113 98 88 79 97 84 ...
$ Total.day.charge      : num  45.1 27.5 41.4 50.9 28.3 ...
$ Total.eve.minutes     : num  197.4 195.5 121.2 61.9 148.3 ...
$ Total.eve.calls       : int  99 103 110 88 122 101 108 94 80 111 ...
$ Total.eve.charge      : num  16.78 16.62 10.3 5.26 12.61 ...
$ Total.night.minutes   : num  245 254 163 197 187 ...
$ Total.night.calls     : int  91 103 104 89 121 118 118 96 90 97 ...
$ Total.night.charge    : num  11.01 11.45 7.32 8.86 8.41 ...
$ Total.intl.minutes    : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
$ Total.intl.calls      : int  3 3 5 7 3 6 7 6 4 5 ...
$ Total.intl.charge     : num  2.7 3.7 3.29 1.78 2.73 1.7 2.03 1.92 2.35 3.02 ...
$ Customer.service.calls: int  1 1 0 2 3 0 3 0 1 0 ...
$ Churn                 : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
```

Gambar 4. Struktur *dataset Churn* BigML

Persentase ketidakseimbangan setelah dilakukan teknik *resample oversampling* persentase ketidakseimbangan kelas *training set* menjadi 54% *negative class* dan 46% *positive class*.

Berdasarkan training *set* diatas, model prediksi dibentuk kemudian diukur performa model diukur menggunakan *confusion matrix*. Berikut ini adalah perhitungan manual dari kinerja model.

a. Tanpa *Resampling*

Tabel 4. *Confusion Matrix* Tanpa *Resample Dataset* BigML

Class		*Actual*	
		Churn	*NonChurn*
Predictor	*Churn*	68	7
	NonChurn	26	565

Accuracy = (TP + TN) / (TP + TN + FP + FN)
= 68 + 565 / 68 + 565 + 26 + 7
= 633 / 666
= 0,9504

Sensitivity = TP / (TP + FN)
= 68 / (68 + 26)
= 68 / 94
= 0,7234

Specificity = TN / (TN + FP)
= 565 / (565 + 7)
= 565 / 572
= 0,9878

AUC = (*sensitivity* + *specificity*) / 2
= (0,7234 + 0,9848) / 2
= 1,7112 / 2
= 0,8556

b. Model MWMOTE

Tabel 5.*Confusion Matrix* MWMOTE *Dataset* BigML

Class		*Actual*	
		Churn	*NonChurn*
Predictor	*Churn*	74	28
	NonChurn	20	544

Accuracy = (TP + TN) / (TP + TN + FP + FN)
= 74 + 544 / 74 + 544 + 20 + 28
= 618 / 666
= 0,9279

Sensitivity = TP / (TP + FN)
= 74 / (74 + 20)
= 74 / 94
= 0,7872

Specificity = TN / (TN + FP)
= 544 / (544 + 28)
= 544 / 572
= 0,9510

AUC = (*sensitivity* + *specificity*) / 2
= (0,7872 + 0,9510) / 2
= 1,738 / 2
= 0,869

2. *Dataset Churn* Telco

   Dataset ini terdiri dari 20 *variabel* dan 7.032 data sampel persentase ketidakseimbangan kelas adalah 73% *negative class* dan 27% *positive class*.

```
$ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
$ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
$ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2 1 ...
$ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2 ...
$ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
$ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 2 1 2 2 ...
$ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1 2 1 2 1 3 3 2 3 1 ...
$ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2 2 2 1 2 1 ...
$ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1 3 3 3 1 1 1 3 1 3 ...
$ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3 1 3 1 1 1 3 1 1 3 ...
$ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1 3 1 3 1 3 1 1 3 1 ...
$ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 3 1 1 1 1 3 1 ...
$ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
$ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1 1 1 1 1 3 1 1 3 1 ...
$ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1 1 1 2 ...
$ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
$ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3 4 4 1 3 3 2 4 3 1 ...
$ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
$ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
$ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
```

Gambar 5. Struktur *Dataset Churn* Telcom

Persentase ketidakseimbangan setelah dilakukan teknik *resample oversampling* persentase ketidakseimbangan kelas *training set* menjadi 52% *negative class* dan 48% *positive class*.

Berdasarkan *training set* diatas, model prediksi dibentuk kemudian diukur performa model diukur menggunakan *confusion matrix*. Berikut ini adalah perhitungan manual dari kinerja model.

a. Model Tanpa *Resampling*

Tabel 6. *Confusion Matrix* Tanpa *Resample Dataset* Telco

Class		*Actual*	
		Churn	*NonChurn*
Predictor	*Churn*	193	116
	NonChurn	203	894

Accuracy = (TP + TN) / (TP + TN + FP + FN)
= 193 + 894 / 193 + 894 + 203 + 116
= 1087 / 1406
= 0,7731

Sensitivity = TP / (TP + FN)
= 193 / (193 + 203)
= 193 / 396
= 0,4874

Specificity = TN / (TN + FP)
= 894 / (894 + 116)
= 894 / 1010
= 0,8851

AUC = (*sensitivity* + *specificity*) / 2
= (0,4874 + 0,8851) / 2
= 1,3725 / 2
= 0,6863

b. Model MWMOTE

Tabel 7. *Confusion Matrix* MWMOTE *Dataset* Telco

Class		*Actual*	
		Churn	*NonChurn*
Predictor	*Churn*	230	160
	NonChurn	166	850

Accuracy = (TP + TN) / (TP + TN + FP + FN)
= 230 + 850 / 230 + 850 + 166 + 160
= 1080 / 1406
= 0,7681

Sensitivity = TP / (TP + FN)
= 230 / (230 + 166)
= 230 / 396
= 0,5808

Specificity = TN / (TN + FP)
= 850 / (850 + 160)
= 850 / 1010
= 0,8416

AUC = (*sensitivity* + *specificity*) / 2
= (0,5808 + 0,8416) / 2
= 1,422 / 2
= 0,7112

3. *Dataset Churn* Kaggle
Dataset ini terdiri dari 15 *variabel* dan 1.397 data sampel persentase ketidakseimabangan kelas adalah 86% *negative class* dan 14% *positive class*.

```
$ network_age                       : int  117 123 1342 1316 247 425 698 619 261 4082 ...
$ Customer_tenur_in_month           : num  3.9 4.1 44.73 43.87 8.23 ...
$ Total_Spend_in_Months_1_and_2_of_2017  : num  49.7 76.7 76.9 98.9 152.9 ...
$ Total_SMS_Spend                   : num  0 0 11.96 4.14 0.02 ...
$ Total_Data_Spend                  : num  38.75 1.25 1.25 1.25 15 ...
$ Total_Data_Consumption            : num  6.94e+03 1.50 7.51 1.02 3.24e+07 ...
$ Tota_Unique_calls                 : int  1 14 5 27 6 4 8 17 26 79 ...
$ Total_Onnet_spend                 : int  0 564 251 1626 12 114 1372 3972 228 2640 ...
$ Total_Offnet_spend                : int  1092 6408 1004 4373 1145 1594 632 2339 5008 16450 ...
$ Total_Cal_Centre_complaint_calls  : int  1 2 2 1 2 1 4 1 1 2 ...
$ Network_type_subscription_in_Month_1  : int  0 1 2 1 0 2 2 0 2 0 ...
$ Network_type_subscription_in_Month_2  : int  2 1 2 1 2 2 2 0 2 0 ...
$ Most_Loved_Competitor_network_in_Month_1: int  1 5 3 5 1 2 1 3 5 3 ...
$ Most_Loved_Competitor_network_in_Month_2: int  1 1 1 1 1 1 1 1 1 1 ...
$ Churn                             : int  0 0 0 0 0 0 0 0 0 0 ...
```

Gambar 6. Struktur *Dataset Churn* Kaggle

Persentase ketidakseimbangan setelah dilakukan teknik *resample oversampling* persentase ketidakseimbangan kelas *training set* menjadi 51% *negative class* dan 49% *positive class*.

Berdasarkan training *set* diatas, model prediksi dibentuk kemudian diukur performa model diukur menggunakan *confusion matrix*. Berikut ini adalah perhitungan manual dari kinerja model.

a. Model Tanpa *Resampling*

Tabel 8. *Confusion Matrix* Tanpa *Resample Dataset* Kaggle

Class		*Actual*	
		Churn	*NonChurn*
Predictor	*Churn*	0	0
	NonChurn	40	239

Accuracy = (TP + TN) / (TP + TN + FP + FN)
= 0 + 239 / 0 + 239 + 40 + 0
= 239 / 279
= 0,8566

Sensitivity = TP / (TP + FN)
= 0 / (0 + 40)
= 0 / 40
= 0

Specificity = TN / (TN + FP)
= 239 / (239 + 0)
= 239 / 239
= 1

AUC = (*sensitivity* + *specificity*) / 2
= (0 + 1) / 2
= 1 / 2
= 0,5

b. Model MWMOTE

Tabel 9. *Confusion Matrix* MWMOTE *Dataset* Kaggle

Class		*Actual*	
		Churn	*NonChurn*
Predictor	*Churn*	13	41
	NonChurn	27	198

Accuracy = (TP + TN) / (TP + TN + FP + FN)
= 13 + 198 / 13 + 198 + 27 + 41
= 211 / 279
= 0,7563

Sensitivity = TP / (TP + FN)
= 13 / (13 + 27)
= 13 / 40
= 0,325

Specificity = TN / (TN + FP)
= 198 / (198 + 41)
= 198 / 239
= 0,8284

AUC = (*sensitivity* + *specificity*) / 2
= (0,325 + 0,8284) / 2
= 1,534 / 2
= 0,5767

4.3. Rekap Hasil Pengujian

Rekap hasil pengukuran kinerja model pada masing-masing *dataset* menggunakan *confusion matrix*. Hasil pengujian menggunakan teknik *oversampling* dan tanpa menggunakan teknik *resampling* menggunakan algoritma klasifikasi C4.5. Nilai yang ditampilkan hasil kinerja model pada masing-masing *dataset*.

Tabel 10. Pengukuran *accuracy* pada tiga *dataset*

Accuracy	***Dataset***		
	BigML	Telco	Kaggle
C4.5	0,9509	0,7731	0,8566
MWMOTE + C4.5	0.9279	0.7681	0.7563

Tabel 11. Pengukuran *sensitivity* pada tiga *dataset*

Sensitivity	***Dataset***		
	BigML	Telco	Kaggle
C4.5	0,7234	0,4874	0
MWMOTE + C4.5	0.7872	0.5808	0.325

Tabel 12. Pengukuran *specificity* pada tiga *dataset*

Specificity	***Dataset***		
	BigML	Telco	Kaggle
C4.5	0,9878	0,8851	1
MWMOTE + C4.5	0.951	0.8416	0.8284

Tabel 13. Pengukuran AUC pada tiga dataset

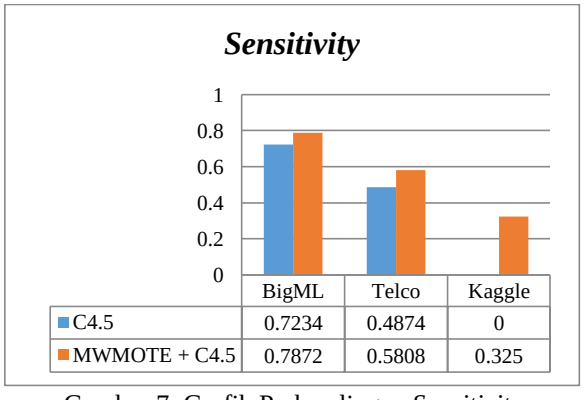
AUC	***Dataset***		
	BigML	Telco	Kaggle
C4.5	0,8556	0,6863	0,5
MWMOTE + C4.5	0.8691	0.7112	0.5767

4.4. Perbandingan Hasil

Berikut hasil pebandingan menggunakan grafik.

4.4.1. Perbandingan Nilai Sensitivity

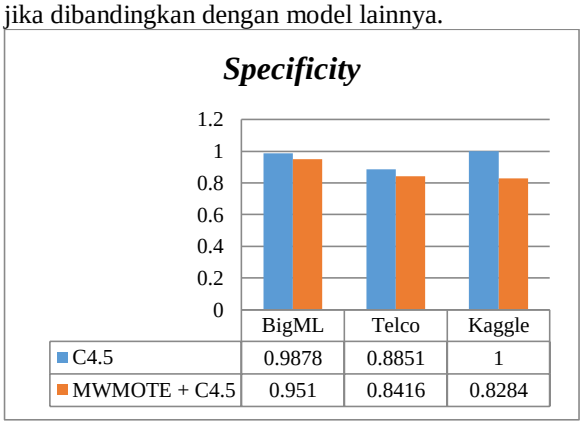
Pada grafik dibawah ini menunjukan hasil pengukuran nilai sensitivity. Melihat nilai sensitivity pada metode yang diusulkan yaitu MWMOTE dengan algoritma C4.5 merupakan hasil tertinggi jika dibandingkan dengan model lainnya.



Gambar 7. Grafik Perbandingan *Sensitivity*

4.4.2. Perbandingan Nilai Specificity

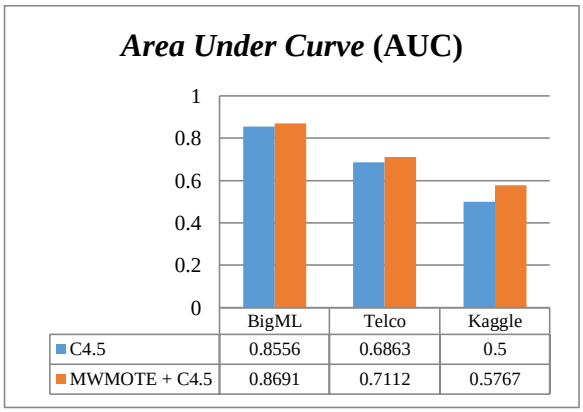
Pada grafik dibawah ini menunjukan hasil pengukuran nilai specificity. Melihat nilai specificity pada metode yang diusulkan yaitu MWMOTE dengan algoritma C4.5 merupakan hasil tengah, tidak menjadi nilai tertinggi dan juga tidak menjadi nilai terendah jika dibandingkan dengan model lainnya.



Gambar 8. Grafik Perbandingan *Specificity*

4.4.3. Perbandingan Nilai AUC

Pada grafik dibawah ini menunjukan hasil pengukuran nilai AUC. Melihat nilai AUC pada metode yang diusulkan yaitu MWMOTE dengan algoritma C4.5 merupakan hasil menjadi nilai tertinggi jika dibandingkan dengan model lainnya.



Gambar 9. Grafik Perbandingan AUC

Berdasarkan pengujian yang dilakukan dan dilihat dari grafik hasil perbandingan jika dibandingkan dengan metode C4.5 tanpa *resampling*, metode usulan meningkat dari nilai *sensitivity* sebesar 0.16073%, nilai *specificity* menurun sebesar 0.084% dan nilai AUC meningkat sebesar 0.03837% hal ini membuktikan bahwa *dataset* yang tidak seimbang menyebabkan algoritma pengklasifikasi tidak optimal.

Selain itu dilihat dari grafik perbandingan *sensitivity*, *specificity*, dan AUC menunjukan bahwa model MWMOTE lebih tinggi nilai *sensitivity* 0,0768%. Nilai AUC MWMOTE meningkat ddibandingkan dengan tanpa MWMOTE. Dari hasil penelitian yang telah di uji pada tiga *dataset churn* yang berbeda, menunjukan MWMOTE lebih unggul.

5. KESIMPULAN DAN SARAN

Penelitian terkait ketidakseimbangan kelas mencoba untuk mengatasi ketidakseimbangan. Pada penelitian ini teknik yang digunakan adalah *oversampling*, tujuan penelitian ini mengatasi

permasalahan *imbalance class* pada *churn prediction* agar performa algoritma klasifikasi C4.5 meningkat. Eksperimen ini dilakukan dengan menguji coba tiga *dataset churn imbalance* yang didapat dari situs *dataset repository*. Hasil penelitian yang sudah dilakukan dapat dilihat pada hasil grafik perbandingan nilai *sensitivity*, *specificity* dan AUC. Hasil tersebut menunjukan peningkatan yang signifikan pada nilai *sensitifity* dan nilai AUC ketika menggunakan MWMOTE dibandingkan tanpa menggunakan model *resampling*. Model MWMOTE membuktikan menghasilkan nilai yang lebih baik pada algoritma klasifikasi C4.5 *dataset churn prediction*.

Penelitian lainnya disarankan menguji hasil klasifikasi menggunakan algoritma klasifikasi lainnya seperti C5.0, Naïve Bayer, *Random Forest*, *Logistic regression*. Melakukan pendekatan level data yang dipadukan antara metode *oversampling* dan *undersampling*, agar terhindar dari efek buruk kedua metode tersebut, seperti terjadinya *over fitting* maupun hilangnya data yang bersifat *informative*.

Melakukan pendekatan level algoritma seperti metode *Adaboost* yang telah terbukti dapat meningkatkan hasil klasifikasi.

DAFTAR PUSTAKA

[1] I. M. Latief, A. Subekti, and W. Gata, "Prediksi Tingkat Pelanggan Churn Pada Perusahaan Telekomunikasi Dengan Algoritma Adaboost," *J. Inform.*, vol. 21, no. 1, pp. 34–43, 2021, doi: 10.30873/ji.v21i1.2867.

[2] B. Huang, M. T. Kechadi, and B. Buckley, "Customer Churn Prediction in Telecommunications," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1414–1425, Jan. 2012, doi: 10.1016/j.eswa.2011.08.024.

[3] X.-Y. Liu and Z.-H. Zhou, "Ensemble Methods for Class Imbalance Learning," in *Imbalanced Learning*, John Wiley & Sons, Ltd, 2013, pp. 61–82.

[4] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognit.*, vol. 40, no. 12, pp. 3358–3378, 2007, doi: https://doi.org/10.1016/j.patcog.2007.04.009.

[5] T. M. Khoshgoftaar, K. Gao, and N. Seliya, "Attribute Selection and Imbalanced Data: Problems in Software Defect Prediction," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, 2010, vol. 1, pp. 137–144, doi: 10.1109/ICTAI.2010.27.

[6] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," in *Artificial Intelligence in Medicine*, 2001, pp. 63–66.

[7] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE--Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, 2014, doi: 10.1109/TKDE.2012.232.

[8] I. Zulfa, R. Rayuwati, and K. Koko, "Implementasi data mining untuk menentukan strategi penjualan buku bekas dengan pola pembelian konsumen menggunakan metode apriori," *Tek. J. Sains dan Teknol.*, vol. 16, no. 1, p. 69, 2020, doi: 10.36055/tjst.v16i1.7601.

[9] D. N. Sari, H. Oktavianto, and I. Saifudin3, "Penerapan Data Mining Untuk Klasifikasi Gaya Belajar Siswa Menggunakan Algoritma C4.5 Application," *J. Smart Teknol.*, vol. 3, no. 2, pp. 184–190, 2022.

[10] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining*. 2008.

[11] A. H. Nasrullah, "Implementasi Algoritma Decision Tree Untuk Klasifikasi Produk Laris," *J. Ilm. Ilmu Komput.*, vol. 7, no. 2, pp. 45–51, 2021, doi: 10.35329/jiik.v7i2.203.