SKRIPSI

Klasifikasi *Website Phishing* Menggunakan Algoritma *Random Forest* dengan Teknik *Random Oversampling*

Classification of Phishing Websites Using Algorithms Random Forest with Random Oversampling Technique

YUSRINA

D 0220367

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS SULAWESI BARAT

MAJENE

2025

HALAMAN PERSETUJUAN

SKRIPSI

KLASIFIKASI WEBSITE PHISHING MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN TEKNIK RANDOM OVERSAMPLING

Telah dipersiapkan dan disusun oleh

YUSRINA

D 0220367

Telah dipertanggung jawabkan didepan Tim Penguji

Pada tanggal 12 Desember 2024

Susunan Tim Penguji

Pembimbing I	Penguji I
Nurhikma Arifin, S.Kom., M.T	**Ir. Sugiarto Cokrowibowo, S.Si., M.T.**
NIP: 199304252022032011	**NIP: 198605242015041004**
Pembimbing II	Penguji II
A. Amirul Asnan Cirua, S.T.,M. Kom	**Farid Wajidi, S. Kom., MT.**
NIP: 199804022024061001	**NIP: 198904182019031018**
	Penguji III
	Heliawati Hamrul, S. Kom., M. Kom
	NIP: 198710152019032008

LEMBAR PENGESAHAN

SKRIPSI

KLASIFIKASI WEBSITE PHISHING MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN TEKNIK RANDOM OVERSAMPLING

Disusun dan diajukan oleh:

YUSRINA

NIM. D0220367

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Teknik Informatika Fakultas Teknik Universitas Sulawesi Barat
pada tanggal 12 Desember 2024
dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing I	Pembimbing II
Nurhikma Arifin, S.Kom., M.T	**A. Amirul Asnan Cirua, S.T., M.Kom**
NIP.199304252022032011	**NIP.199804022024061001**

Dekan Fakultas Teknik, Universitas Sulawesi Barat	Ketua Program Studi Informatika,
Dr. Ir. Hafsah Nirwana, M.T	**Muh Rafli Rasyid, S.Kom., M.T**
NIP. 196404051990032002	**NIP. 198808182022031006**

iii

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsurunsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang undangan yang berlaku **(UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70)**.

Majene, 12 Desember 2024

Yusrina

NIM. D0220371

ABSTRAK

Yusrina. Klasifikasi *Website Phishing* Menggunakan Algoritma *Random Forest* dengan Teknik *Random Oversampling*. (Dibimbing oleh **Nurhikma Arifin** dan **Andi Amirul Asnan Cirua**)

Klasifikasi *phishing* dan *Non-Phishing* merupakan tantangan dalam keamanan siber, terutama dalam menangani ketidakseimbangan kelas. Penelitian ini mengevaluasi performa *model Random Forest* dengan dan tanpa teknik *Oversampling*, yaitu *Random Oversampling* dan *Synthetic Minority Over-sampling Technique* (SMOTE). Tanpa *Oversampling*, *model* mencapai akurasi 95.93% dengan *precision* 93.65% untuk *Non-Phishing* dan 95.11% untuk *phishing*, *Recall* 97.03% untuk *Non-Phishing* dan 97.73% untuk *phishing*, serta *F1-score* masing-masing 95.31% dan 96.41%. Penerapan *Random Oversampling* meningkatkan akurasi menjadi 96.16%, *precision* menjadi 96.36% untuk *Non-Phishing* dan 96.00% untuk *phishing*, serta *F1-score* yang lebih baik. Namun, *Recall* kelas *Non-Phishing* dan *phishing* menurun menjadi 94.88% dan 97.17%, menunjukkan potensi *overfitting* pada kelas *minoritas*. Sementara itu, SMOTE menghasilkan akurasi 95.62% dengan *precision* lebih tinggi untuk *Non-Phishing*, yaitu 96.89%, tetapi *Recall* menurun menjadi 94.41% untuk *Non-Phishing* dan 96.86% untuk *phishing*. *Precision* kelas *phishing* juga menurun menjadi 94.37%, sedangkan *F1-score* untuk kelas *Non-Phishing* dan *phishing* masing-masing adalah 95.63% dan 95.60%. Hasil penelitian menunjukkan bahwa *model Random Forest* sudah cukup andal tanpa *Oversampling*, sementara penggunaan teknik *Oversampling* harus dipertimbangkan dengan hati-hati untuk menjaga keseimbangan klasifikasi.

Kata kunci: *Random Forest*, *Random Oversampling*, *SMOTE*, ketidakseimbangan kelas, klasifikasi *phishing*.

ABSTRACT

Yusrina. *Classification of Phishing Websites Using Algorithms Random Forest with Random Oversampling Technique.* (Dibimbing oleh **Nurhikma Arifin** dan **Andi Amirul Asnan Cirua**)

Phishing and Non-Phishing classification is a challenge in cybersecurity, particularly in handling class imbalance. This study evaluates the performance of the Random Forest model with and without Oversampling techniques, namely Random Oversampling and the Synthetic Minority Over-sampling Technique (SMOTE). Without Oversampling, the model achieves an accuracy of 95.93%, with a precision of 93.65% for Non-Phishing and 95.11% for phishing, Recall of 97.03% for Non-Phishing and 97.73% for phishing, and F1-scores of 95.31% and 96.41%, respectively. The application of Random Oversampling increases accuracy to 96.16%, precision to 96.36% for Non-Phishing and 96.00% for phishing, and results in improved F1-scores. However, the Recall for Non-Phishing and phishing decreases to 94.88% and 97.17%, indicating potential overfitting in the minority class. Meanwhile, SMOTE yields an accuracy of 95.62%, with a higher precision for Non-Phishing at 96.89%, but Recall decreases to 94.41% for Non-Phishing and 96.86% for phishing. The precision for the phishing class also decreases to 94.37%, while the F1-scores for Non-Phishing and phishing are 95.63% and 95.60%, respectively. The results indicate that the Random Forest model is already reliable without Oversampling, while the use of Oversampling techniques should be carefully considered to maintain classification balance.

***Keywords**: Random Forest, Random Oversampling, SMOTE, class imbalance, phishing classification.*

BAB I

PENDAHULUAN

A. Latar Belakang

Pada masa kini, perkembangan teknologi terus berkembang, menyebabkan masyarakat sulit terlepas dari penggunaan internet dan *gadget.* Kemajuan internet ini sejalan dengan perkembangan perangkat lunak yang semakin canggih (Aprelia Windarni *et al.*, 2023). Di Indonesia, sekitar 73,7% dari total penduduk sebanyak 274,9 juta jiwa, atau sekitar 202,6 juta jiwa, telah aktif menggunakan internet (Subarkah and Ikhsan, 2021). Dengan pertumbuhan teknologi internet yang cepat, diperkirakan angka ini akan terus meningkat. Namun, semakin banyaknya pengguna internet juga meningkatkan risiko terhadap keamanan data, karena datapengguna dapat rentan dicuri oleh pihak yang tidak bertanggung jawab (Fandru Al Rifqi *et al.*, 2022). Salah satu ancaman serius terhadap keamanan data di internet adalah praktik *phishing*.

Phishing merupakan kegiatan yang bersifat mengancam dan menjebak seseorang dengan cara memancing target untuk secara tidak langsung memberikan informasi kepada penjebak(Fandru Al Rifqi *et al.*, 2022). Tujuan dari *phishing* adalah membuat pengguna meyakini bahwa mereka berinteraksi dengan situs resmi. Umumnya, *phisher* (pelaku *phishing*) mencari informasi seperti *username, password*, baik untuk akun media sosial maupun akun nomor kartu kredit, dengan

mengarahkan pengguna ke situs web palsu melalui URL(Farida and Mustopa, 2023). Menurut laporan dari *Kaspersky Security Network* pada tahun 2021, serangan *phishing* telah menjadi ancaman keamanan siber utama di Indonesia. Terdapat sekitar 1,6 juta serangan *phishing* yang terdeteksi selama periode kuartal keempat 2020. Serangan *phishing* juga menjadi ancaman terbesar di Asia Tenggara secara keseluruhan, dengan Indonesia menempati posisi teratas dalam jumlah serangan *phishing*. Beberapa faktor yang menyebabkan peningkatan serangan *phishing* di Indonesia termasuk peningkatan jumlah pengguna internet, kurangnya kesadaran akan keamanan *cyber*, dan kelemahan dalam keamanan infrastruktur internet. Selain itu, banyaknya pengguna internet yang masih belum teredukasi dengan baik mengenai taktik dan strategi yang digunakan oleh pelaku *phishing* juga menjadi faktor penyebab (Ramadhan and Desyani, 2023).

Serangan *phishing* umumnya dikembangkan dan disebarluaskan melalui internet dengan menggunakan dua metode utama: email palsu dan replikasi situs web yang sah. Email palsu sering dikirimkan kepada pengguna dengan menyamar sebagai perusahaan atau organisasi resmi. Di samping itu, penyerang juga menciptakan dan menyebarkan replika situs web yang mirip dengan situs asli melalui platform media sosial seperti *Twitter, Facebook, dan Google*. Situs web *phishing* ini menggunakan protokol *Hypertext Transfer Protocol Secure* (HTTPS) untuk mengecoh pengguna agar mempercayai keaslian situs tersebut. Dalam upaya untuk mendeteksi dan mencegah serangan *phishing*, berbagai metode telah diusulkan dalam literatur, termasuk penggunaan daftar hitam, ekstraksi fitur, dan penerapan pembelajaran mesin. Daftar hitam merupakan kumpulan URL *phishing*

yang biasanya diblokir oleh *browser* modern seperti *Chrome, Opera, dan Mozilla.* Namun, metode ini kurang efektif dalam mengidentifikasi dan mencegah situs *phishing zero-day* yang memiliki umur pendek. Sementara ekstraksi fitur melibatkan pengambilan ciri-ciri khas dari situs web *phishing* untuk tujuan identifikasi dan pencegahan, namun tidak semua situs *phishing* memiliki fitur yang serupa, sehingga metode ini mungkin tidak dapat diandalkan untuk semua jenis situs web. Sebagai solusi alternatif, *model* klasifikasi *machine learning* digunakan untuk mendeteksi serangan *phishing*. Hasil penelitian yang ada menunjukkan bahwa metode berbasis pembelajaran mesin dapat mencapai tingkat akurasi yang tinggi dalam mendeteksi situs web *phishing*, melebihi kinerja teknik daftar hitam dan ekstraksi fitur (Kalabarige *et al.*, 2023). Menurut penelitian yang dilakukan oleh (Zieni *et al.*, 2023) dimana pada penelitian ini berfokus pada tiga kategori penting dalam pendekatan deteksi *website phisihing*, yaitu berbasis daftar, berbasis kesamaan, dan berbasis *machine learning* dari penelitian ini menunjukkan bahwa *machine learning* memberikan solusi yang efektif dalam deteksi *phishing*, dengan kemampuan untuk mendeteksi halaman *web phishing* baru secara efisien. Analisis ini juga mencerminkan bahwa *machine learning* merupakan bidang penelitian yang sangat dinamis. Oleh karena itu dalam penelitian ini akan menggunakan *machine learning*.

Dalam data *mining*, terdapat berbagai metode dengan fungsi dan tujuan yang berbeda, salah satunya adalah klasifikasi. Salah satu metode klasifikasi yang paling dikenal adalah pohon keputusan, yang mudah dipahami dan diinterpretasikan dengan penjelasan singkat. Pohon keputusan mampu menangani *overfitting*, atribut

kontinu, pemilihan atribut yang relevan, data pelatihan dengan nilai atribut yang hilang, dan juga dapat meningkatkan efisiensi komputasi. Namun, salah satu tantangan utama dalam klasifikasi data *mining* adalah ketidakseimbangan *dataset*. Masalah umum dalam komunitas data *mining* dan *machine learning* adalah menangani data yang tidak seimbang dan memilih fitur terbaik. Ketidakseimbangan kelas atau *dataset* terjadi ketika distribusi data tidak merata, biasanya terbagi menjadi kelas *mayoritas* dan kelas *minoritas* (Tanti, 2023).

Kelompok kelas yang memiliki jumlah data yang banyak disebut sebagai kelas *mayoritas,* sedangkan kelompok kelas dengan jumlah data yang lebih sedikit disebut kelas *minoritas*. Perbandingan antara jumlah data kelas *minoritas* dan kelas *mayoritas* dikenal sebagai *Imbalance Ratio* (IR) atau rasio ketidakseimbangan. Semakin besar perbedaan jumlah antara kelas *minoritas* dan kelas *mayoritas*, maka semakin tinggi nilai *Imbalance Ratio* (IR) atau rasio ketidakseimbangan tersebut. Ketidakseimbangan *dataset* dalam data mining merupakan masalah serius, karena ketidakseimbangan kelas (*class imbalance)* dapat menyulitkan proses pembelajaran klasifikasi (Sulistiyono *et al.*, 2021). Algoritma klasifikasi yang ada sering memberikan hasil yang kurang memuaskan pada *dataset* yang tidak seimbang, karena algoritma ini dirancang untuk *dataset* yang seimbang. Kelas *minoritas* sering kali mengalami kesalahan klasifikasi karena algoritma *machine learning* cenderung memprioritaskan kelas *mayoritas* dan mengabaikan kelas *minoritas.* Akibatnya, kelas *mayoritas* mungkin diklasifikasikan secara berlebihan karena *probabilitas* yang lebih tinggi, menyebabkan kelas *minoritas* lebih sering salah diklasifikasikan. Hal ini mempengaruhi kualitas data dan kinerja klasifikasi secara

keseluruhan, yang mengakibatkan hasil klasifikasi yang kurang optimal. Teknik data *mining* dan *machine learning* umumnya memerlukan distribusi kelas yang seimbang untuk bekerja dengan efektif. Jika ketidakseimbangan kelas sangat ekstrem, akurasi prediksi keseluruhan bisa menjadi tinggi karena *model* cenderung memprediksi sebagian besar sampel sebagai kelas *mayoritas*. Ketidakseimbangan kelas dapat menghasilkan akurasi yang jauh lebih tinggi untuk kelas *mayoritas* dibandingkan dengan kelas *minoritas* (Tanti, 2023). Beberapa teknik dapat digunakan untuk mengatasi masalah ketidakseimbangan kelas, namun dalam penelitian ini, metode yang akan digunakan adalah *Random Oversampling*.

Pada penelitian yang dilakukan oleh (Fandru Al Rifqi *et al.*, 2022) mereka membandingkan algoritma klasifikasi yaitu *regresi logistik, decision tree, dan Random Forest* dalam melakukan prediksi *website phishing* dari penelitian ini didapatkan akurasi akhir yaitu *Random Forest* memiliki akurasi tertinggi yaitu 97,10%, *decision tree* 94,57% dan *regresi logistik* 92,79%.

Pada penelitian lain yang dilakukan oleh (Diantika, 2023) dimana pada penelitian ini peneliti menggunakan algoritma *lightGBM* dengan teknik *Random Oversampling* untuk mengatasi *imbalance class* dari penelitian ini didapatkan hasil akurasi sebesar 96,9%, dimana hasil ini secara signifikan lebih baik dari pada metode lainnya. Berdasarkan hasil dari beberapa penelitian terdahulu dapat disimpulkan bahwa algoritma *Random Forest* memiliki tingkat akurasi yang cukup baik dalam mendeteksi *website phishing*.

Berdasarkan latar belakang yang telah diuraikan diatas, penulis mengangkat penelitian dengan judul "Klasifikasi *Website Phishing* Menggunakan Algoritma *Random Forest* dengan Teknik *Random Oversampling*".

B. Rumusan Masalah

1. Bagaimana hasil dari penerapan algoritma *Random Forest* dalam klasifikasi *website phishing*?
2. Bagaimana hasil perbandingan klasifikasi *website phishing* menggunakan Algoritma *Random Forest* dan Algoritma *Random Forest* dengan teknik *Random Oversampling*?

C. Tujuan Penelitian

1. Untuk membuat *model* klasifikasi *website phishing* menggunakan algoritma *Random Forest*
2. Untuk mengetahui hasil perbandingan klasifikasi *website phishing* menggunakan Algoritma *Random Forest* dan Algoritma *Random Forest* dengan Teknik *Random Oversampling.*

D. Batasan masalah

1. Data yang digunakan merupakan data sekunder *Phishing Website Detector* yang di ambil dari *kaggle.com*
2. Klasifikasi data menggunakan algoritma *Random Forest*
3. Teknik *resampling* yang digunakan yaitu teknik *Random Oversampling* untuk mengatasi kelas yang tidak seimbang
4. Evaluasi kinerja menggunakan *confusion matrix*
5. Hanya menggunakan rasio 80:20

6. Menggunakan bahasa pemprograman pyhton

E. Manfaat

1. Menghasilkan *model* klasifikasi efektif untuk deteksi *website phishing*, yang dapat digunakan sebagai alat bantu keamanan dalam menyaring *website* berpotensi berbahaya.
2. Bagi para peneliti, hal ini bisa berfungsi sebagai tambahan pengetahuan dan sebagai sumber referensi mengenai penerapan *machine learning* dalam klasifikasi *website phishing*.
3. Untuk lembaga pendidikan, dapat menjadi sumber referensi yang bermanfaat dalam pengembangan penelitian terkait topik ini, khususnya dalam memahami lebih lanjut klasifikasi *website phishing* menggunakan pendekatan *machine learning.*

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

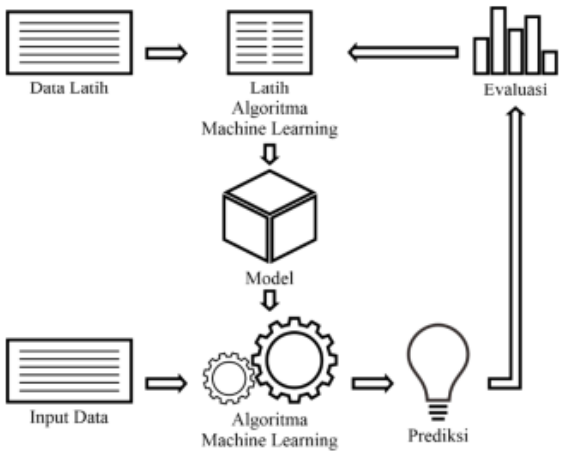
1. Phishing

Istilah *phishing* dalam bahasa Inggris berasal dari kata *fishing* yang berarti memancing. *Phishing* merupakan bentuk tindakan kriminal yang bertujuan untuk memperoleh data rahasia dari seseorang, seperti nama pengguna, kata sandi, dan informasi kartu kredit. Modus operasi ini melibatkan penyamaran sebagai individu atau bisnis tepercaya dalam komunikasi elektronik resmi. Serangan *phishing* biasanya berupa sebuah email yang seakan-akan berasal dari perusahaan resmi, misalnya dari *website-website* yang biasa digunakan oleh pengguna (Diki Wahyudi, 2020).

Menurut George W. Reynolds, *phishing* adalah "tindakan menggunakan email secara menipu untuk mencoba menipu penerima untuk mengungkapkan data pribadi". *Phishing* dilakukan dengan mengirimkan tautan yang tampak asli dari organisasi terkait kepada pengguna internet melalui email dan situs web. Ketika pengguna mengklik tautan tersebut, penyerang memperoleh informasi dari pengguna dan menggunakannya untuk keuntungan pribadi, contohnya untuk mengambil uang dari rekening pengguna atau menggunakan akun tersebut untuk pembayaran online (Prawira *et al.*, 2021).

2. Machine learning

Machine learning adalah cabang ilmu dari kecerdasan buatan yang menggunakan teknik statistika untuk membuat *model* otomatis dari kumpulan data, yang disebut sebagai *Dataset*. Tujuannya adalah memberikan kemampuan komputer untuk "belajar." Proses pembelajaran ini memungkinkan komputer untuk mengeksplorasi dan memahami data sehingga dapat membentuk *model* untuk menjalankan proses *input-output* tanpa perlu menggunakan kode program yang dibuat secara eksplisit. Proses ini menggunakan algoritma khusus yang disebut algoritma pembelajaran mesin. Ada berbagai macam algoritma pembelajaran mesin dengan tingkat efisiensi dan spesifikasi yang berbeda, sesuai dengan kebutuhan kasus tertentu. Proses pembelajaran ini tidak hanya meningkatkan kecerdasan individu, tetapi juga memungkinkan mesin untuk meningkatkan kemampuannya dan memiliki kecerdasan yang unik, tidak seperti mesin lainnya (Diki Wahyudi, 2020).



(Sumber: Diki Wahyudi, 2020)

Gambar 2. 1 Ilustrasi Proses *Machine learning*

Pada dasarnya, cara kerja *machine learning* mirip dengan pembelajaran manusia, di mana *model* belajar dari contoh-contoh untuk kemudian memberikan jawaban terhadap pertanyaan yang terkait. Proses pembelajaran ini bergantung pada penggunaan data yang disebut *Dataset*. Berbeda dengan program statis, *machine learning* dirancang untuk menciptakan program yang dapat belajar sendiri. Dari *Dataset* tersebut, komputer akan melakukan proses pembelajaran (*training*) untuk mengembangkan suatu *model*. Proses ini melibatkan penggunaan algoritma *machine learning* sebagai implementasi dari teknik statistika. *Model* yang dihasilkan dari proses ini berfungsi untuk menghasilkan informasi, yang kemudian dapat digunakan sebagai pengetahuan untuk mengatasi suatu permasalahan dalam bentuk proses *input-output*. *Model* ini mampu melakukan klasifikasi atau prediksi untuk kejadian di masa depan (Diki Wahyudi, 2020).

3. Klasifikasi

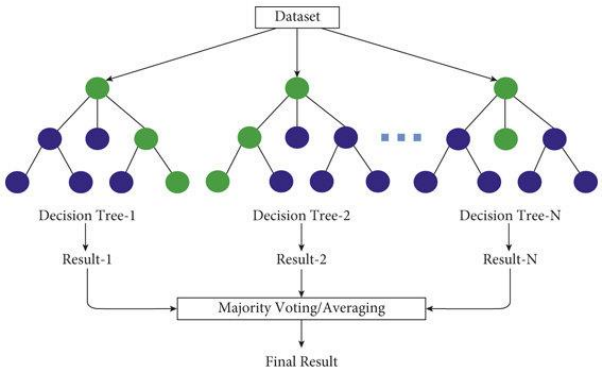
Klasifikasi adalah proses untuk menemukan *model* atau fungsi yang membedakan konsep atau kelas data dengan tujuan untuk memprediksi kelas yang tidak diketahui dari suatu objek. Dalam klasifikasi, terdapat dua tahapan, yaitu tahapan pelatihan (*training*) dan tahapan pengujian (*testing*). Tahapan pelatihan menggunakan set data pelatihan yang sudah diberi label untuk membangun *model*. Tahapan pengujian digunakan untuk menguji akurasi *model* yang telah dibangun selama tahapan pelatihan (Aziz, 2021).

4. *Random Forest*

Random Forest merupakan salah satu algoritma dalam *machine learning* yang termasuk dalam kategori *ensemble learning*. *Ensemble learning* adalah metode yang menggabungkan beberapa *model machine learning* sederhana untuk membentuk *model* yang lebih kuat. Dalam algoritma *Random Forest*, dibentuk kumpulan pohon keputusan, di mana setiap pohon dibangun menggunakan *subset* acak dari data pelatihan. Setiap pohon dalam *ensemble* melakukan prediksi secara *independen*, dan hasil prediksi dari semua pohon digabungkan untuk menghasilkan prediksi akhir (Supriadi and Andarsyah, 2023) .

Random Forest adalah algoritma yang sering digunakan untuk menyelesaikan berbagai masalah, termasuk masalah klasifikasi, regresi, dan lainnya (Aziz,2021). *Random Forest* memanfaatkan teknik *bagging* dan pemilihan atribut secara acak untuk membangun *model*nya. *Random Forest* adalah teknik yang dapat meningkatkan akurasi dengan menerapkan atribut secara acak pada setiap *node*. Algoritma ini terdiri dari beberapa pohon keputusan yang digunakan untuk mengelompokkan data ke dalam kelas-kelas tertentu. Setiap pohon keputusan memiliki *node* akar dan *node* daun yang menghasilkan hasil akhir (Suci Amaliah, Nusrang and Aswi, 2022). Algoritma *Random Forest* menggunakan vektor acak yang diambil secara acak dan merata pada semua pohon dalam *ensemble*. Hasil prediksi dari *Random Forest* diperoleh dengan mengambil suara terbanyak dari setiap pohon keputusan (*voting* untuk klasifikasi dan rata-rata untuk regresi) (Aziz, 2021).

a. Setiap pohon dalam *Random Forest* dibangun menggunakan sampel *bootstrap* yang diambil secara acak dari data pelatihan.
b. Saat memilih atribut untuk membagi *node* dalam pohon keputusan, sebagian variabel dipilih secara acak dari seluruh kumpulan data, dan kemudian variabel terbaik digunakan untuk membagi *node* tersebut.



Gambar 2. 2 Ilustrasi Proses Algoritma *Random Forest*

(Sumber : analyticsvidhya.com)

Berikut cara kerja *Random Forest*:

a. Pemilihan data secara acak:

Pemilihan data secara acak ini menggunakan *bootstraping*. *Bootstraping* adalah teknik *sampling* dengan penempatan yang digunakan untuk membuat *Dataset* yang baru dari *Dataset* yang ada. Pada dasarnya *bootstraping* dilakukan dengan cara mengambil sampel secara acak dari *Dataset* asli sebanyak jumlah data yang sama dengan *Dataset* asli, namun dengan penempatan (beberapa data dapat muncul lebih dari sekali dalam sampel tersebut).

b. Membangun pohon keputusan:

1) Pilih fitur *subset*

Pilih sejumlah fitur secara acak dari total fitur yang tersedia. Jumlah fitur yang dipilih biasanya lebih kecil dari total fitur yang ada. Nilai ini dapat diatur sebagai *"auto", "sqrt", "log2*" atau nilai *integer* yang mewakili jumlah fitur yang akan dipertimbangkan.

2) Pilih fitur terbaik

Pilih fitur terbaik untuk membagi data pada setiap *node* pohon keputusan, pemilihan fitur ini dapat menggunakan kriteria seperti *entropi* atau *indeks gini* untuk mengukur seberapa baik fitur tersebut dapat memisahkan data ke dalam kelas-kelas yang berbeda. Berikut ini rumus dari *entropi* dan juga *indeks gini*:

- *Entropi*

$$E(S) = \sum_{i=1}^{C} -P_i \, log_2(P_i) \qquad (2.1)$$

Dimana:

E(S) = nilai entropi dari kumpulan data *S*

Pi = proporsi dari kelas i dalam kumpulan data *S*

C = jumlah total kelas yang mungkin ada dalam *Dataset*

Log2= fungsi logaritma basis 2

Setelah menghitung nilai *entropi* selanjutnya mencari *information gain* yang akan digunakan untuk mengukur efektifitas suatu atribut dalam pengklasifikasian data. Berikut ini rumus dari *information gain*:

$$Gain\ (S, A) = Entropi(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropi(S_i) \qquad (2.2)$$

Dimana:

Gain (S,A)= *information gain* dari sebuah atribut *A* terhadap data himpunan *S*

Entropi(S)= Nilai Entropi dari kumpulan data *S*

n = Jumlah partisi atribut *A*

S = Jumlah total Sampel dalam kumpulan data *S*

Si = Jumlah sampel pada kelas ke-i setelah data *S* dibagi berdasarkan atribut *A*.

Entropi (Si)= entropi dari himpunan data yang terbagi ke kelas ke-i

- *Indeks gini*

$$Gini(S) = 1 - \sum_{i=1}^{C}(P_i)^2 \quad (2.3)$$

Dimana:

Gini(S) = nilai indeks gini dari kumpulan data S

Pi = proporsi dari kelas i dalam kumpulan data S

C = jumlah total kelas yang mungkin ada dalam *Dataset*

Setelah menghitung *indeks gini* tahap selanjutnya yaitu menghitung *gini splitnya*, yaitu dengan menggunakan rumus:

$$Gini_{Split} = \sum_{i=1}^{P}\frac{n_i}{n} * Gini(S) \quad (2.4)$$

Dimana:

Gini split = indeks gini untuk pemisahan

ni = Jumlah sampel dalam partisi ke-i setelah pemisahan.

n = Jumlah total sampel sebelum pemisahan

Setelah menghitung nilai *information gainnya* pada *entropi* atau *indeks gini* maka selanjutnya akan di pilih fitur terbaik yang akan dijadikan sebagai

root node atau *node* pertama. Fitur terbaik untuk *entropi* adalah fitur yang memiliki nilai *gain* tertinggi sedangkan *untuk indeks gini* di ambil fitur yang memiliki *gini* terendah.

3) Membagi data

Gunakan fitur terbaik yang telah dipilih untuk membagi data pada *node* saat ini menjadi *subset* yang lebih kecil. Setiap *subset* akan menjadi cabang dari *node* saat ini dalam pohon keputusan.

Lakukan langkah-langkah di atas secara *rekursif* untuk setiap *node* anak (cabang) yang dihasilkan dari pembagian data. setelah membagi data pada *node* saat ini berdasarkan fitur terbaik, kita akan memiliki anak cabang baru yang merupakan *subset* dari data asli. Untuk setiap anak cabang ini, langkah-langkah yang sama diterapkan secara *rekursif*: memilih fitur terbaik untuk membagi data di *node* saat ini (yang sekarang adalah anak cabang), membagi data, dan membangun anak cabang baru. Proses ini terus berlanjut hingga tercapai kondisi berhenti yang ditentukan, seperti mencapai jumlah maksimum *node* atau sampel minimum pada sebuah *node*.

c. Ulangi langkah a-b sebanyak n kali untuk mendapatkan n pohon keputusan.

d. Prediksi dengan menggunakan *ensemble*

Setelah semua pohon keputusan dibangun, prediksi dilakukan dengan cara melakukan *voting* (menggunakan *mayoritas* suara untuk klasifikasi dan rata-rata untuk regresi).

Random Forest pertama kali diperkenalkan oleh Breiman pada tahun 2001 (Aziz, 2021). Penelitian ini menunjukkan beberapa kelebihan dari *Random Forest*, antara lain:

a. kemampuannya menghasilkan *error* yang lebih rendah,
b. memberikan hasil yang baik dalam klasifikasi,
c. dapat mengatasi data latih dalam jumlah besar dengan efisien,
d. dan merupakan metode yang efektif untuk mengestimasi data yang hilang.

Selain kelebihan *Random Forest* juga memiliki kekurangan yaitu:

a. Waktu pemrosesan yang lama karena menggunakan data yang banyak dan membangun *model tree* yang banyak pula untuk membentuk *Random trees* karena menggunakan *single processor.*
b. Interpretasi yang sulit dan membutuhkan mode penyetelan yang tepat untuk data.
c. ketika digunakan untuk regresi, mereka tidak dapat memprediksi di luar kisaran dalam data percobaan, hal ini di mungkinkan data terlalu cocok dengan kumpulan data pengganggu *(noisy).*

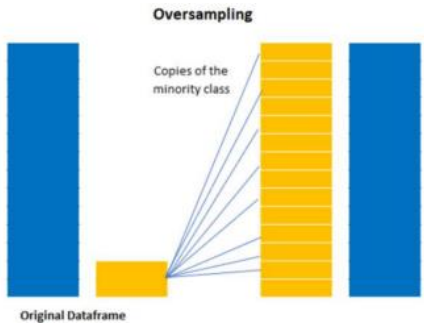
5. Ketidakseimbangan Data (Data *Imbalance*)

Proses klasifikasi pada *Dataset* yang memiliki ketidakseimbangan kelas merupakan tantangan utama dalam bidang *machine learning* dan data mining. Ketidakseimbangan kelas, atau *imbalanced* data, terjadi ketika distribusi kelas dalam *Dataset* tidak seimbang, dengan jumlah data *mayoritas* (*positif)* lebih banyak daripada jumlah data *minoritas* (*negatif*). Ketidakseimbangan ini dapat menyebabkan masalah *misclassification*, di mana *classifier* cenderung

memprediksi kelas *mayoritas* dan menganggap data *minoritas* sebagai *noise* atau *outlier,* yang dapat mengurangi kinerja *classifier*.

Ada beberapa metode yang dapat digunakan untuk mengatasi ketidakseimbangan kelas pada *Dataset*. Pertama, dapat dilakukan dengan cara menyeimbangkan distribusi kelas menggunakan metode *Oversampling* dan *undersampling*. Kedua, dapat dilakukan dengan pendekatan pada tingkat algoritma, seperti menciptakan algoritma baru atau *mentransformasi* metode yang ada untuk memperhitungkan kelas *minoritas*. Ketiga, dapat dilakukan dengan menggabungkan pendekatan algoritma dan pendekatan level data.

Salah satu metode *Oversampling* yang dapat digunakan adalah *Random Oversampling* (ROS), di mana data dari kelas *minoritas* ditambahkan ke dalam data *training* secara acak sampai jumlah data kelas *minoritas* sama dengan jumlah kelas *mayoritas*. Proses ini dimulai dengan menghitung selisih antara jumlah kelas *mayoritas* dan kelas *minoritas*, lalu dilakukan perulangan untuk menambahkan data kelas *minoritas* secara acak ke dalam data *training* sampai jumlahnya seimbang dengan kelas *mayoritas* (Ferdita Nugraha et al., 2022).



(Sumber: Ferditas Nugraha et al., 2022)

Gambar 2. 3 Ilustrasi Proses *Random Oversampling*

6. Random Search

Metode *Random Search* digunakan untuk memilih nilai *hyperparameter* secara *independen* dengan distribusi *probabilitas*, yang membuatnya efektif dalam *tuning hyperparameter* karena efisiensi waktu komputasinya. Dibandingkan dengan *Grid Search, Random Search* memiliki rentang percobaan yang lebih besar dengan jumlah percobaan yang sama, sehingga lebih efektif ketika jumlah dimensi *parameter* yang dicoba besar. Teknik *Random Search* secara acak memilih konfigurasi *hyperparameter* dari ruang *parameter* yang ditentukan. Meskipun teknik ini acak, hasilnya tetap dapat *optimal.* Setiap konfigurasi *hyperparameter* akan divalidasi menggunakan aturan *cross-validation*, dan hasil validasi akan disimpan. *Random Search* melakukan pencarian menyeluruh terhadap *hyperparameter* yang telah ditentukan, dan nilai *hyperparameter* terbaik yang dihasilkan dari proses *tuning* akan digunakan dalam pembentukan *model* (Abubakar *et al.*, 2023).

7. Confusion Matriks

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja suatu algoritma dalam mengklasifikasikan data ke dalam kelas yang berbeda. *Confusion matrix* mengandung empat istilah untuk merepresentasikan hasil dari proses klasifikasi, yaitu *True Positif (TP), False Positif (FP), True Negatif (TN), dan False Negatif (FN)* (Ferdita Nugraha et al., 2022).

Tabel 2. 1 *Confusion Matriks*

classification		*Actual*	
		TRUE	*FALSE*
prediction	*TRUE*	*True Positif (TP)*	*False Positif(FN)*
	FALSE	*False Negatif(FP)*	*True Negatif (TN)*

Penjelasannya:

a. *True Positif (TP*) = memiliki arti bahwa banyak data yang aktual kelasnya positif, kemudian *model* juga memprediksi positif
b. *True Negatif (TN)* = memiliki arti bahwa banyak data yang aktual kelasnya negative dan *model* juga memprediksi negatif
c. *False Positif* (*FP)* = memiliki arti banyak data yang aktual kelasnya negatif akan tetapi *model* memprediksi positif
d. *False Negatif (FN)* = memiliki arti banyak data yang aktual kelasnya positif, akan tetapi *model* memprediksi negatif

Dengan data *confussion matrix*, maka akan dapatkan sebuah data yang lain yang pastinya akan sangat berguna untuk mengukur performa sebuah algoritma atau *model* yang digunakan, adapun data tersebut antara lain:

a. Akurasi mengukur seberapa baik *model* membuat prediksi yang benar dari total prediksi yang dilakukan. Dalam konteks klasifikasi, akurasi memberikan gambaran mengenai seberapa sering *model* memprediksi kelas yang benar, baik

itu kelas positif maupun negatif. Untuk Formula menghitung nilai akurasi dapat dituliskan menggunakan rumus persamaan berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.5)$$

b. *Precision* mengukur seberapa baik *model* membuat prediksi yang benar untuk kelas *positif* dari total prediksi *positif* yang dilakukan. Dalam konteks klasifikasi, *precision* memberikan gambaran mengenai seberapa sering *model* memprediksi kelas *positif* dengan benar, di antara semua prediksi *positif* yang dibuat oleh *model*. Untuk Formula menghitung nilai *precision* dapat dituliskan menggunakan rumus persamaan berikut:

$$Presisi = \frac{True\ positif\ (TP)}{True\ positif\,(TP)+False\ Positif\,(FP)} \quad (2.6)$$

c. *Recall* adalah ukuran berapa banyak kasus positif yang di prediksi dengan tepat oleh pengklasikasian atas seluruh kasus positif dalam data. Untuk Formula menghitung nilai *Recall* dapat dituliskan menggunakan rumus persamaan berikut:

$$Recall = \frac{True\ Positif\ (TP)}{True\ positf\,(TP)+False\ negatif\,(FN)} \quad (2.7)$$

d. *F1-score* digunakan untuk menilai rata-rata *Precision* dan *Recall* hasil klasifikasi. Perhitungan *F1-score* dapat dinyatakan dalam bentuk formula sebagai berikut:

$$F1\ score = 2 * \frac{Recal*Precision}{Recal+Precision} \quad (2.8)$$

B. Penelitian Sebelumnya

Adapun beberapa penelitian sebelumnya yang dijadikan sebagai sumber referensi terkait topik penelitian ini:

Tabel 2. 2 Penelitian Sebelumnya

NO	Penelitian	Keterkaitan
1.	Penulis: Muhammad Fandru Al Rifqi, dkk. Judul: ***Comparative Analysis Of Phishing Website Prediction Classification Algorithm Using Logistic Regression, Decision Tree, And Random Forest (2022).*** Pada penelitian ini peneliti membandingkan hasil analisa dari penggunaan 3 algoritma klasifikasi untuk deteksi *website phishing*, yaitu *logistic regression, decision tree,* dan *Random Forest*. Hasil dari penelitian ini di dapatkan bahwa algoritma *Random Forest* merupakan algoritma terbaik dalam mengklasifikasikan *website phishing* dengan tingkat akurasi sebersar 97,10% disusul oleh algoritma *decision tree* dengan tingkat akurasi 94,57%, dan *logistic regression* dengan akurasi 92,76% (Fandru Al Rifqi *et al.*, 2022).	Persamaannya terletak pada objek dan algoritma yang di gunakan. Sedangkan perbedaannya yaitu pada sebelumnya membandingkan hasil analisa dari penggunaan 3 algoritma klasifikasi yaitu *logistic regression, decision tree*, dan *Random Forest* sedangkan pada penelitian ini yaitu menggunakan algoritma *Random Forest* untuk deteksi *web phishing* dengan teknik *Random Oversampling* untuk mengatasi kelas yang tidak seimbang.
2.	Peneliti: Sri Diantika Judul: **Penerapakan teknik *Random Oversampling* untuk mengatasi *imbalance class* dalam klasifikasi *website phishing* menggunakan algoritma *LIGHTGBM* (2023).** Pada penelitian ini peneliti menggunakan teknik *resampling* yaitu *Random Oversampling* untuk mengatasi kelas yang tidak seimbang dari *Dataset* yang digunakan dan menggunakan algoritma *LIGHTGBM*	Persamaanya terletak pada kasus yang digunakan yaitu deteksi *website phishing* dan teknik *Random Oversampling* yang digunakan sedangkan perbedaannya terletak pada algoritma klasifikasi yang digunakan yaitu pada penelitian sebelumnya

	untuk melakukan klasifikasi *website phishing*. Berdasarkan pengujian yang telah dilakukan pada penelitan ini didapatkan nilai akurasi sebesar 96,9%, *Recall* 96,9%, dan *F1-score* 96,9%. (Diantika, 2023).	menggunakan algoritma LIGHTGBM sedangkan dalam penelitian ini menggunakan algoritma *Random Forest*.
3.	Peneliti: Anggit Ferdita, dkk. Judul: **Penerapan metode stacking dan *Random Forest* untuk meningkatkan kinerja klasifikasi pada proses deteksi *web phishing* (2022).** Pada penelitian ini peneliti menerapkan algoritma *Random Forest* dan metode *stacking* yaitu dengan menggabungkan algoritma *decision tree* dan *naïve bayes* untuk meningkatkan kinerja klasifikasi untuk deteksi web *phishing*. Hasil dari penelitian ini didapatkan algoritma *Random Forest* menghasilkan nilai akurasi tertinggi, yaitu 96.4% pada *Dataset web* kelas biner, sedangkan metode *stacking* yang menggabungkan algoritma *Decision Tree* dengan *Naïve Bayes* memberikan kinerja terbaik pada *Dataset* web kelas banyak, dengan nilai akurasi sebesar 88.8% (Ferdita Nugraha et al., 2022).	Persamaannya terletak pada objek yang sama dan algoritma yang sama. Sedangkan perbedaannya yaitu pada sebelumnya lebih fokus kepada penerapan metode stacking dan *Random Forest* untuk meningkatkan kinerja klasifikasi untuk deteksi web *phishing* sedangkan penelitian ini fokusnya yaitu penerapan algoritma *Random Forest* dengan teknik *Random Oversampling* untuk klasifikasi web *phishing*.
4.	Peneliti: farida, dkk. Judul: **Perbandingan *logistic regression dan Random Forest* menggunakan *corelation-based feature selection* untuk deteksi *website* (2023).** Penelitian ini membandingkan dua algoritma klasifikasi, yaitu *Random Forest* dan logistic regression, dalam mendeteksi *website phishing* menggunakan correlation-based feature selection (CFS). Hasilnya menunjukkan bahwa penggunaan CFS	Sama-sama menggunakan algoritma *Random Forest* dan objek yang sama perbedaannya pada *dataset* yang digunakan dan pada penelitan sebelumnya mereka menggunakan *corelation based feature* untuk memilih beberapa fitur sedangkan dalam penelitian ini

	meningkatkan akurasi *Random Forest* dari 96.834% menjadi 97.015%, sementara logistic regression mengalami penurunan yang tidak signifikan dari 93.935% menjadi 92.718%. Dengan demikian, *Random Forest* dengan CFS lebih akurat dalam mendeteksi *website phishing* daripada logistic regression.	menggunakan semua fitur yang ada, lalu pada penelitan sebelumnya juga tidak menggunakan *Random Oversampling* untuk mengatasi kelas yang tidak seimbang.
5.	Penulis: Muhamad Abdul Ghanni Al Ghifari Judul: **Implementasi ekstensi google chrome dalam mendeteksi situs web *phishing* menggunakan algoritma *Random Forest* (2022).** Pada penelitian ini peneliti menggunakan pendekatan *machine learning* yaitu metode *Random Forest*, dengan mengimplementasikannya ke dalam ekstensi peramban seperti *Google Chrome* untuk deteksi web *phishing*. Hasil evaluasi dari penelitian ini didapatkan Hasil evaluasi *model* klasifikasi mempunyai hasil akurasi 90,2%, *Recall* 88,8% dan *precision* 88,8%. Ekstensi peramban dilakukan evaluasi kinerja menggunakan data baru dengan akurasi 88%, *Recall* 84% dan *precision* 91,3% (Abdul Ghanni Al Ghifari et al. 2022).	Persamaannya terletak pada algoritma yang digunakan dan objeknya sedangkan perbedaannya yaitu sebelumnya menggunakan pendekatan *machine learning* yaitu metode *Random Forest*, dengan mengimplementasikannya ke dalam ekstensi peramban seperti *Google Chrome* untuk deteksi *web phishing.* Sedangkan penelitian ini akan menerapkan algoritma *Random Forest* dan teknik *Random Oversampling* untuk melakukan klasifikasi *web phishing*
6.	Peneliti: Vikki Aprelia Windarni, dkk. Judul: **Deteksi *website phishing* menggunakan teknik filter pada *model machine learning* (2023).** Penelitian ini membandingkan algoritma klasifikasi yaitu *Random Forest*, *naïve bayes* dan *decision tree* dengan menerapkan teknik filter yaitu *pearson corelation* untuk memilih atribut berdasarkan korelasinya. Hasil dari penelitian ini didapatkan algoritma yang	Persamaannya terletak pada objek yang sama yaitu deteksi *website phishing* dan algoritma yang digunakan sedangkan perbedaannya pada penelitian sebelumnya lebih fokus kepada penggunaan teknik filter untuk memilih atribut terbaik

	paling efektif mendeteksi *website phishing* setelah dilakukan filter menggunakan *pearson correlation* yaitu *Random Forest* karena memiliki tingkat akurasi sebesar 96,3%. Sedangkan *naïve bayes* hanya memiliki akurasi 60,4% dan *decision tree* 94,4% (Aprelia Windarni *et al.*, 2023).	dengan membandingkan tiga algoritma yaitu *decision tree, naïve bayes, dan Random Forest* sedangkan pada ini fokus kepada membandingkan apakah ada pengaruh penggunaan teknik terhadap algoritama *Random Forest* untuk mengatasi kelas yang tidak seimbang.
7.	Peneliti: Pungkas Subarkah Judul: **Identifikasi *website phishing* menggunakan algoritma *Classification and Regression Tree* (CART)(2021).** Pada penelitian ini peneliti menggunakan algoritma *machine learning* yaitu *classification and regression tree* (CART) untuk melakukan identifikasi *website phishing*. Hasil dari penelitian ini di dapatkan bahwa dari nilai *confusion matrix* diperoleh hasil akurasi sebesar 95.28%, dengan rincian nilai *Precision* sebesar 0.953%, nilai *Recall* sebesar 0.953% dan nilai *F-Measure* sebesar 0.953% (Subarkah and Ikhsan, 2021).	Persamaannya terletak pada objeknya sedangkan perbedaannya terletak pada algoritma yang digunakan.
8.	Penulis: Sri Diantika, dkk. Judul**: Penerapan Teknik *Random Oversampling* Untuk Memprediksi Ketepatan Waktu Lulus Menggunakan Algoritma *Random Forest* (2024).** Pada penelitian ini peneliti menerapkan teknik *Random Oversampling* untuk mengatasi kelas yang tidak seimbang dari data yang digunakan dengan menggunakan algoritma *Random Forest* untuk memprediksi waktu kelulusan mahasiswa. penelitian ini bertujuan untuk	Persamaannya terletak pada metode klasifikasi yang digunakanan yaitu *Random Forest* dan *teknik resampling* yang digunakan yaitu teknik *Random Oversampling*. Sedangkan perbedaannya pada objeknya

	mengidentifikasi ketepatan waktu kelulusan mahasiswa sejak dini dengan memanfaatkan nilai Indeks Prestasi Kumulatif (IPK) yang dicapai mahasiswa selama masa *study* mereka. Penelitian ini menggunakan *split validation* dengan membagi data pembelajaran 50% dan data pengujian 50%. Untuk mengkaji *model* yang dibentuk, Penulis memakai *metrics evaluation* seperti akurasi, *Precision*, serta *Recall*. Hasil dari penelitian ini menampilkan bahwa *model* yang diusulkan bisa dengan baik melaksanakan prediksi dibanding dengan *model* lainnya, yaitu dengan hasil nilai *Precision* sebesar 87,05%, uji akurasi sebesar 90,04%, *Recall* 90,04%. Dari hasil ini dapat diartikan bahwa algoritma *Random Forest* dinilai baik dalam memprediksi ketepatan waktu lulus seorang mahasiswa (Diantika *et al.*, 2024).	
9.	Penulis: Gabriel Advent Batan, dkk. Judul: **Penerapan metode *Random Forest*, gaussian NB, Dan KNN terhadap data *Unbalance* dan data *balance* menggunakan *Random over sampling* untuk klasifikasi senyawa keladi tikus (2023).** Penelitian ini bertujuan untuk mengklasifikasikan *Dataset* senyawa keladi tikus menggunakan metode *Random Forest*, *Gaussian* NB, dan KNN pada *Dataset* yang tidak seimbang dan seimbang dengan menerapkan *Random Over Sampling*. Metode penelitian mencakup penerapan *Random Forest*, KNN, dan Gaussian NB pada *Dataset* asli (data tidak seimbang) untuk pembentukan *model* data *training* dan pengujian menggunakan data uji. Evaluasi kinerja algoritma dilakukan dengan	Persamaannya terletak pada metode klasifikasi yang digunakanan yaitu *Random Forest* dan *teknik resampling* yang digunakan yaitu teknik *Random Oversampling*. Sedangkan perbedaannya terletak pada objek penelitiannya.

	menggunakan *confusion matrix*, sedangkan metode KNN dievaluasi dengan *K-fold cross validation*. Sebelum data diseimbangkan, akurasi pada data tidak seimbang mencapai rata-rata 80%, namun *parameter* lainnya memiliki nilai yang rendah. Setelah menerapkan *Random Oversampling*, akurasi meningkat untuk *Random Forest*, tetapi terjadi penurunan akurasi rata-rata pada KNN dan *Gaussian NB*. Penurunan ini disebabkan oleh penduplikatan pada kelas *minoritas* dan pengaruh nilai k yang terlalu besar pada KNN. Analisis waktu komputasi menunjukkan bahwa *Random Forest* dan KNN memerlukan waktu yang lebih lama daripada *Naïve Bayes*. Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa algoritma *Random Forest* adalah yang terbaik untuk melakukan klasifikasi pada *Dataset* senyawa keladi tikus, baik pada data tidak seimbang maupun data seimbang (Advent Batan *et al.*, 2023).	
10.	Penulis: Siti Mutmainah Judul: **Penanganan *Imbalance* Data Pada Klasifikasi Kemungkinan Penyakit Stroke (2021).** Pada penelitian ini peneliti menerapkan teknik *Random Oversampling* dan *Random* undersampling untuk mengatasi kelas yang tidak seimbang pada data kemungkinan penyakit stroke dengan menggunakan algoritma *Random Forest*. Hasil dari penelitian ini didapatkan bahwa teknik *Random Oversampling* mendapat performa yang lebih tinggi yaitu 95% daripada teknik *Random undersampling* yang mendapat performa 76% (Mutmainah, 2021).	Persamaannya terletak pada metode klasifikasi yang digunakanan yaitu *Random Forest* dan teknik *resampling* yang digunakan yaitu teknik *Random Oversampling*. Sedangkan perbedaannya pada objeknya

11	Muhammad Ali Abubakar dkk. Judul: ***Random Forest* Dengan *Random Search* Terhadap Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung** Pada penelitian ini peneliti membandingkan kinerja algoritma *Random Forest* tanpa menggunakan *Random Search* dan menggunakan *Random Search* terhadap ketidakseimbangan kelas pada *dataset Heart Failure Clinical Record* di dapatkan bahwa *Random Forest* dengan *Random Search* memiliki nilai AUC tertinggi yaitu mencapai 0.906 sedangkan *Random Forest* tanpa *Random Search* hanya mendapatkan nilai AUC 0.866% (Abubakar *et al.*, 2023).	Persamaannya terletak pada algoritma yang digunakan dan *hyperparameter* tuning yang digunakan, sedangkan perbedaannya pada objeknya dan teknik resampling yang digunakan dan objek yang digunakan

BAB V

PENUTUP

A. Kesimpulan

1. *Model Random Forest* tanpa teknik *Oversampling* menunjukkan performa yang sangat baik dalam mengklasifikasikan *website phishing* dan *Non-Phishing*, dengan akurasi 95.93%. *Model* ini memiliki recall yang tinggi untuk kedua kelas, yaitu 97.03% untuk *Non-Phishing* dan 97.73% untuk *phishing*, yang menunjukkan bahwa *model* sangat sensitif dalam mendeteksi sampel dari kedua kategori. Namun, dari segi precision, masih terdapat beberapa kesalahan klasifikasi, terutama untuk *Non-Phishing*, dengan nilai 93.65%. Precision untuk *phishing* lebih tinggi, yaitu 95.11%, yang berarti *model* lebih akurat dalam mengenali *phishing* dibandingkan *Non-Phishing*. Secara keseluruhan, keseimbangan antara precision dan recall terlihat dari F1-score, yaitu 95.31% untuk *Non-Phishing* dan 96.41% untuk *phishing*. Hal ini menunjukkan bahwa meskipun terdapat sedikit ketidakseimbangan dalam prediksi antara kedua kelas, *model* tetap memiliki performa yang baik dalam mendeteksi *website phishing* tanpa terlalu mengorbankan akurasi pada kelas lainnya.
2. *Random Oversampling* meningkatkan akurasi menjadi 96.16%, tetapi menyebabkan penurunan *recall* untuk kelas *Non-Phishing* (94.88%) dari sebelumnya 97.03%. Sebaliknya, *precision* meningkat menjadi 96.36% untuk *Non-Phishing* dan 96.00% untuk *phishing*, yang berarti *model* lebih selektif dalam mengklasifikasikan *phishing*, sehingga jumlah *False Positive*s

berkurang. *F1-score* juga meningkat, menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall*. Namun, penurunan *recall* ini dapat menyebabkan *model* kurang mampu mengenali sampel baru dari kelas minoritas (*Non-Phishing*) dengan baik.

SMOTE menghasilkan akurasi 95.62%, sedikit lebih rendah dibandingkan metode lainnya. Meskipun *precision* untuk kelas *Non-Phishing* meningkat menjadi 96.89%, *recall* untuk kelas ini justru menurun menjadi 94.41%, yang menunjukkan bahwa meskipun SMOTE menambahkan sampel sintetis, *model* tetap kesulitan mengenali kelas *Non-Phishing*. *Recall* kelas *phishing* juga sedikit menurun menjadi 96.86%, sedangkan *precision* untuk kelas ini turun menjadi 94.37%, yang mengindikasikan peningkatan jumlah *False Positive* untuk *phishing*.

Secara keseluruhan, *model Random Forest* sudah cukup andal dalam mendeteksi *phishing* dan *Non-Phishing* tanpa perlu menerapkan *Oversampling*. *Random Oversampling* meningkatkan akurasi dan *precision*, tetapi menurunkan *recall* untuk kelas *Non-Phishing*, yang dapat berdampak pada kemampuan *model* mengenali kelas minoritas. SMOTE meningkatkan *precision* untuk kelas *Non-Phishing*, tetapi menurunkan *recall* pada kedua kelas serta meningkatkan *False Positive*s untuk *phishing*. Oleh karena itu, meskipun teknik *Oversampling* dapat meningkatkan beberapa aspek performa *model*, efeknya terhadap keseimbangan klasifikasi perlu diperhatikan agar *model* tetap efektif dalam mendeteksi *phishing* tanpa mengorbankan akurasi kelas lainnya.

B. Saran

1. Membandingkan performa algoritma *Random Forest* dengan algoritma *machine learning* lainnya guna mengetahui apakah algoritma lain dapat memberikan hasil yang lebih baik atau lebih sesuai untuk *dataset* yang serupa.
2. Melakukan perbandingan antara teknik *Random Oversampling* dan *SMOTE* dengan menggunakan *dataset* yang berbeda, atau mencoba teknik resampling lainnya untuk mendapatkan pemahaman yang lebih mendalam mengenai pengaruh masing-masing teknik dalam mengatasi ketidakseimbangan kelas.

DAFTAR PUSTAKA

Abdul Ghanni Al Ghifari, M., Hananto, B. and Tri Wahyono, B. (2022) Implementasi Ekstensi Google Chrome Dalam Mendeteksi Situs Web *Phishing* Menggunakan Algoritma *Random Forest*.

Abubakar, M.A. et al. (2023) '*Random Forest* Dengan *Random Search* Terhadap Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung', Jurnal Informatika, 10(1), pp. 13–18. Available at: https://doi.org/10.31294/inf.v10i1.14531.

Advent Batan, G. et al. (2023) Penerapan Metode *Random Forest*, Gaussian NB, Dan KNN Terhadap Data Unbalance dan Data Balance Menggunakan *Random* Over Sampling Untuk Klasifikasi Senyawa Keladi Tikus.

Aprelia Windarni, V. et al. (2023) Deteksi *Website Phishing* Menggunakan Teknik Filter Pada *Model* Machine LEARNING, Information System Journal (INFOS) |.

Aziz, W.A. (2021) Implementasi Metode *Random Forest* Pada Klasifikasi Data Ulasan Konsumen Perusahaan (Studi Kasus: Aplikasi KAI Access).

Diantika, S. (2023) Penerapan Teknik *Random Oversampling* Untuk Mengatasi Imbalance Class Dalam Klasifikasi *Website Phishing* Menggunakan Algoritma Lightgbm, Jurnal Mahasiswa Teknik Informatika.

Diantika, S. et al. (2024) Penerapan Teknik *Random Oversampling* Untuk Memprediksi Ketepatan Waktu Lulus Menggunakan Algoritma *Random Forest*, Computer Science (CO-SCIENCE. Available at: https://www.kaggle.com/.

Diki Wahyudi (2020) 'Aplikasi Pendeteksi *Website Phishing* Menggunakan Machine Learning'.

Fandru Al Rifqi, M. et al. (2022) 'Comparative Analysis Of *Phishing Website* Prediction Classification Algorithm Using Logistic Regression, Decision Tree, And *Random Forest*', Jurnal Infokum, 10(2). Available at: http://infor.seaninstitute.org/index.php/infokum/index.

Farida and Mustopa, ali (2023) 'Perbandingan Logistic Regression dan *Random Forest* menggunakan Correlation-based Feature Selection untuk Deteksi *Website Phishing*', SISTEMASI: Jurnal Sistem Informasi, 12(1). Available at: http://sistemasi.ftik.unisi.ac.id.

Ferdita Nugraha, A. et al. (2022a) 'Penerapan metode Stacking dan *Random Forest* untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web *Phishing*', Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan, 7(1).

Ferdita Nugraha, A. et al. (2022b) 'Penerapan metode Stacking dan *Random Forest* untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web *Phishing*', Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan , 7(1).

Kalabarige, L.R. et al. (2023) 'A Boosting-Based Hybrid Feature Selection and Multi-Layer Stacked Ensemble Learning *Model* to Detect *Phishing Website*s', IEEE Access, 11, pp. 71180–71193. Available at: https://doi.org/10.1109/ACCESS.2023.3293649.

Mutmainah, S. (2021) 'Penanganan Imbalance Data Pada Klasifikasi Kemungkinan Penyakit Stroke', Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi, 1(1). Available at: https://doi.org/10.20885/snati.v1i1.2.

Prawira, P.A. et al. (2021) 'Implementation of the Support Vector Machine (SVM) Algorithm in Classifying *Website Phishing*', Jurnal Elektronik Komputer Udayana, 9(4), pp. 2654–5101.

Ramadhan, R.P. and Desyani, T. (2023) 'Implementasi Algoritma J48 Untuk Identifikasi *Website* Phising', Teknik dan Multimedia, 1(2).

Sahara Munte, R. et al. (2023) 'Jenis Penelitian Eksperimen dan *Non*eksperimen (Design Klausal Komparatif dan Design Korelasional)', 7(3).

Subarkah, P. and Ikhsan, A.N. (2021) 'Identifikasi *Website Phishing* Menggunakan Algoritma Classification And Regression Trees (CART)', Jurnal Ilmiah Informatika, 6(2), pp. 127–136. Available at: https://doi.org/10.35316/jimi.v6i2.1342.

Suci Amaliah, Nusrang, M. and Aswi, A. (2022) 'Penerapan Metode *Random Forest* Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng', VARIANSI: Journal of Statistics and Its application on Teaching and Research, 4(3), pp. 121–127. Available at: https://doi.org/10.35580/variansiunm31.

Sulistiyono, M. et al. (2021) SISTEMASI: Jurnal Sistem Informasi Implementasi Algoritma Synthetic Minority Over-Sampling Technique untuk Menangani Ketidakseimbangan Kelas pada *Dataset* Klasifikasi. Available at: http://sistemasi.ftik.unisi.ac.id.

Supriadi, M.R. and Andarsyah, R. (2023) Deteksi Halaman *Website Phishing* Menggunakan Algoritma Machine Learning Gradient Boosting Classifier. Edited by N. Harani. Bandung: Buku Pedia.

Tanti, T. (2023) '*Random Oversampling*, Chi-Square, dan AdaBoost dalam Penanganan Ketidakseimbangan Kelas pada Klasifikasi C5.0', JURNAL MEDIA INFORMATIKA BUDIDARMA, 7(2), p. 714. Available at: https://doi.org/10.30865/mib.v7i2.5862.

Zieni, R. et al. (2023) '*Phishing* or Not *Phishing* ? A Survey on the Detection of *Phishing Website*s', IEEE Access, 11(January), pp. 18499–18519. Available at: https://doi.org/10.1109/ACCESS.2023.3247135.