Data Balancing untuk Mengatasi Imbalance Dataset pada Prediksi Produksi Padi

Balancing Data to Overcome Imbalance Dataset on Rice Production Prediction

Khafid Akbar*[1], Mardhiya Hayaty[2]
[1,2,3] *Program Studi Informatika , Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta*
e-mail: *[1]khafid.akbar@students.amikom.ac.id, [2]mardhiya_hayati@amikom.ac.id

Abstrak

Padi adalah salah satu hasil tanaman pangan di Indonesia yang merupakan salah satu makanan pokok. Tingginya permintaan untuk produksi beras telah mengakibatkan sejumlah daerah di Indonesia membuat lumbung padi dengan setidaknya delapan daerah, yang salah satunya di Jawa Timur. Kebutuhan untuk memproses data terkait pasokan makanan, terutama beras, merupakan salah satu faktor penting untuk mengantisipasi tingkat permintaan di masa depan. Langkah yang dilakukan dalam penelitian ini untuk mencapai hasil yang diharapkan adalah dengan melakukan proses penambangan data yaitu mengumpulkan data, preprocessing data, mengimplementasikan algoritma pada data yang ada, dan mengevaluasi hasil. Dalam penelitian ini, peneliti menggunakan dua algoritma klasifikasi untuk menguji keakuratan data balancing, yaitu Naive Bayes dan CART. Proses balancing yang dilakukan oleh peneliti menggunakan metode balancing data dengan algoritma SMOTE. Dengan langkah-langkah yang peneliti lakukan di atas menghasilkan akurasi algoritma Naive Bayes dengan data sebelum menyeimbangkan 43,8% dengan nilai AUC 0,373 dan setelah menyeimbangkan data menghasilkan akurasi 39,06% dengan nilai AUC 0,475. Untuk algoritma CART, nilai akurasi sebelum menyeimbangkan data adalah 47,67% dengan nilai AUC 0,391, kemudian akurasi yang dihasilkan setelah saldo data untuk algoritma CART mencapai 55,73% dengan nilai AUC 0,492. dengan demikian menunjukkan pengaruh keseimbangan dan ketidakseimbangan data terhadap kinerja algoritma klasifikasi Naive Bayes dan CART.

Kata kunci*—data mining, naïve bayes, CART, produksi padi*

Abstract

Rice is one of the results of planting food crops in Indonesia which is one of the staple foods. The high demand for rice production has resulted in a number of regions in Indonesia making its rice granary / rice producing regions with at least eight regions, one of which is in East Java. This continues to grow from year to year following the rapid growth of the population of Indonesia. The need to process data related to food supply, especially rice, is one of the important factors to anticipate the level of future demand. The steps taken in this study to achieve the expected results are by conducting a data mining process that is collecting data, preprocessing data, implementing algorithms on existing data, and evaluating the results that emerge. In this case the researchers used two classification algorithms to test the accuracy of the imbalance and balance data, namely Naive Bayes and CART. Data balancing process conducted by researchers in this case uses the data balancing method with the SMOTE algorithm. With the steps that the researchers did above resulted in an accuracy of the Naive Bayes algorithm with data before balancing of 43.8% with an AUC value of 0.373 and after balancing the data resulted in an accuracy of 39.06% with an AUC value of 0.475. For CART algorithm, the accuracy value before balancing data is 47.67% with AUC value of 0.391, then the accuracy generated after data balance for CART algorithm reaches 55.73% with AUC value of 0.492. thus showing the influence of data balance and imbalance on the performance of the Naive Bayes classification algorithm and CART.

Keywords—data mining, naïve bayes, CART, rice production

Informasi Artikel:
Submitted: Mei 2020, **Accepted:** September 2020, **Published:** November 2020
ISSN: 2685-4902 (media online), Website: http://jurnal.umus.ac.id/index.php/intech

PENDAHULUAN

Indonesia merupakan wilayah yang sebagian besar penduduknya berprofesi sebagai petani / sektor pertanian, hal ini diterbitkan oleh BPS (Badan Pusat Statistik) pada tahun 2019 yang menyatakan sekitar 33 juta jiwa penduduk Indonesia yang berkerja disektor pertanian[1]. Sektor pertanian yang luas ini membuka peluang bagi Indonesia sebagai modal besar membangun negara terutama pada sektor padi yang menjadi makanan pokok masyarakat Indonesia[1]. Sedikit banyaknya hasil panen padi tentu ditentukan oleh banyak faktor internal maupun external seperti bibit padi, kontur tanah, cara pengolahan lahan, cuaca, pengairan, dll. Perlunya pengawasan terhadap produksi padi tentu akan berpengaruh terhadap penyebaran penjualan padi maupun pengolahan pasca panen seperti mengetahui jumlah stok padi yang akan ada guna menentukan perlakuan yang harus dilakukan pemerintah untuk peningkatan jumlah produksi padi berkualitas di Indonesia. Dengan mengetahui akan hal itu tentu pemerintah dapat lebih bijak dalam mengambil keputusan. Dalam pengambilan keputusan tentu pemerintah membutuhkan hasil dari pengolahan data yang ada untuk menunjang keputusannya. hal ini berkaitan dengan teknik data mining seperti klasifikasi, clustering, regression[2].

Saat ini perkembangan ilmu pengetahuan sudah sangat pesat terutama terkait dengan data mining, bahkan penerapannya sudah menyentuh sektor-sektor penting seperti sektor pendidikan, perdagangan, kependudukan, bahkan pertanian. Penerapan data mining dalam sektor pertanian dapat diterapkan di berbagai tempat seperti pada penentuan tingkat kesuburan tanah dengan parameter jenis tanah yaitu litosol, gluosol, rigosol, orgasonol, humus, dan bergambut[3]. Kemudian, ada juga yang membahas terkait pengaruh perubahan iklim terhadap produksi pertanian[4]. Pada bidang pendidikan dapat digunakan untuk memprediksi jumlah pendaftaran mahasiswa baru [5]. Penerapan teknik data mining yaitu prediksi tentu dapat meningkatkan tingkat keakurasian dalam pemerintah memutuskan sesuatu. Dengan adanya penelitian terkait prediksi hasil panen padi diharapkan dapat digunakan sebagai antisipasi terhadap impor bahan pangan yang berlebih atau tidak sesuai dengan data yang ada.

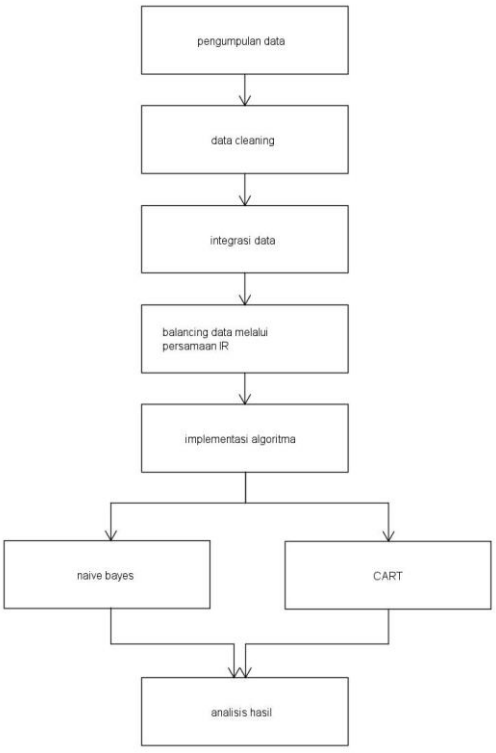
Berdasarkan uraian permasalahan diatas, maka penulis merumuskan masalah sebagai berikut bagaimana pengaruh *balancing data* terhadap akurasi dari setiap algoritma klasifikasi yang diuji? Adapun batasan masalah dalam penelitian ini yaitu lokasi pengujian yang dipilih adalah wilayah Jawa Timur dengan variabel yang akan diuji adalah luas panen, curah hujan, dan jumlah produksi padi serta dataset yang diambil bersumber dari BPS (Badan Pusat Statistik) Jawa Timur dari tahun 2000 – 2016. Sedangkan tujuan dari penelitian yang dilakukan adalah untuk mengetahui dan menganalisis pengaruh *balancing data* terhadap algoritma Naïve Bayes atau CART untuk melakukan prediksi pada hasil produksi padi yang nantinya akan menghasilkan sebuah ilmu pengetahuan yang dapat diterapkan maupun dikaji lebih lanjut.

METODE PENELITIAN

Tahapan Penelitian

Pada penelitian ini, penulis melakukan beberapa tahapan seperti yang terangkum pada Gambar 1 melalui pendekatan secara *prototype*, karena mampu mendefinisikan semua kebutuhan *software* secara obyektif[6]. Data Mining adalah sebuah aktifitas dan bukanlah sebuah algoritma atau program. Dalam pelaksanaan aktifitas Data Mining maka seringkali digunakan berbagai teknik ataupun algoritma yang berasala dari berbagai disiplin ilmu misalnya statistik, *artificial intelligence* ataupun *machine learning*[2]. Proses datamining tidak serta merta langsung mengolah dataset menggunakan sebuah algortima namun mempunya langkah – langkah atau prosedur yang harus ditempuh agar tercapainya tujuan mengolah data menggunakan Teknik data mining. Proses tersebut sering diberi nama proses *Knowledge Discovery* Database (KDD) dengan fase sebagai berikut:

a. Seleksi Data (*Selection*)
   Data seleksi merupakan sekumpulan data operasional yang perlu dilakukan pemilihan informasi sebelum melakukan proses selanjutnya yang kemudian disimpan untuk selanjutnya dilakukan tahap *preprocessing*.
b. Pemilihan Data (*Preprocessing* / *Cleaning*)
   Data *Preprocessing* merupakan langkah yang dilakukan untuk menyesuaikan data atau informasi yang relevan untuk digunakan dengan metode / cara yaitu melakukan duplikasi data, memeriksa data yang inkonsisten dan memperbaiki kesalahan dalam data.
c. Transformasi
   Dalam tahap ini proses yang dilakukan adalah mentransformasikan bentuk data yang belum memiliki entitas yang jelas kedalam bentuk data yang sudah siap dilakukan proses data mining.
d. Pengaplikasian Metode
   Fase ini merupakan fase dimana algoritma diterapkan untuk mendapatkan sebuah solusi atau *knowledge*.
e. Interpretasi / Evaluasi
   Pada fase terakhir ini merupakan fase dimana hasil dari proses pengaplikasian algoritma diolah agar mudah dimengerti dan bersumber pada proses data mining serta melakukan pengujian terhadap algoritma tersebut sesuai dengan metode pengujian yang ada.



Gambar 1. Kerangka penelitian

Algoritma Naïve Bayes

Metode naïve bayes pertama kali ditemukan pada abad ke-18 oleh seorang ilmuan bernama Thomas bayes dan teori tersebut dinamai teorema Bayes[7]. Dalam teorema Bayes mempunyai probabilitas sebagai berikut.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (1)$$

Dalam teorema probabilitas diatas, X merupakan sebuah tuple atau objek data, sedangkan H adalah hipotesis atau dugaan bahwa tuple X adalah kelas C. Penjelasan lebih mudahnya probabilitas P(H|X) ialah saat H benar untuk bukti X. Sedangkan P(X|H) adalah probabilitas bahwa bukti X benar untuk hipotesis. Kemudian P(H) adalah probabilitas prior hipotesis H dan P(X) adalah probabilitas prior bukti X[8]. Penggunaan probabilitas dapat diaplikasikan dalam beberapa interval nilai misalnya dibagi menjadi tiga interval yaitu untuk kondisi "Panggilan" dibagi kedalam tiga nilai: Sedikit, Cukup, dan Banyak sedangkan "Blok" dibagi kedalam tiga nilai: Rendah, Sedang dan Tinggi. Kemudian untuk pengukuran data statistik probabilitas masuknya data baru kedalam masing-masing kelas dapat menggunakan algoritma Naïve Bayes. Algoritma Naïve Bayes mempunyai dua probabilitas yaitu untuk data kategorial dan untuk data kontinue. Untuk data kontinue perhitungan probabilitasnya ialah sebagai berikut[7].

$$P(x_k|C_i) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x_k-\mu_{ik})^2}{2\sigma_{ik}^2}} \quad (2)$$

Dengan:

σ_{ik} = Standar deviasi
μ_{ik} = Rata-rata
x_k = Nilai pada atribut yang diuji
C_i = Kelas yang diuji

CART

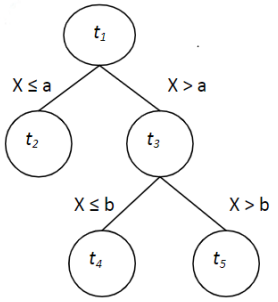
CART (*Classification and Regression Tree*) merupakan sebuah algoritma dari salah satu teknik pengambilan keputusan menggunakan pohon keputusan. Algoritma CART memiliki tujuan yaitu mendapatkan sebuah kelompok data yang kuat kemudian digunakan sebagai penciri dalam suatu klasifikasi. Pada dasarnya CART akan menggambarkan hubungan antara variabel respon (*variabel dependen*) dengan satu atau lebih variabel prediktor (*variabel independen*), model pohon yang terbentuk sangat bergantung pada skala variabel respon yang ada[9]. Jika variabel respon memiliki bentuk kategorial maka CART akan mengidentifikasinya menggunakan klasifikasi sedangkan jika variabel respon berupa data berbentuk *numeric* dan kontinue, maka CART akan mengidentifikasinya menggunakan *regression*.

Algoritma CART tersebut memiliki kekurangan dan kelebihan tersendiri dibandingkan dengan algoritma lainnya. Kelebihan CART jika dibandingkan dengan algoritma klasifikasi yang lain adalah metode klasifikasi ini lebih mudah di interpretasikan, mempunyai tingkat akurasi yang tinggi. CART juga dapat melakukan *handling* variabel dalam jumlah banyak dengan skala variabel campuran melalui prosedur pemilihan biner[10]. Sedangkan kekurangan yang dimiliki CART ialah hasil akhir tidak didasarkan pada model probabilistik, tidak ada tingkat probabilitas atau selang kepercayaan yang berhubungan dengan dugaan yang didapat dari pohon CART untuk pengelompokan data baru. Tingkat kepercayaan dalam keakurasian CART benar-benar didasarkan pada keakuratan saat membuat pohon keputusan.

a. *Binary Recursive Partitioning*

Proses pembuatan pohon keputusan dalam prosedur pemecahan CART biasa dikenal dengan istilah *Binary Recursive Partitioning*. Proses pemecahan ke dalam 2 *child node* merupakan penjabaran dari istilah *binary*[11]. Sedangkan *recursive* berarti bahwa proses pemecahan tersebut akan diulang kembali dalam setiap *child nodes output* pemecahan terdahulu, sehingga *child nodes* tadi sekarang sebagai *parent*. Proses pemecahan ini akan terus dilakukan sampai nir terdapat kesempatan lagi buat melakukan pemecahan berikutnya. Dan untuk proses *learning sample* yang dipecah ke dalam bagian-bagian atau partisi-partisi yang lebih kecil merupakan penjabaran dari isitilah *partitioning* [12]. Untuk memilih kriteria pemecahan yang didasarkan pada nilai menurut variabel independen yg terdapat. Dapat dimisalkan jika variabel dependen y yang bertipe kategorik dan variabel independen $1, 2, \cdots, p$. Kemudian dilakukanlah proses *splitting* yang nantinya akan berlanjut hingga *leaf node*[9].

Pada proses tersebut tergambar bahwa node t_1 dipilah berdaarkan kriteria pemecahan $x \leq a$ dan $x > a$, hal tersebut menghasilkan node t_2 dan t_3. Kemudian proses selanjutnya adalah pemilahan t_3 berdasarkan kriteria pemecahan $x \leq$ b dan $x >$ b, yang menghasilkan node t_4 dan t_5.



Gambar 2. Proses Partisi

b. Proses Klasifikasi CART

Proses pemecahan node pertama diawali dengan menentukan *root node* berdasarkan pada *goodness of split criterion* (kriteria pemecahan terbaik) berdasarkan nilai Gini index dari kelas variabel yang ada, rumus Gini Index dapat ditulis sebagai berikut.

$$i(t) = 1 - \sum_{j=1}^{j} P_j^2 \quad (3)$$

Dimana nilai j adalah total dari semua kategori, Pj adalah proporsi dari Jth kategori dalam node t, yang mana

$$P_j = \frac{P(j|t)}{P(t)} = \frac{\frac{N_j(t)}{N}}{\frac{N(t)}{N}} = \frac{N_j(t)}{N(t)} \quad (4)$$

Dengan:

N_t = Banyaknya objek atau cases dalam $\mathscr{L}$ yang mana $x_0 \in t$ (banyaknya objek dalam node-t).

$N_j(t)$ = Banyaknya objek atau cases kelas j yang berada dalam node t

$\frac{N_j(t)}{N_j}$ = proporsi objek-objek dalam kelas j yang berada di node t

$P(j,t)$ =probabilitas bahwa sebuah objek adalah anggota kelas j dan berada dalam node t.

Sehingga $\sum_{j=1}^{j} P_j = 1$.

Kemudian untuk melakukan pengujian *goodness of split criteration* (kriteria uji pemecahan terbaik) dapat digunakan persamaan $\Delta i(s,t)$ yang mana split s akan digunakan untuk memecah node t menjadi dua buah node yaitu tR (right node) dan tL (left node) jika nilai s berikut dimaksimalkan[13][14].

$$\Delta \mathrm{i}(\mathrm{s},\mathrm{t}) = i(t) - P_L i(t_L) - P_R i(t_R) \quad (5)$$

Yang mana i(tL) adalah gini index node kiri dan i(tR) adalah gini index untuk node kanan.

$$i(t_L) = 1 - \sum_{j=1}^{j} {P_{j|L}}^2 \quad (6)$$

$$i(t_R) = 1 - \sum_{j=1}^{j} {P_{j|R}}^2 \quad (7)$$

Kemudian untuk PL dan PR adalah probabilitas setiap cabang. Yang mana nilai NL adalah banyaknya variabel pada cabang kiri, nilai NR adalah banyaknya variabel pada cabang kanan, dan N adalah total data uji.

$$P_L = \frac{N_L}{N} \quad (8)$$

$$P_R = \frac{N_R}{N} \quad (9)$$

Nilai $\Delta i\ (s,\ t)$ akan maksimum apabila nilai probabilitas node kanan dan kiri memiliki keragaman lebih kecil (homogen) jika dibandingkan dengan *parent* node. Saat nilai $\Delta i\ (s,\ t)$ maksimum, maka nilai tersebut merupakan *best of split* yang sesuai dijadikan *root* node[12].

c. Pelabelan kelas

Pelabelan kelas merupakan pengidentifikasian tiap nodes pada suatu kelas tertentu. Pelabelan tidak terjadi hanya pada *root* node saja, nonterminal node dan *root* node juga mengalami proses ini. Hal ini terjadi karena setiap iterasi yang ada setiap node memiliki potensi sebagai terminal node, dan proses tersebut akan terus berlanjut hingga proses *splitting* selesai[12].

Proses pelabelan ini dilihat dari *missclasification cost* dari setiap node. Proses ini mengenal aturan dimana $C(i|j) > 0$ jika $i \neq j$ dan $C(i|j) = 0$ jika $i=j$, hal ini mengacu pada probabilitas *missclassification* yang dialami setiap node yang sering disebut Resubstitution estimate (Rt).

$$\mathrm{R(t)} \geq \mathrm{R}(t_R) + \mathrm{R}(t_L) \quad (10)$$

Data *Imbalance*

Data *Imbalance* / data tidak seimbang merupakan kondisi dimana suatu kelompok kelas memiliki jumlah data yang jauh berbeda dibandingkan dengan kelas lainnya. Kelas yang memiliki jumlah data lebih banyak sering kita sebut dengan *majority class* dan kelas yang mempunyai jumlah data lebih sedikit disebut dengan *minority class*[15].

Karakteristik dari data *imbalance* tentu dapat mempengaruhi terhadap hasil prediksi yang dilakukan oleh algoritma. Untuk mengetahui seberapa besar tingkat ketidak seimbangan data yang ada dapat dihitung menggunakan IR (*Imbalanced Ratio*) dengan perbandingan sebagai berikut.

$$Imbalanced\ Ratio\ (IR) = \frac{n_{majority}}{n_{minority}} \quad (11)$$

Perbandingan diatas menunjukkan besarnya tingkat ketidakseimbangan data berdasarkan perbandingan kelas major dan kelas minor. Metode yang dapat dilakukan untuk mengatasi permasalahan terkait data tidak seimbang (*imbalanced data*) dapat dibagi menjadi tiga bagian antara lain.

SMOTE

SMOTE merupakan metode untuk menangani kasus ketidakseimbangan data (data *imbalance*) yang diajukan oleh Chawla, dkk. Algoritma ini menggunakan pendekatan oversampling terhadap kelas minoritas yang kemudian membuat sebuah data sintesis berdasarkan nilai k-neighbor. Jumlah k-neighbor ditentukan berdasarkan pertimbangan kemudahan dalam melaksanakannya[16]. Pembangkitan data berskala numerik berbeda dengan data berskala kategorik. Data numerik mengukur jarak kedekatannya dengan jarak Euclidean sedangkan data kategorik hanya menggunakan nilai modus. Untuk mengukur jarak Euclidean dari setiap vektor dapat menggunakan rumus berikut.

$$D(x, y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \quad (12)$$

Dengan:

D : jarak antara titik x dan y
x dan y : nilai atribut
n : dimensi atribut

Untuk mengukur jarak antara kelas minor yang peubahnya berskala kategorik dilakukan dengan rumus Value Difference Metric (VDM) sebagai berikut.

$$\Delta\,(X,Y)\;=\;w_x w_y\;\sum_{i=1}^{N}\delta\,(x_i\,,y_i)^r \qquad (13)$$

Dengan:

Δ (X,Y)	: Jarak antara amatan X dengan Y
wxwy	: bobot amatan (dapat diabaikan)
N	: banyaknya peubah penjelas
R	: bernilai 1 (jarak Manhattan) atau 2 (jarak Euclidean)
$\delta\,(x_i\,,y_i)^r$	: jarak antar kategori

Untuk mendapatkan nilai jarak kategori dapat di cari menggunakan rumus.

$$\delta\,(V_1\,,V_2)\;=\;\sum_{i=1}^{n}\left|\frac{C_{1i}}{C_1}-\frac{C_{2i}}{C_2}\right|^k \qquad (14)$$

Dengan:

$\delta\,(V_1\,,V_2)$	: jarak antara nilai V1 dan V2
C_{1i}	: banyaknya V1 yang termasuk kelas i
C_{2i}	: banyaknya V2 yang termasuk kelas i
I	: banyaknya kelas; i = 1,2,...,m
C1	: banyaknya nilai 1 terjadi
C2	: banyaknya nilai 2 terjadi
N	: banyaknya kategori
K	: konstanta (biasanya 1)

Prosedur pembangkitan data untuk :

a. Data Numerik
   Hitung perbedaan antar vektor utama dengan k-tetangga terdekatnya.
   Kalikan perbedaan dengan angka yang diacak diantara 0 dan 1.
   Tambahkan perbedaan tersebut ke dalam nilai utama pada vektor utama asal sehingga diperoleh vektor utama baru[15].

$$s.feat = \big(D(x,y)\times rand(0{,}1)\big) + x_{awal} \qquad (15)$$

b. Data Kategorik
   Pilih mayoritas antara vektor utama yang dipertimbangkan dengan k-tetangga terdekatnya untuk nilai nominal. Jika terjadi nilai samamaka pilih secara acak.
   Jadikan nilai tersebut data contoh kelas buatan baru.

Confusion Matrix

Metode *confusion matrix* ini dapat menganalisa dengan baik kualitas *classifier* dalam mengenali tuple-tuple dari kelas yang ada[17]. Dalam metode ini terdapat istilah TP (*True Positives*), TN (*True Negatives*), FP (*False Positives*), FN (*False Negatives*), P adalah jumlah TP + FN, sedangkan N adalah jumlah FP + TN. TP mempunyai makna bahwa *tuple* dikenali *classifier* sebagai *tuple positives* dan TN bernilai *tuple negative*. Sebaliknya FP dan FN menyatakan bahwa *classifier* salah dalam mengenali *tuple* yang berasumsi bahwa *tuple* negatif dilabeli positif dan *tuple* positif dikenali sebagai *tuple negative*. Dalam melakukan evaluasi terhadap algoritma terkait dengan akurasi dengan rumus sebagai berikut.

$$\frac{TP+TN}{P+N} \qquad (16)$$

Kemudian untuk mengukur tingkat *error rate* atau tingkat kesalahan dari sebuah klasifikasi dapat menggunakan rumus sebagai berikut.

$$\frac{FP+FN}{P+N} \qquad (17)$$

Setelah itu kita juga dapat mengukur tingkat recall atau sensitivity dari hasil algoritma tersebut dengan rumus.

$$\frac{TP}{P} \quad (18)$$

Untuk mengukur tingkat *specificity* atau *true negative rate* dapat menggunakan rumus sebagai berikut.

$$\frac{TN}{N} \quad (19)$$

Selanjutnya adalah pengukuran tingkat presisi dari suatu algoritma dapat menggunakan rumus sebagai berikut.

$$\frac{TP}{TP+FP} \quad (20)$$

Kemudian adalah pengukuran F-Score menggunakan rumus sebagai berikut.

$$\frac{2\ x\ Precision\ x\ recall}{precision+recall} \quad (21)$$

Lalu untuk menentukan nilai AUC yang mana nilai tersebut merupakan luas daerah dibawah kurva ROC diukur menggunakan luas trapesium. Koordinat untuk nilai ROC didapat dari nilai *false positive rate* (1-specificity) untuk sumbu x dan *true positive rate* (*recall*) untuk sumbu y. Dengan mengetahui nilai yang dihasilkan dari AUC berkisar antara 0 hingga 1, yang mana nilai AUC akan baik jika semakin mendekati 1 dan buruk jika semakin mendekati 0. Supaya mempermudah dalam melihat aspek TP, TN, FP, FN, P, N maka dapat direalisasikan dalam sebuah matrix sebagai berikut.

Tabel 1. Tabel Confusion Matrix

	Ya	Tidak
Ya	TP	FN
Tidak	FP	TN
Jumlah	P	N

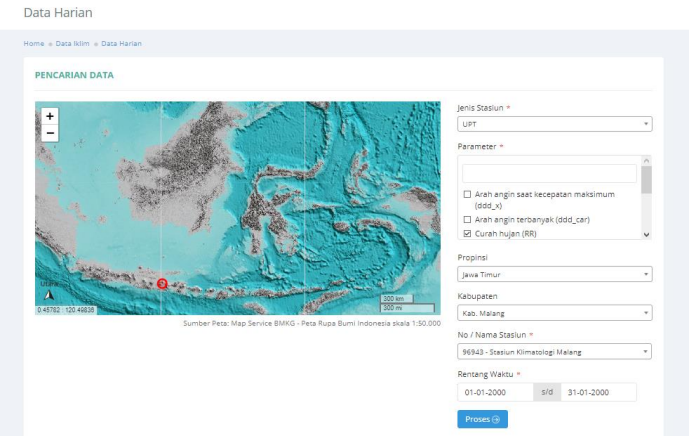
HASIL DAN PEMBAHASAN

Pengumpulan Data

Tahap ini merupakan tahapan dimana data diambil dari sumber data yaitu website dataonline.bmkg.go.id serta publikasi dari website jatim.bps.go.id yang dimana data yang diambil dari website BMKG adalah data curah hujan untuk wilayah Jawa Timur. Berikut merupakan gambar proses dan hasil dari pengambilan data curah hujan dari BMKG.

Proses pengambilan data tersebut dapat dilakukan dengan cara mengunjungi website data online BMKG, kemudian masuk ke halaman data iklim harian. Setelah itu dapat mengisi form dengan parameter curah hujan dan wilayah yang dipilih adalah Jawa Timur. Lalu rentan waktu yang dapat dipilih maksimal yang dapat diambil dalam satu sesi pengambilan adalah satu bulan. Untuk stasiun yang dipilih pada penelitian ini adalah beberapa stasiun yang memiliki data yang mendekati lengkap untuk tahun 2000 hingga 2016, maka terpilihlah beberapa stasiun di kawasan Jawa Timur, antara lain Stasiun Geofisika Tretes, Stasiun Klimatologi Malang, Stasiun Meteorologi Banyuwangi, Stasiun Meteorologi Kalianget, dan Stasiun Meteorologi Sangkapura. Kemudian data berikutnya ialah data yang diambil dari publikasi BPS yang berada di dalam website jatim.bps.go.id. data tersebut berisikan beberapa data terkait produksi padi dan palawija,

serta luas panen pertanian dari beberapa sektor dari tahun 2000 hingga 2016. Kemudian, dikarenakan data yang peneliti gunakan dimulai dari tahun 2000 maka untuk melengkapi data tersebut diambil dari publikasi BPS Jawa Timur tahun 2005 terkait luas panen dan produksi padi



Gambar 3. Pengumpulan data curah hujan

Data *Cleaning*

Proses ini merupakan tahap dimana data yang telah didapat dari sumber data, kita lakukan pembersihan dari *missing* data dan melakukan penyesuaian dengan data yang akan diuji. Pada tahap pembersihan *missing* data, yang kita lakukan adalah menggunakan teknik *binning*, yaitu mengambil nilai rata-rata dari sekumpulan data yang ada kemudian dimasukkan kedalam *missing* data tersebut. Salah satu data yang kita lakukan *cleaning* adalah data curah hujan stasiun Meteorologi Banyuwangi. Berikut gambar proses melakukan *binning* data. Setelah proses *binning* selesai, maka langkah berikutnya adalah proses penjumlahan data dan penggabungan data kedalam format tahunan, karena data curah hujan yang dilakukan *cleaning* masih dalam format bulanan.

Integrasi Data

Integrasi data merupakan proses yang dilakukan dengan tujuan untuk menggabungkan data curah hujan yang sudah dilakukan *cleaning* dan di susun menjadi format tahunan, kemudian dilakukan penggabungan dengan data dengan luas panen dan jumlah produksi yang di dapat dari publikasi BPS Jawa Timur. Setelah data terkumpul menjadi satu kesatuan yang disimpan dalam file excel, langkah selanjutnya adalah melakukan *labeling* data target berdasarkan jumlah produksi tiap tahun, dengan label naik dan turun yang menandakan kenaikan atau penurunan jumlah produksi tiap tahunnya.

***Balancing* Data**

Proses ini merupakan pengecekan terhadap data yang akan digunakan, apakah data tersebut sudah *balance* atau belum, karena berdasarkan penelitian sebelumnya hal tersebut mempengaruhi terhadap proses data mining yang akan dilakukan. Dalam kasus ini pengecekan yang dilakukan adalah dengan melihat perbandingan kelas target menggunakan rumus IR (*Imbalance Ratio*).

$$IR = \frac{Majority\ Class}{Minority\ Class} = \frac{NAIK}{TURUN} = \frac{11}{5}$$

Jika dilihat dari hasil perbandingan tersebut, nampak terlihat bahwa dataset yang digunakan tidak pada kondisi *balance* (seimbang). Langkah selanjutnya adalah membuat data tersebut menjadi *balance* menggunakan teknik *oversampling* SMOTE. Melihat data yang kita miliki merupakan data numerik maka langkah pertama yang dilakukan adalah mengukur jarak *Euclidence* dari titik *minority class* yang nantinya akan digunakan untuk membangkitkan data sintetis. Berikut adalah contoh salah satu perhitungan guna membangkitkan data numerik.

$$D(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

Data kelas minor yang digunakan adalah data tahun 2005 dengan 2001, berikut adalah pembangkitan data untuk variabel luas panen.

$$D(x,y) = \sqrt{\sum_{k=1}^{n}(1594188 - 1708478)^2} = 114290$$
$$s.feat = \big(D(x,y) \times rand(0,1)\big) + x_{awal} = (114290 \times 0.67946) + 1594188 = 1671843$$

Selanjutnya adalah pembangkitan data untuk jumlah curah hujan menggunakan variabel dari tahun 2005 dan 2016,

$$D(x,y) = \sqrt{\sum_{k=1}^{n}(2185.42 - 2647.72)^2} = 462.3$$
$$s.feat = \big(D(x,y) \times rand(0,1)\big) + x_{awal} = (462.3 \times 0.475993) + 2185.42 = 2405.47156$$

dan yang ke-tiga adalah data produksi padi yang menggunakan contoh variabel dari tahun 2001 dan 2005,

$$D(x,y) = \sqrt{\sum_{k=1}^{n}(8672791 - 9007265)^2} = 334474$$
$$s.feat = \big(D(x,y) \times rand(0,1)\big) + x_{awal} = (334474 \times 0.320535) + 8672791 = 8780002$$

dan karena data target yang kita miliki adalah *binary class*, maka langkah yang diambil adalah mengambil nilai modus dari vektor utama dan vektor k-tetangga yang menghasilkan kelas baru yakni TURUN. Proses diatas akan terus dilakukan hingga *minority class* mempunyai data yang seimbang dengan *majority class*.

Untuk melakukan implementasi dalam sistem kita dapat menggunakan bantuan dari *library imblearn* untuk mengimplementasikan hal tersebut dengan melakukan pengaturan berupa mengatur nilai k yang digunakan.

Implementasi Algoritma

Naive Bayes

Dataset yang sudah *balance* kemudian diambil dan diolah menggunakan naïve bayes agar algoritma tersebut mengenali pola yang berasal dari data *training*. Kemudian naïve bayes dapat melakukan prediksi berdasarkan pola tersebut, berikut skema prediksi yang dilakukan algoritma naïve bayes. Pertama algoritma ini akan melakukan pengurutan data secara *ascending* (kecil ke besar), kemudian algoritma tersebut akan mengelompokkan data berdasarkan kelas targetnya untuk menghitung nilai rata-rata C_1, C_2 dan standar deviasi C_1, C_2. Berikut adalah tabel pengurutan data berdasarkan kelas targetnya.

Tabel 2. Tabel Pengurutan NAIK

	Luas panen	Curah hujan	Produksi padi	Jumlah
	1787354	1994,12	11259085	Naik
	1897816	1393,8	12049342	Naik
	2112563	2647.72	12995445	Naik
	1818239	1972,5	10753206	Naik
Rata-rata C1	1903993	2002,035	11764269,5	
Standar deviasi C1	146626,5154	512,4431344	978863,3242	

Lalu untuk langkas selanjutnya adalah mengurutkan kelas turun dan menghitung nilai rata-rata dan standar deviasinya.

Tabel 3. Tabel Pengurutan TURUN

	Luas panen	Curah hujan	Produksi padi	Jumlah
	1597767	1865,08	8803878	Turun
	1595392	2332,36	9002025	Turun
	1594188	2185,42	9007265	Turun
	1934293	3066,12	12397049	Turun
	2047894	2270,12	12710538	Turun
Rata-rata C1	1753906,8	2343,82	10384151	
Standar deviasi C1	220218,2605	442,0302193	1985395,737	

Kemudian naïve bayes akan mengukur probabilitas dari kelas target yang ada akan diprediksi. Data yang digunakan sebagai sample adalah seperti Tabel 4.

Table 4. data sampel yang digunakan

2112563	2647,72	12995445	Naik

Yang kemudian dihitung menggunakan prediksi naïve bayes.

$$P(x_k|C_i) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x_k-\mu_{ik})^2}{2\sigma_{ik}^2}} = \frac{1}{standar\ deviasi\sqrt{2\pi}} e^{-\frac{(nilai\ sample-mean)^2}{2(standar\ deviasi)^2}}$$

$$P(x|jumlah = NAIK) = P(jumlah = NAIK) \frac{1}{146626.5154\sqrt{2\pi}} e^{-\frac{(2112563-1903993)^2}{2(146626.5154)^2}} \text{ X } \frac{1}{512.4431344\sqrt{2\pi}} e^{-\frac{(2647.72-2002.035)^2}{2(512.4431344)^2}} \text{ X } \frac{1}{978863.3242\sqrt{2\pi}} e^{-\frac{(12995445-11764269.5)^2}{2(978863.3242)^2}} =$$

$\frac{4}{9}$ x 0.00000098929 x 0.00035197703 x 0.00000017694 = 2.738303e-23

$$P(x|jumlah = TURUN) = P(jumlah = TURUN) \frac{1}{220218.2605\sqrt{2\pi}} e^{-\frac{(2112563-1753906.8)^2}{2(220218.2605)^2}} \text{ X } \frac{1}{442.0302193\sqrt{2\pi}} e^{-\frac{(2647.72-2343.82)^2}{2(442.0302193)^2}} \text{ X } \frac{1}{1985395.737\sqrt{2\pi}} e^{-\frac{(12995445-10384151)^2}{2(1985395.737)^2}} =$$

$\frac{5}{9}$ x 0.0000004809 x 0.00071255641 x 0.00000008461 = 1.610832e-23

Dilihat dari hasil probabilitas diatas maka hasil prediksi untuk data percobaan diatas adalah naik karena nilai probabilitas naik lebih tinggi dibanding turun.

CART

Dataset yang telah dilakukan proses *splitting* serta pengecekan *balancing* data selanjutnya akan digunakan untuk membuat model klasifikasi CART. Langkah selanjutnya adalah menentukan *goodness of split* menggunakan rumus *impurity* yaitu $\Delta i(s^*,t)$ adalah nilai yang paling maksimal. Untuk menentukan nilai *impurity* tersebut langkah yang perlu dilakukan adalah membuat split model untuk mengetahui batasan dari setiap variabel, karena variabel yang ada bersifat numerik. Cara untuk menghitung batas setiap variabel numerik adalah dengan menggunakan nilai median dari setiap variabel yang berurutan. Setelah mendapatkan nilai *split* dari setiap variabel, langkah berikutnya adalah mencari nilai *goodness of split* berdasarkan nilai *gini index* dari setiap *split value* tersebut.

Tabel 4. Nilai Improvement

No	Nama Kolom	Variabel *Split*	*Improvement*
1	Luas Panen (Ha)	x ≤ 1594790, x > 1594790	0.049382556
		x ≤ 1596579.5, x > 1596579.5	0.112874619
		x ≤ 1692560.5, x > 1692560.5	0.197530704
		x ≤ 1802796.5, x > 1802796.5	0.060493667

		x ≤ 1858027.5, x > 1858027.5	0.004938111
		x ≤ 1916054.5, x > 1916054.5	0.012345519
		x ≤ 1991093.5, x > 1991093.5	0.001763508
		x ≤ 2080228.5, x > 2080228.5	-0.021948034
2	Jumlah Curah Hujan (milimeter)	x ≤ 1629.44, x >1629.44	0.077160333
		x ≤ 1918.79, x >1918.79	0.001763508
		x ≤ 1983.31, x >1983.31	0.049382556
		x ≤ 2089.77, x >2089.77	0.149382556
		x ≤ 2227.77, x >2227.77	0.060493667
		x ≤ 2301.24, x >2301.24	0.012345519
		x ≤ 2490.04, x >2490.04	0.001763508
		x ≤ 2856.92, x >2856.92	0.049382556
3	Produksi Padi (ton)	x ≤ 8902951.5, x >8902951.5	0.049382556
		x ≤ 9004645, x >9004645	0.112874619
		x ≤ 9880235.5, x >9880235.5	0.197530704
		x ≤ 11006145.5, x >11006145.5	0.060493667
		x ≤ 11654213.5, x >11654213.5	0.004938111
		x ≤ 12223195.5, x >12223195.5	0.012345519
		x ≤ 12553793.5, x >12553793.5	0.001763508
		x ≤ 12852991.5, x >12852991.5	0.077160333

Dimana untuk mencari nilai *root* node adalah dengan memilih nilai *improvement* yang terbesar disebabkan memiliki nilai kehomogenan yang baik. Sebagai contoh perhitungan yang peneliti ambil adalah pada kolom produksi padi. Langkah pertama yang dilakukan agar memudahkan perhitungan adalah dengan mengurutkan nilai produksi padi mulai dari terkecil hingga terbesar dengan kelas target yang tetap mengikuti nilai produksi. Setelah itu kita dapat menghitung nilai gini berdasarkan variabel split yang sudah ditentukan, sebagai contoh peneliti mengambil variabel *split* x ≤ 9880235.5 dan x >9880235.5. Berikut adalah penjelasan terkait mencari nilai gini index.

$$i(t) = 1 - \sum_j P^2(j|t) = 1 - \left(\frac{4}{9}\right)^2 - \left(\frac{5}{9}\right)^2 = 0.49382716$$

$$i(t_L) = i(\text{x} \leq 9880235.5) = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$i(t_R) = i(\text{x} > 9880235.5) = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 0.44444$$

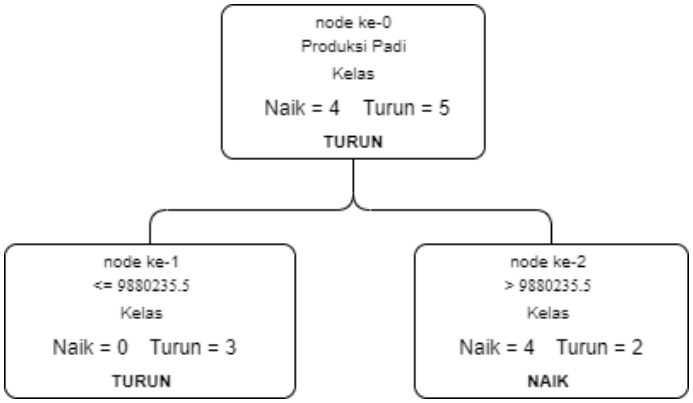
$$\Delta\text{i(s,t)} = i(t) - P_L i(t_L) - P_R i(t_R) = 0.494 - \left(\frac{3}{9} \times 0\right) - \left(\frac{6}{9} \times 0.44\right) = 0.198$$

Dengan demikian, *best split of goodness* adalah produksi padi dengan *root node* x ≤ 9880235.5, x >9880235.5. Jika dilihat dari pengujian diatas ada beberapa yang memiliki nilai *improvement* yang sama dengan x ≤ 9880235.5, x >9880235.5, maka dipilih salah satu secara acak untuk dijadikan *root node*. Setelah *root node* terpenuhi, langkah berikutnya adalah melakukan pelabelan, dengan melihat prosentasi kelas naik dan turun. Berdasarkan *class assignment rule* yaitu jika $p(j|t) = \max_i (pi|t)\ maka\ j^*(t)=j$, dimana $j^*(t)$ adalah kelas yang diidentifikasikan pada node t. Sebagai contoh

$$P(NAIK\ |t) = \frac{4}{9} = 0.44$$

$$P(TURUN\ |t) = \frac{5}{9} = 0.55$$

Maka melihat peluang dari label naik ataupun turun dari perhitungan tersebut maka untuk *root node* memiliki label turun.



Gambar 4. Pemecahan root node

Langkah rekursif terkait pembentukan *node* tersebut akan terus berlangsung hingga *node* terakhir.

Analisis Hasil Prediksi

Tahapan analisis terhadap akurasi dari dua model klasifikasi tersebut dilakukan dengan melakukan percobaan prediksi sebanyak 15 iterasi sehingga menghasilkan rata-rata tingkat akurasi dari algoritma naïve bayes sebelum dilakukan *oversampling* adalah 43,8%, *recall* sebesar 0.654, *specifisitas* sebesar 0.056, f-score sebesar 0.572, AUC sebesar 0.373, *presisi* sebesar 0.607 dan setelah dilakukan *oversampling* nilai akurasinya menjadi sebesar 39,06%, *recall* sebesar 0.446, *specifisitas* sebesar 0.449, f-score sebesar 0.396, AUC sebesar 0.475, presisi sebesar 0.507. Sedangkan untuk algoritma CART mendapatkan tingkat akurasi sebelum dilakukan *oversampling* sebanyak 47,67%, *recall* sebesar 0.055, *specifisitas* sebesar 0.67, f-score sebesar 0.627, AUC sebesar 0.390, presisi sebesar 0.044 dan setelah dilakukan *oversampling* nilai akurasinya menjadi sebesar 55,73%, recall sebesar 0.547, specifisitas sebesar 0.651, f-score sebesar 0.518, AUC sebesar 0.491, presisi sebesar 0.594. Hal tersebut menunjukkan adanya pengaruh hasil *oversampling* terhadap data dengan tingkat akurasi algoritma, walaupun dalam kasus ini algoritma naïve bayes relatif sedikit turun antara sebelum dilakukan *oversampling* dan setelahnya.

KESIMPULAN

Berdasarkan hasil dari pengujian yang telah dilakukan terkait menganalisis tingkat akurasi dari algoritma naïve bayes dan CART dalam memprediksi produksi padi dapat ditarik kesimpulan bahwa dalam proses melakukan prediksi, peneliti menemukan masalah terkait dataset yang tidak seimbang sehingga dalam proses memprediksi, peneliti mengukur pengaruh data *imbalance* terhadap performa akurasi dari setiap algoritma. Dalam percobaan pembangkitan data sintetik menggunakan algoritma SMOTE, peneliti berhasil melakukan *balancing* data terhadap data awal yang memiliki perbandingan kelas minor dan mayor adalah 5 berbanding 11 menjadi 11 berbanding 11. Kemudian data tersebut dilihat pengaruhnya terhadap algoritma yang diuji yaitu naïve bayes dengan menggunakan data sebelum *balancing* menghasilkan akurasi sebesar 43,8% dengan nilai AUC sebesar 0.373 lalu setelah dilakukan *oversampling* berubah nilai akurasinya menjadi 39.06% dengan nilai AUC meningkat menjadi 0.475. Untuk pengujian yang kedua yakni terhadap algoritma CART dengan menggunakan data sebelum *balancing* menghasilkan nilai akurasi 47.67% dengan nilai AUC sebesar 0.391, kemudian dilakukan *balancing* data yang menghasilkan nilai akurasi meningkat menjadi 55.73% dengan nilai AUC sebesar 0.492. Hal ini menunjukan adanya pengaruh data *imbalance* terhadap hasil prediksi yang dihasilkan dengan meningkatnya parameter nilai AUC mengindikasikan bahwa data atau model yang digunakan semakin baik digunakan dalam proses klasifikasi.

DAFTAR PUSTAKA

[1] BPS, *Hasil Survey Pertanian Antar Sensus (SUTAS) 2018 Seri-A2*, Seri-A2. Jakarta: BPS-Statistics, 2019.

[2] Suyanto, *Data Mining Untuk Klasifikasi dan Klasterisasi Data*. Bandung: Informatika Bandung, 2017.

[3] F. H. Utami, "Penentuan Tingkat Kesuburan Tanah di Balai Penyuluhan Pertanian Perikanan dan Kehutanan," *Riau J. Comput. Sci.*, vol. 1, no. 1, pp. 27–39, 2015, [Online]. Available: http://e-journal.upp.ac.id/index.php/RJOCS/article/view/485.

[4] I. N. Hidayati and S. Suryanto, "Pengaruh Perubahan Iklim Terhadap Produksi Pertanian Dan Strategi Adaptasi Pada Lahan Rawan Kekeringan," *J. Ekon. Stud. Pembangunan.*, vol. 16, no. 1, pp. 42–52, 2015, doi: 10.18196/jesp.16.1.1217.

[5] B. Landia, "Peramalan Jumlah Mahasiswa Baru Dengan Exponential Smoothing dan Moving Average," *J. Ilm. Intech Informatioan Technol. J. UMUS*, vol. 2, no. 1, pp. 71–78, 2020, doi: 10.46772/intech.v2i01.188.

[6] Ronida and Kosim, "Implementasi Prototype Dalam Pembuatan Website Sebagai Media Promosi Di MA Darul Masholeh Cirebon," *J. Ilm. Intech Informatioan Technol. J. UMUS*, vol. 1, no. 2, pp. 33–42, 2019, doi: 10.46772/intech.v1i02.68.

[7] R. A. Putra, "Penerapan Naïve Bayes Classifier dengan Gaussian Function Untuk Menentukan Kelompok UKT," *J. Inform. Glob.*, vol. 9, no. 2, pp. 112–117, 2018, doi: 10.36982/jig.v9i2.583.

[8] E. Zhang, B. Li, P. Li, and Y. Chen, "A deep learning based printing defect classification method with imbalanced samples," *Symmetry (Basel).*, vol. 11, no. 12, pp. 1–14, 2019, doi: 10.3390/SYM11121440.

[9] S. H. Sumartini and S. W. Purnami, "Penggunaan Metode Classification and Regression Trees (CART) untuk Klasifikasi Rekurensi Pasien Kanker Serviks di RSUD Dr. Soetomo Surabaya," *J. Sains dan Seni ITS*, vol. 4, no. 2, pp. 211–216, 2015, doi: 10.12962/j23373520.v4i2.10673

[10] Hariati, M. Wati, and B. Cahyono, "Penerapan Algoritma C4.5 Decision Tree pada Penentuan Penerima Program Bantuan Pemerintah Daerah Kabupaten Kutai Kartanegara," *Jurti*, vol. 2, no. 1, pp. 27–36, 2018, doi: 10.30872/jurti.v2i2.1861.

[11] A. Walyuno, moch abdul Mukid, and T. Wuryandari, "Perbandingan Analisis Klasifikasi Nasabah Kredit Menggunakan Regresi Logistik Biner Dan Cart (Classification and Regression Trees)," *J. Gaussian*, vol. 4, no. 2, pp. 215–225, 2015, doi: 10.15797/concom.2019..23.009.

[12] W. Y. Loh, "Classification and regression trees," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, 2011, doi: 10.1002/widm.8.

[13] F. E. Pratiwi, F. E. Pratiwi, and I. Zain, "Klasifikasi Pengangguran Terbuka Menggunakan CART (Classification and Regression Tree) di Provinsi Sulawesi Utara," *J. Sains dan Seni ITS*, vol. 3, no. 1, pp. D54–D59, 2014, doi: 10.12962/j23373520.v3i1.6129

[14] L. Breiman, "Technical note: Some properties of splitting criteria," *Mach. Learn.*, vol. 24, no. 1, pp. 41–47, 1996, doi: 10.1007/bf00117831.

[15] R. A. Barro, I. D. Sulvianti, and F. M. Afendi, "Penerapan Synthetic Minority Oversampling Technique (Smote) Terhadap Data Tidak Seimbang Pada Pembuatan Model Komposisi Jamu," *Xplore J. Stat.*, vol. 1, no. 1, pp. 1–6, 2013, doi: 10.29244/xplore.v1i1.12424.

[16] N. Rout, D. Mishra, and M. K. Mallick, "Handling imbalanced data: A survey," *Adv. Intell. Syst. Comput.*, vol. 628, pp. 431–443, 2018, doi: 10.1007/978-981-10-5272-9_39.

[17] Harliana and W. Widayani, "Analisis Dempster Shafer Pada Sistem Pakar Pendeteksi ISPA," *FAHMA*, vol. 17, no. 2, pp. 60–69, 2019, [Online]. Available: https://stmikelrahma.e-journal.id/FAHMA/article/view/34/22.