PENERAPAN TEKNIK RANDOM OVERSAMPLING UNTUK MENGATASI IMBALANCE CLASS DALAM KLASIFIKASI WEBSITE PHISHING MENGGUNAKAN ALGORITMA LIGHTGBM

Sri Diantika
Program Studi Teknik Informatika S1, Fakultas Teknik
Universitas Bina Sarana Informatika
Sri.szd@bsi.ac.id

ABSTRAK

Kemudahan mendapatkan Segala informasi dari *website*, membuat masyarakat lebih memilih *website* sebagai sarana mencari sebuah informasi yang cepat. maraknya penggunaan *website*, membuat beberapa oknum yang tidak bertanggungjawab menyalahgunakan penggunaan *website*, seperti melakukan penipuan atau *phishing*. *Phishing* menjadi salah satu kejahatan siber yang memiliki sifat mengancam serta menjebak *user* dengan cara memancing *user* atau pengguna untuk secara tidak langsung memberikan suatu informasi kepada pelaku *phishing*. Dari permasalahan tersebut peneliti melakukan penelitian menggunakan *dataset* publik dari Kaggle yang berisi kumpulan URL situs web berjumlah lebih dari 11000 situs web. Peneliti mengusulkan model untuk mengklasifikasikan *website phishing* dan *non phishing* menggunakan lightGBM. Kami juga menerapkan *Random Over Sampling* (ROS) pada data untuk menyelesaikan permasalahan yang ada dikelas data yaitu adanya kelas data yang tidak seimbang. Eksperimen kami menunjukkan bahwa metode yang diusulkan mencapai akurasi sebesar 96,9%, *recall* 96,9%, , F1-*Score* 96,9%, dan nilai ROC 99,7%. Ini secara signifikan lebih baik daripada beberapa metode lain

Kata kunci: *website phishing, klasifikasi, lightGBM, random oversampling, data mining*

1. PENDAHULUAN

Perkembangan ilmu teknologi yang semakin maju, membuat masyarakat dimudahkan dalam beberapa hal, salah satunya mencari informasi yang dibutuhkan. Salah satu cara memenuhi kebutuhan informasi tersebut adalah dengan menggunakan *website*.

Website merupakan suatu kumpulan homepage atau halaman yang difungsikan untuk menyajikan suatu informasi yang berisi gambar, teks, suara, animasi, *video* maupun gabungan dari semua objek tersebut, *website* dapat bersifat statis ataupun dinamis yang menyelaraskan suatu rangkaian bangunan yang saling berkaitan, kemudian masing-masing ditautkan dengan jaringan sehingga mampu memberikan suatu informasi yang dibutuhkan atau dicari oleh pengguna [1]

Dengan kata lain, *website* merupakan suatu situs yang dapat dikunjungi atau diakses oleh user guna mendapatkan suatu informasi yang dibutuhkan dengan cepat. *Website* sendiri ada karena adanya perkembangan dari teknologi informasi dan komunikasi saat ini [2] maraknya penggunaan *website*, membuat beberapa oknum yang tidak bertanggungjawab menyalahgunakan penggunaan *website*, seperti melakukan penipuan atau phishing

Phishing menjadi bagian dari kejahatan dunia maya yang memiliki sifat mengancam serta menjebak user, untuk tahapannya yaitu pertama pelaku akan memancing user atau pengguna untuk secara tidak langsung memberikan suatu informasi kepada pelaku *phishing*. Sebagian besar pelaku phishing ini memakai alamat link yang ketika diklik kemudian akan menuju ke *website* palsu untuk menjebak target. Kegiatan Phishing dapat berpotensi menimbulkan kerugian, baik dalam hal kerugian privasi, eksploitasi data bahkan hingaa ke kerugian *financial* [3]

Cara kerja pada situs *website phishing*, pertama korban atau pengunjung *website phishing* akan diarahkan untuk memasukkan suatu informasi yang memiliki sifat rahasia pribadi, contohnya seperti kata sandi atau password serta nomor rekening bank dan alhasil data tersebut digunakan untuk pencurian privasi atau identitas. Di sisi lain pelaku *phishing* juga memanfaatkan suatu tools untuk mencuri kode sumber laman *website* yang legal kemudian akan menggantinya dengan sebuah *website* yang ilegal, Selain itu, pelaku *phishing* juga akan membangun suatu embedding link untuk mendapatkan informasi yang sifatnya pribadi milik korban *phishing* [4]

Kasus *Phishing* ini sudah marak terjadi, sehingga ada beberapa penelitian yang telah dilakukan guna mendeteksi *website phishing* salah satunya penelitian [5] mengaplikasikan metode algoritma random forest, untuk besaran akurasi yang didapat adalah 90,77%. Kasus *phishing* ini juga bukan hanya terjadi melalui *website*, pada penelitian [6] kasus phishing kerap terjadi di *social media*, email, *website*, dan malware [4]

Dari banyaknya kasus *phishing* ini, peneliti akan melakukan penelitian untuk mengklasifikasikan *website phishing* menggunakan model algoritma LightGBM. Pada penelitian ini, dataset yang dijadikan sebagai bahan atau objek penelitian merupakan data yang diambil secara umum atau bersifat publik yaitu diambil dari kaggle. Karena

dataset memiliki ketidakseimbangan kelas data, maka peneliti akan melakukan teknik resampling menggunakan *random oversampling* (ROS).

2. TINJAUAN PUSTAKA

Tinjauan pustaka yang dijabarkan oleh peneliti dalam penelitian ini menggunakan beberapa referensi, seperti dari buku, prosiding dan jurnal. Berikut tinjauan pustaka yang mendukung teoritis dari penelitian ini

2.1. *Data Mining*

Data mining adalah suatu teknik yang dilakukan guna memahami pola atau informasi menarik dari perspektif yang berbeda menggunakan metode atau teknik tertentu. Teknik atau metode yang bisa digunakan dalam proses *data mining* jumlahnya sangatlah bervariasi, sehingga pemilihan teknik, algoritma atau metode yang pas dan sesuai sangat bertumpu pada proses dan tujuan *Knowledge discovery in database* (KDD) secara menyeluruh.
Data mining juga kerap kali disebut sebagai bagian dari *knowledge discovery in database* (KDD) yaitu suatu tindakan yang meliputi proses untuk mengumpulkan data hingga kegiatan pemanfaatan data historis guna menemukan keteraturan suatu hubungan atau pola dalam set data yang memiliki ukuran besar. Keluaran yang dihasilkan dari proses data mining dapat dipergunakan untuk memperbaiki proses pengambilan keputusan diwaktu yang akan datang [7] Dari penjelasan mengenai data mining yang telah diuraikan, hal pokok yang perlu diketahui terkait dengan data mining adalah [8]:

a. Data mining merupakan suatu sebuah proses otomatis terhadap data yang telah ada.
b. Data yang akan diproses dalam data mining berupa data yang berjumlah besar
c. Misi yang dituju dengan melakukan proses data mining ini adalah untuk memperoleh pola atau keterkaitan yang akan berkemungkinan dapat memberikan petunjuk yang bermanfaat.

2.2. Klasifikasi

Salah satu tugas yang dapat dilakukan oleh data *mining* adalah untuk klasifikasi. Klasifikasi adalah suatu proses yang dilakukan guna mengelompokkan objek yang memiliki karakteristik, pola atau ciri yang sama ke dalam beberapa kelas [9]

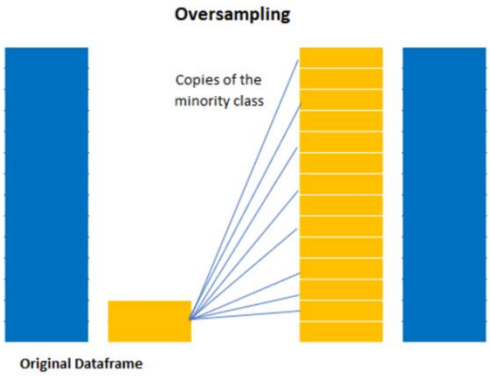
Algoritma klasifikasi yang banyak digunakan data mining, yaitu analisa statistik, Decision atau classification trees, Algoritma Genetika, Bayesian classifiers atau Naïve Bayes classifiers, k-nearest neighbor, Neural networks, Rough sets, Memory based reasoning, Metode Rule Based dan Support vector machines (SVM) [10]

2.3. Ketidakseimbangan Data (Data *Imbalance*)

Melakukan proses klasifikasi pada data yang memiliki kelas tidak seimbang adalah masalah utama yang harus diselesaikan pada bidang *machine learning* dan *data mining*. Dalam suatu sistem komputasi, ketidakseimbangan data (*imbalanced* data) adalah proses pendistribusian data yang memiliki kelas data tidak seimbang. Jumlah data mayoritas (positif) yang lebih banyak dibandingkan dengan jumlah data minoritas (negatif). Ketidakseimbangan data ini dapat memungkinkan menimbulkan kejadian *misclassification*, dimana *classifier* lebih condong kearah data mayoritas. Data minoritas akan dianggap sebagai *noise* dan *outlier* serta dapat menurunkan kinerja dari *classifier*[11]

Beberapa metode yang dapat digunakan untuk menyelesaikan permasalahan pada jumlah kelas data yang tidak seimbang (*imbalance*) dapat diatasi menggunakan tiga *opsi*. *Opsi* pertama yaitu dengan mencoba menyeimbangkan pendistribusian kelas data yaitu dengan menerapkan metode *oversampling* dan *undersampling*. *Opsi* kedua dapat diatasi dengan melakukan pendekatan tingkat algoritma yaitu dengan mencoba membangun algoritma baru ataupun mentransformasikan metode yang sudah ada untuk memperhitungkan arti dari kelas minor. Opsi ketiga yaitu dengan menggabungkan antara pendekatan algoritma dengan pendekatan level data [12]

Oversampling merupakan metode Pemerataan data minoritas sehingga menjadi sebanyak data mayoritas [13] Pada penelitan ini metode *random oversampling* (ROS) diterapkan sebagai teknik *oversampling*. *Random Oversampling* (ROS) merupakan pemberian data dari kelas minoritas ke dalam data *training* secara acak. Proses pemberian data ini diulang sampai jumlah data kelas minoritas sama rata dengan jumlah kelas mayoritas. Langkah yang dilakukan pada saat awal adalah dengan menghitung selisih antara kelas mayoritas dan kelas minoritas. Setelah itu, dilakukan perulangan sebanyak hasil penghitungan beda data sambil membaca data kelas minoritas secara acak dan dimasukkan ke dalam data *training* [14]



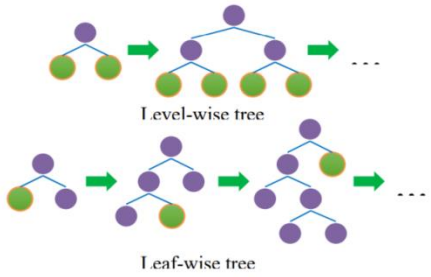
Gambar 1. *Random Oversampling* (ROS)

2.4. LightGBM

LightGBM merupakan bagian dari keluarga algoritma *gradient boosting* yang menerapkan algoritma pembelajaran berbasis pohon keputusan atau *tree*, LightGBM ini memiliki suatu keunggulan yaitu ia menerapkan *Exclusive Feature Bundling*

(EFB) dan *Gradient-Based One Side Sampling* (GOSS) dan untuk meningkatkan akurasi dengan tetap menjaga kekompleksitasan algoritmanya [15]

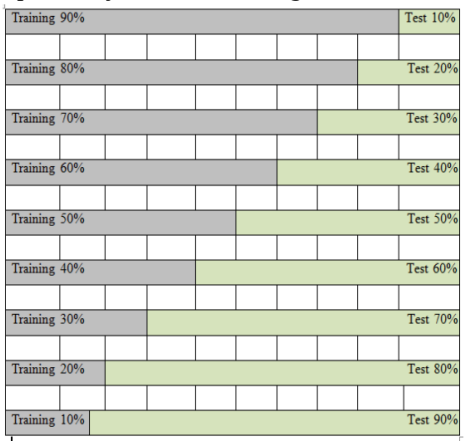
Katrakteristik yang dimiliki LightGBM membuatnya berbeda dengan algoritma *tree boosting* yang lainnya, yaitu LightGBM dapat membelah pohon secara *horizontal* atau memanjang dengan yang paling cocok atau sering disebut *leaf-wise tree growth*. Sedangkan algoritma *tree boosting* yang lain, mereka membelah atau mebagi pohonnya secara mendalam atau sejajar (*level-wise tree growth*), dari hal ini dapat dilihat bahwa pada lightGBM, ketika tumbuh pada daun yang sama algoritma *leaf-wise* dapat mengurangi lebih banyak kerugian daripada algoritma *level wise* dan juga dapat menghasilkan akurasi yang jauh lebih baik yang tidak dapat dipenuhi oleh algoritma *boosting* lainnya, akan tetapi pada algoritma *leaf-wise* cenderung lebih rentan terkena *overfitting* [16] gambaran dari *level-wise tree* dan *leaf-wise tree* dapat dilihat pada gambar 2



Gambar 2. Konstruksi level-wise dan leaf-wise

2.5. *Split Validation*

Split validation merupakan cara menguji validitas dan memisahkan antara data latih (*training)* dengan data uji *testing*. Percobaan *training* dilakukan berdasarkan jumlah *split ratio* yang telah ditetapkan sebelumnya yaitu dengan menggunakan *split validation*, sisa dari *split ratio* ini akan dijadikan sebagai data *testing*. Data *testing* ini akan menjadi data pengujian keakuratan atau kebenaran hasil *training* atau pembelajaran yang telah dilakukan sebelumnya, data yang dijalankan untuk *testing* merupakan data yang belum dijalankan pada saat proses pembelajaran atau *training* [17]



Gambar 3. Ilustrasi *Split Validation*

2.6. *Confussion matrix*

Confussion matrix adalah sebuah tabel yang digunakan untuk melakukan pengukuran performa sebuah algoritma, apakah algortima tersebut memiliki performa yang bagus dalam membedakan *tuple* dari kelas yang berbeda [18] Dalam *confussion matrix* terdapat 4 istilah dalam merepresentasikan hasil proses klasifikasi yaitu *True Positif* (TP), *False Positif* (FP), *True Negatif* (TN) serta *False Negatif* (FN) [19]

Tabel 1. Pengujian *Confussion matrix*

Classification		**Prediction Class**	
		TRUE	**FALSE**
Actual	**TRUE**	*True Positif* (TP)	*False Negatif* (FN)
	FALSE	*False Positif* (FP)	*True Negatif* (TN)

Penjelasannya [20]:

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

a. Akurasi
Jumlah total seberapa seringnya sebuat model betul dalam mengklasifikasikan. Untuk Formula menghitung nilai akurasi dapat dituliskan menggunakan rumus persamaan berikut:

$$\text{Akurasi} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

b. Presisi
Tingkat keakuratan hasil dari klasifikasi dengan jumlah total pengenalan yang dilakukan sistem. Untuk Formula menghitung nilai presisi dapat dituliskan menggunakan rumus persamaan berikut:

$$\text{Presisi} = \frac{\text{True positif (TP)}}{\text{True positif (TP)+False positif (FP)}}$$

c. Recall
Recall menunjukan total data yang betul diklasifikasi dalam sebuah kelas dibagi dengan jumlah total dalam kelas tersebut. Untuk Formula menghitung nilai *recall* dapat dituliskan menggunakan rumus persamaan berikut:

$$Recall = \frac{True\ positif\ (\text{TP})}{True\ positif\ (\text{TP}) + False\ negatif\ (\text{FN})}$$

d. F1-*Score*
F1 *score* digunakan untuk menilai rata-rata *precision* dan *recall* hasil klasifikasi. Perhitungan

F1 *Score* dapat dinyatakan dalam bentuk formula sebagai berikut:

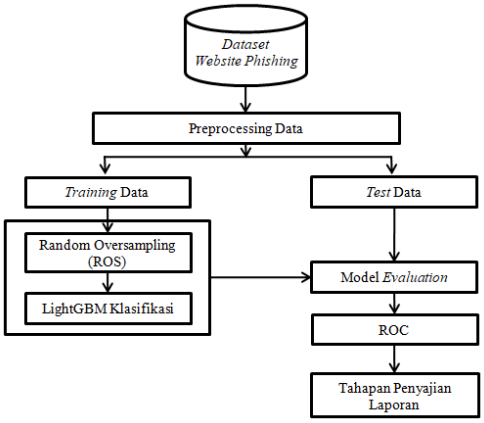
$$\text{F1 } score = 2 * \frac{(Recall * Precision)}{(Recall + Precision)}$$

2.7. *Receiver Operating Characteristic* (ROC)

Kurva *Receiver Operating Characteristic* (ROC) mengekspresikan sebuah confusion matrix serta memperlihatkan nilai akurasi dan melakukan perbandingan klasifikasi secara visual. Kurva *Receiver Operating Characteristic* (ROC) ini merupakan grafik dua sisi dengan sisi pertama adalah sisi *false positives* sebagai garis horizontalnya dan sisi lainnya adalah sisi *true positives* untuk menghitung perbedaaan *performance* metode yang digunakan. Dapat dikatakan bahwa *Kurva Receiver Operating Characteristic* (ROC) merupakan pilihan lain untuk memeriksa sebuah kinerja pengklasifikasian [21]

3. METODE PENELITIAN

Pada bagian ini akan dijabarkan mengenai data yang digunakan pada penelitian serta tahapan atau proses penelitian klasifikasi *website psishing* seperti berikut:



Gambar 4. Diagram alur tahapan penelitian

3.1. *Dataset*

Dataset yang dijadikan objek dalam penelitian ini adalah data *website phishing detector* yang didapat dari data public dengan url https://www.kaggle.com/eswarchandt/phishing-*website*-detector. *Dataset* tersebut berisi kumpulan URL situs web untuk 11000+ situs web. Setiap sampel memiliki 30 parameter situs web dan label kelas yang mengidentifikasinya sebagai situs *website phishing* atau tidak (1 atau -1).

3.2. *Preprocessing* Data

Preprocessing data dilakukan sebelum tahap prediksi dimulai. *Preprocessing data* merupakan langkah penting yang perlu dilakukan adapaun tahapannya seperti melakukan pengisian data yang kosong, membuang data yang ganda atau *double*, memeriksa ketidakkonsistenan data, pembersihan data serta membenahi kesalahan yang mungkin terdapat pada data [22]

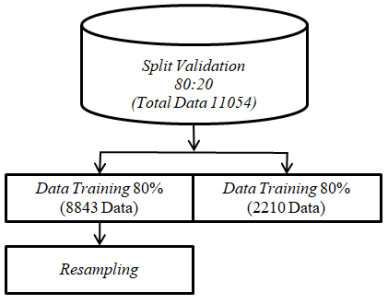
Salah satu proses *preprocessing* data yang dilakukan adalah data *cleaning* ini dijalankan untuk membersihkan data *missing value* atau *noise* pada data. Proses *cleaning* ini dilakukan dengan menggunakan perintah fillna() yang terintegrasi pada alat uji yang digunakan

Tools yang digunakan dalam penelitian ini adalah google *colaboration* menggunakan bahasa pemproraman Python serta mengimplementasikan model lightGBM serta menerapkan teknik *random oversampling* untuk mengatasi ketidakseimbangan pada kelas data.

3.3. *Split Validation*

Langkah selanjutnya setelah melakukan *preprocessing* adalah membagi data membentuk dua unit yaitu sebagai unit data *training* dan unit data *testing*. Data *training* ini digunakan untuk mengajarkan model sedangkan data *testing* digunakan untuk memverifikasi model yang telah dibangun.

Setelah melakukan *preprocessing* didapatkan Jumlah *instance* yang digunakan adalah sebanyak 11054 data, kemudian data di bagi menjadi 2 bagian, 80% data untuk mengajarkan model mengenal data dan 20% untuk menguji model apakah sudah betul dalam mengenali data, dalam penelitian ini didapatkan data *training* sebanyak 8843 data dan data testing sebanyak 2210 data.



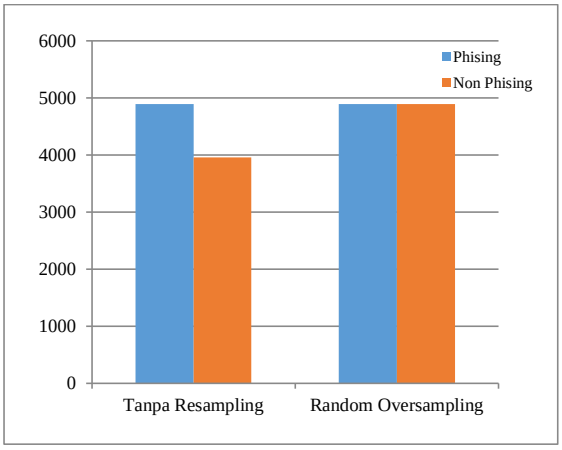
Gambar 5. Proses *Split Validation*

3.4. *Resampling*

Langkah selanjutnya setelah melakukan *preprocessing* adalah melakukan proses *resampling*. Proses *resampling* dilakukan untuk mengatasi masalah ketidakseimbangan kelas data (*imbalance data*) pada *Dataset* yang digunakan. Teknik *resampling* yang digunakan adalah teknik r*andom oversampling* (ROS).

Proses r*andom oversampling* (ROS) dilakukan dengan menambah data dari kelas minoritas ke dalam data *training* secara acak. Proses pemberian data ini berulang hingga jumlah data kelas *minoritas* rata dengan jumlah kelas *mayoritas*. Dengan menerapkan teknik r*andom oversampling* (ROS), meminimalisir ketidakseimbangan kelas data dengan membangkitkan data sintesis kelas minoritas dari

3955 menjadi 4888. Visualisasi jumlah data yang digunakan sebelum dan sesudah diterapkan teknik r*andom oversampling* (ROS) dapat dilihat pada gambar 6



Gambar 6. Penerapan r*andom oversampling* (ROS)

Untuk proses selanjutnya yaitu proses pengujian model lightGBM menggunakan data yang telah melalui tahap *resampling* dengan jumlah data masing-masing kelas sebanyak 4888.

3.5. Modelling

Dalam penelitian ini, metode yang diusulkan adalah model LightGBM untuk mengklasifikasikan sebuah *website phishing* disertai dengan teknik *resampling* yaitu *random oversampling* (ROS) untuk mengatasi ketidakseimbangan data. Pada model lightGBM digunakan fungsi *training* yang telah diubah ke bentuk matriks, maka dimasukkan data train_matrix. Kemudian menyesuaikan serangkaian model LightGBM untuk setiap *parameter*. *Parameter* yang digunakan dalam model LightGBM ini dapat dilihat pada table 2

Tabel 2. *Parameter* LightGBM *Classifier*

No	Parameter	Value
1.	Boosting type	gbdt
2.	Num_leaves	-1
3.	Max_depth	-1
4.	Learning rate	0.1
5.	N_estimator	100
6.	Subsample_for_bin	200000
7.	Objective	None
8.	Class_weight	None
9.	Min_split_gain	0.001
10.	Min_child_samples	1.0
11.	Silent	True
12.	Num_iteration	500

3.6. Evaluation

Evaluasi yang digunakan untuk menguji kinerja hasil klasifikasi adalah dengan *confusion matrics*. Dengan *confusion matrics* ini maka dapat diketahui kebenaran data prediksi terhadap data aktual. Serta dapat diperoleh nilai ROC, *Recall*, dan F1-*Score*, serta Akurasi.

4. HASIL DAN PEMBAHASAN

Untuk menunjukkan bahwa model yang direkomendasikan mempunyai kinerja yang signifikan dalam mengklasifikasikan *website phishing*, maka pada subbab ini akan ditampilkan hasil pengujian menggunakan beberapa model dan menggunakan teknik *resampling* yang sama. Perbandingan ini dilakukan untuk meyakinkan bahwa model yang diusulkan adalah model yang terbaik. Perbandingan beberapa metode tersebut diukur menggunakan kinerja metode klasifikasi yang meliputi akurasi, *recall*, *f1-score* dan ROC.

4.1. Hasil Pengujian Nilai Akurasi Pada Model

Untuk mengetahui rasio prediksi Benar (*phishing* dan *No Phishing*) dengan keseluruhan data, maka dapat kita lihat dari nilai akurasi yang dihasilkan dari masing-masing model yang diujikan. Pada tabel 3 akan ditampilkan hasil pengujian beberapa model untuk membuktikan bahwa model yang diusulkan adalah model yang paling cocok dengan data. Dari model yang telah diuji nilai akurasi tertinggi didapatkan dari model random forest dan lightGBM dengan nilai akurasi 96,9%.

Tabel 3. Perbandingan Hasil Pengujian Nilai Akurasi

Algoritma	Akurasi
Random Forest	96,90%
Gradient Boosting Classifier	92,80%
Decision Tree	91,50%
Naïve Bayes	88%
LightGBM	96,90%

4.2. Hasil Pengujian Nilai *Recall* Pada Model

Untuk mengetahui rasio prediksi berapa persen *website phishing* (*actual*) dibandingkan dengan keseluruhan *website* yang benar-benar melakukan phishing dapat dilihat dari nilai *recall*. Pada tabel 4 akan ditampilkan hasil pengujian beberapa model untuk membuktikan bahwa model yang diusulkan adalah model yang paling cocok dengan data. Dari model yang telah diuji nilai *recall* tertinggi didapatkan dari model random forest dan lightGBM dengan nilai akurasi 96,9%.

Tabel 4. Perbandingan Hasil Pengujian Nilai *Recall*

Algoritma	*Recall*
Random Forest	96,90%
Gradient Boosting Classifier	92,80%
Decision Tree	91,50%
Naïve Bayes	88%
LightGBM	96,90%

4.3. Hasil Pengujian Nilai F1-*Score* Pada Model

Untuk mengetahui rasio perbandingan rata-rata antara presisi dan *recall* yang dibobotkan dapat dilihat dari nilai f1-*score*. Pada tabel 5 akan

ditampilkan hasil pengujian beberapa model untuk membuktikan bahwa model yang diusulkan adalah model yang paling cocok dengan data. Dari model yang telah diuji nilai f1-*score* tertinggi didapatkan dari model random forest dan lightGBM dengan nilai akurasi 96,9%.

Tabel 5. Perbandingan Hasil Pengujian Nilai F1-*Score*

Algoritma	***Recall***
Random Forest	96,90%
Gradient Boosting Classifier	92,80%
Decision Tree	91,50%
Naïve Bayes	88%
LightGBM	96,90%

4.4. Hasil Pengujian Nilai ROC Pada Model

Untuk memberitahukan kepada kita seberapa baik model yang kita usulkan dapat membedakan antara dua hal (*web phishing* dan *non web phishing*) maka perlu dilihat dari kurva ROC, model dapat dikatakan baik jika dapat secara akurat membedakan anatar kedua kelas data. Pada tabel 6 akan ditampilkan hasil pengujian beberapa model untuk membuktikan bahwa model yang diusulkan adalah model yang paling akurat untuk membedakan kelas data. Dari model yang telah diuji nilai ROC tertinggi didapatkan dari model lightGBM dengan nilai akurasi 99,7%.

Tabel 6. Perbandingan Hasil Pengujian Nilai ROC

Algoritma	***Recall***
Random Forest	99,50%
Gradient Boosting Classifier	98,30%
Decision Tree	95,40%
Naïve Bayes	96,80%
LightGBM	99,70%

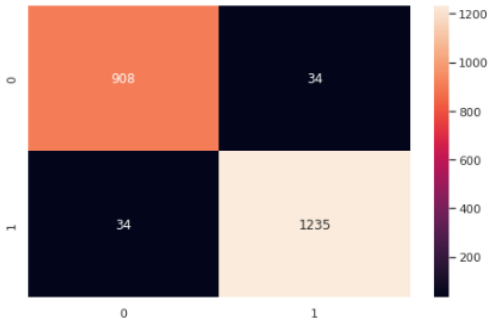
4.5. *Confussion matrix*

Representasi dari *confusion Matrics* dapat dilihat dari nilai *True positif* (TP), *False positif* (FP), *False Negatif* (FN) dan *True negative* (TN). *Confussion matrix* ini akan memperlihatkan serta membandingkan nilai sebenarnya (*actual*) dengan nilai yang diprediksi oleh model yang diusulkan sehingga peneliti dapat mengevaluasi nilai *Accuracy* (akurasi), *Recall*, dan *F1-Score* atau ROC. Pada tabel 7 akan ditampilkan hasil *confussion matrix* untuk membuktikan bahwa model yang diusulkan adalah model yang paling cocok dengan data.

Tabel 7. Perbandingan Hasil *Confussion matrix*

Algoritma	**TP**	**FP**	**FN**	**TN**
Random Forest	908	34	35	1234
Gradient Boosting	872	70	89	1180
Decision Tree	853	89	100	1169
Naïve Bayes	893	49	217	1052
LightGBM	908	34	34	1235

Dari *confussion matrix* dapat diakatakan bahwa model yang diusilkan yaitu model LightGBM dapat membedakan kelas data antara *web phishing* dan *non phishing* paling baik. Pada gambar 7 akan ditampilkan *confusion matrix* dari model lightGBM.



Gambar 7. *Confussion matrix* LightGBM

5. KESIMPULAN DAN SARAN

Untuk mengklasifikasikan anatar *website phishing* dan non *phishing*, penulis menggunakan model lightGBM, prediksi terbaik didapatkan dengan menerapkan parameter boosting type 'gbdt', n_estimators 100, max_depth -1, *num_leaves* 31, *leraning rate* 0,1, *num iteration* 500 dan *silent* bernilai *true*. untuk mengatasi ketidakseimbagan (*imbalance*) data menggunakan *random oversampling* (ROS).

Untuk megevaluasi model yang dibangun, Penulis menggunakan *performance metrics* eperti akurasi, *recall*, F1-*score*, and ROC *curve* dan hasil penelitian menyatakan bahwa model yang diusulkan lebih baik dari bebarapa model lain yang juga telah diuji dengan nilai akurasi sebesar 96,9%, *recal* 96,9%, F1-*score* 96,9% dan ROC 99,7%.

Pada eksperimen penelitian selanjutnya ada hal-hal yang perlu dicoba untuk ditambahkan agar menghasilkan *performance* yang lebih baik salah satunya adalah dengan menambah data amatan sehingga sebaran data dapat seimbang dan *representative*

DAFTAR PUSTAKA

[1] M. Al, K. Rizki, and A. F. Op, "Rancang Bangun Aplikasi E-Cuti Pegawai Berbasis *Website* (Studi Kasus : Pengadilan Tata Usaha Negara)," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 3, pp. 1–13, 2021, [Online]. Available: http://jim.teknokrat.ac.id/index.php/JTSI

[2] U. Rahardja, N. Lutfiani, and R. Rahmawati, "APTISI Student Perception to the News on The APTISI *Website*," *J. Ilm. SISFOTENIKA*, vol. 8, no. 2, pp. 117–127, 2018.

[3] A. S. Y. Irawan, N. Heryana, H. S. Hopipah, and D. Rahma, "Identifikasi *Website* Phishing dengan Perbandingan Algoritma Klasifikasi," *Syntax J. Inform.*, vol. 10, no. 01, pp. 57–67, 2021, doi: 10.35706/syji.v10i01.5292.

[4] M. H. Wibowo and N. Fatimah, "Ancaman

Phishing Terhadap Pengguna Sosial Media dalam Dunia Cyber Crime," *JoEICT (Journal Educ. ICT)*, vol. 1, no. 1, pp. 1–5, 2017, [Online]. Available: https://jurnal.stkippgritulungagung.ac.id/index.php/joeict/article/view/69

[5] N. B. Putri and A. W. Wijayanto, "Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi *Website* Phishing Comparative Analysis Of Data Mining Classification Algorithm In Phishing *Website* Classification," vol. 11, no. 28, pp. 59–66, 2022, doi: 10.34010/komputika.v11i1.4350.

[6] A. D. Alexander and T. S. Lestari, "Pengaruh Jumlah Hiden Layer Terhadap Performa Neural Network Dalam Prediksi *Website* Phishing," no. May, pp. 14–18, 2017.

[7] L. Henando, "Algoritma Apriori Dan Fp-Growth Untuk Analisa Perbandingan Data Penjualan Leptop Berdasarkan Merk Yang Diminati Konsumen (Studi Kasus : Indocomputer Payakumbuh)," *J-Click*, vol. 6, no. 2, pp. 201–207, 2019.

[8] D. S. O. Panggabean, E. Buulolo, and N. Silalahi, "Penerapan Data Mining Untuk Memprediksi Pemesanan Bibit Pohon Dengan Regresi Linear Berganda," *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 1, p. 56, 2020, doi: 10.30865/jurikom.v7i1.1947.

[9] N. I. Widiastuti, E. Rainarli, and K. E. Dewi, "Peringkasan dan Support Vector Machine pada Klasifikasi Dokumen," *J. Infotel*, vol. 9, no. 4, p. 416, 2017, doi: 10.20895/infotel.v9i4.312.

[10] H. Annur, "Klasifikasi Masyarakat Miskin Menggunakan Metode Naive Bayes," *Ilk. J. Ilm.*, vol. 10, no. 2, pp. 160–165, 2018, doi: 10.33096/ilkom.v10i2.303.160-165.

[11] M. C. Untoro, "MWMOTE optimization for imbalanced data using complete linkage," *J. Teknol. dan Sist. Komput.*, vol. 9, no. 2, pp. 77–82, 2021, doi: 10.14710/jtsiskom.2021.13748.

[12] A. Syukron and A. Subekti, "Penerapan Metode Random Over-Under Sampling dan Random Forest Untuk Klasifikasi Penilaian Kredit," *J. Inform.*, vol. 5, no. 2, pp. 175–185, 2018, doi: 10.31311/ji.v5i2.4158.

[13] A. Triyanto and R. Kusumaningrum, "Implementasi Teknik Sampling Untuk Mengatasi Imbalanced Data Pada Penentuan Status Gizi Balita Dengan Menggunakan Learning Vector Quantization Implementation Of Sampling Techniques For Solving Imbalanced Data Problem In Determination Of Toddler Nutritiona," vol. 19, p. Pp. 39–50, 2017.

[14] R. D. Fitriani, H. Yasin, and Tarno, "PENANGANAN KLASIFIKASI KELAS DATA TIDAK SEIMBANG DENGAN RANDOM OVERSAMPLING PADA NAIVE BAYES (Studi Kasus: Status Peserta KB IUD di Kabupaten Kendal)," vol. 10, pp. 11–20, 2021.

[15] Mushthofa, C. L. Abdulbaaqiy, S. H. Wijaya, M. A. Agmalaro, and L. S. Hasibuan, "Pemodelan berbasis jaringan untuk pengklasifikasian kanker payudara berdasarkan data molekuler," *J. Ilmu Komput. dan Agri-Informatika*, vol. 9, no. 1, pp. 101–113, 2022, doi: 10.29244/jika.9.1.101-113.

[16] P. S. Rizky *et al.*, "Perbandingan Metode LightGBM dan XGBoost dalam Menangani Data dengan Kelas Tidak Seimbang Universitas Hamzanwadi Selong," vol. 15, no. 2, pp. 228–236, 2022.

[17] A. S. Febriarini and E. Z. Astuti, "Penerapan Algoritma C4.5 untuk Prediksi Kepuasan Penumpang Bus Rapid Transit (BRT) Trans Semarang," *Eksplora Inform.*, vol. 8, no. 2, pp. 95–103, 2019, doi: 10.30864/eksplora.v8i2.156.

[18] R. Rusliyawati, K. Muludi, A. Wantoro, and D. A. Saputra, "Implementasi Metode International Prostate Symptom Score (IPSS) Untuk E-Screening Penentuan Gejala Benign Prostate Hyperplasia (BPH)," *J. Sains dan Inform.*, vol. 7, no. 1, pp. 28–37, 2021, doi: 10.34128/jsi.v7i1.298.

[19] H. Hozairi, A. Anwari, and S. Alim, "Implementasi Orange Data Mining Untuk Klasifikasi Kelulusan Mahasiswa Dengan Model K-Nearest Neighbor, Decision Tree Serta Naive Bayes," *Netw. Eng. Res. Oper.*, vol. 6, no. 2, p. 133, 2021, doi: 10.21107/nero.v6i2.237.

[20] I. W. Saputro and B. W. Sari, "Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa," *Creat. Inf. Technol. J.*, vol. 6, no. 1, p. 1, 2020, doi: 10.24076/citec.2019v6i1.178.

[21] S. Hendrian, "Algoritma Klasifikasi Data Mining Untuk Memprediksi Siswa Dalam Memperoleh Bantuan Dana Pendidikan," *Fakt. Exacta*, vol. 11, no. 3, pp. 266–274, 2018, doi: 10.30998/faktorexacta.v11i3.2777.

[22] Y. Pristyanto, "Penerapan Metode Ensemble Untuk Meningkatkan Kinerja Algoritme Klasifikasi Pada Imbalanced Dataset," *J. Teknoinfo*, vol. 13, no. 1, p. 11, 2019, doi: 10.33365/jti.v13i1.184.