Komputika: Jurnal Sistem Komputer
Volume 11, Nomor 1, April 2022, hlm. 59 - 66
Terakreditasi Peringkat 3, SK No. 28/E/KPT/2019
DOI: **10.34010/komputika.v11i1.4350**

ISSN: 2252-9039 (print)
ISSN: 2655-3198 (online)

Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing

Nabila Bianca Putri[1*], Arie Wahyu Wijayanto[2]

[1,2)]Program Studi Komputasi Statistik, Politeknik Statistika STIS
Jl. Otto Iskandardinata No. 64C
**email: 221810484@stis.ac.id*

(Naskah masuk: *2 April 2021*; diterima untuk diterbitkan: *26 April 2022*)

ABSTRAK – *Phishing adalah tindakan penipuan yang dilakukan untuk mencoba mendapatkan informasi penting dari user yang menggunakan internet dengan mengirim sejumlah e-mail palsu kepada para user. Teknik klasifikasi data mining dapat digunakan untuk prediksi website phishing. Banyak algoritma klasifikasi di dalam data mining yang dapat digunakan, sehingga perlu dilakukan komparasi setiap algoritma untuk mengetahui tingkat akurasi dari masing-masing algoritma. Beberapa algoritma klasifikasi di dalam data mining yang banyak digunakan antara lain Naïve Bayes, Random Forest, Decision Tree, dan Support Vector Machine. Pada penelitian ini, peneliti akan membandingkan akurasi, presisi, dan sensitivitas dari tiap-tiap algoritma tersebut untuk menemukan algoritma terbaik dalam mengklasifikasi website phishing. Data yang digunakan dalam penelitian ini adalah sebanyak 1.353 data website yang terdiri dari 702 data situs bukan phishing, 103 data situs mencurigakan, dan 548 data situs phishing. Hasil dari proses klasifikasi dievaluasi dengan menggunakan cross validation dan confusion matrix untuk mengetahui algoritma klasifikasi data mining yang paling akurat untuk prediksi website phishing.*

Kata Kunci *– Website Phishing; Naïve Bayes; Random Forest; Decision Tree; Support Vector.*

Comparative Analysis Of Data Mining Classification Algorithm In Phishing Website Classification

ABSTRACT *– Phishing is a fraudulent activity carried out to try to get important information from users who use the internet by sending a few fake e-mails to the users. Data mining classification techniques can be used to predict phishing websites. Many classification algorithms in data mining can be used, so it is necessary to compare each algorithm to determine the level of accuracy of each algorithm. Several classification algorithms in data mining that are widely used include Naïve Bayes, Random Forest, Decision Tree, and Support Vector Machine. In this research, researchers will compare the accuracy, precision, and sensitivity of each of these algorithms to find the best algorithm for classifying phishing websites. The data used in this research were 1,353 website data consisting of 702 non-phishing site data, 103 suspicious site data, and 548 phishing site data. The results of the classification process are evaluated using cross-validation and confusion matrix to determine the most accurate data mining classification algorithm for predicting phishing websites.*

Keywords *- Website Phishing; Naïve Bayes; Random Forest; Decision Tree; Support Vector.*

1. PENDAHULUAN

Seiring berkembangnya zaman, teknologi informasi pun ikut berkembang pesat. Hal ini dibuktikan dengan adanya internet yang saat ini sangat mudah diakses oleh masyarakat. Salah satu contoh penggunaan internet yang sering kita jumpai adalah media sosial. Dengan adanya media sosial, masyarakat semakin mudah dalam berkomunikasi, mencari teman, dan mengetahui hal-hal yang sedang *trending* di sekitarnya. Tidak hanya itu, masyarakat pun dapat berbisnis *online* melalui media sosial. Namun di balik semua keuntungan tersebut ada pihak yang tidak bertanggung jawab yang melakukan tindakan merugikan banyak orang, salah satunya adalah tindakan *phishing*.

Phishing merupakan upaya untuk mendapatkan suatu informasi penting dan bersifat rahasia secara ilegal, seperti *user id, password*, PIN, informasi rekening bank, informasi kartu kredit, atau informasi rahasia yang lainnya[1]. Sedangkan situs *phishing* merupakan sebuah situs yang didesain sedemikian rupa oleh penjahat internet dengan menyerupai situs aslinya mulai dari tampilan, konten, URL domain dan sejenisnya untuk mengelabui korban (pengguna internet) dengan membuat korban seolah-olah sedang mengakses halaman situs dari sumber yang sah. Pada penelitian ini, dikumpulkan sebuah data yang akan diperlukan untuk mengidentifikasi *website phishing*, kemudian menganalisa data tersebut dengan menggunakan data mining. Data mining merupakan sebuah konsep untuk mengenali pola yang tersembunyi dan menemukan relasi antar parameter didalam data dengan jumlah yang besar[2]. Pada data ini telah diindentifikasikan fitur yang berbeda-beda yang terkait dengan *website* yang sah dan phishy serta mengumpulkan 1.353 *website* yang berbeda dari sumber yang berbeda. *Website phishing* dikumpulkan dari arsip data Phishtank (www.phishtank.com)[3], yang merupakan situs komunitas gratis tempat pengguna dapat mengirimkan, memverifikasi, melacak dan berbagi data *phishing*.

Ada beberapa fungsionalitas dari data mining, antara lain analisis asosiasi antar data, klasifikasi data, klastering data dan lain-lain. Dalam penelitian ini, fungsionalitas yang dipakai adalah klasifikasi data. Klasifikasi data adalah proses menemukan model atau fungsi yang menjelaskan dan membedakan kelas data serta konsepnya. Peneliti akan melakukan komparasi beberapa algoritma untuk menemukan akurasi tertinggi. Algoritma yang digunakan pada penilitian ini adalah *naïve bayes, random forest, decision tree,* dan *support vector machine*.

2. METODE DAN BAHAN

2.1 *Phising* dan Penelitian Sebelumnya

Phishing adalah suatu taktik penipuan dengan mengelabui target untuk mencuri informasi dari akun korban[4]. Istilah ini berasal dari kata *fishing* yang artinya memancing korban agar terperangkap kedalam jebakan pelaku. Pada dasarnya *phishing* didefinisikan sebagai tindak penipuan yang memanfaatkan *email* dari pengguna untuk menggali informasi sensitif milik korban[4].

Tabel 1 menunjukkan daftar penelitian lain yang telah dilakukan sebelumnya. Secara umum penelitian tersebut relevan dengan kasus yang diambil oleh peneliti saat ini.

Tabel 1. Penelitian Klasifikasi Web Phishing

No.	Judul Penelitian	Penulis	Hasil
1	Perbandingan Klasifikasi Algoritma K-Nn, Neural Network, Naïve Bayes, C 4.5 untuk Mendeteksi Web Phishing	Eza Nanda, Istikoma, Nurindah A. Amari, Yoga Pristyanto	Nilai akurasi tertinggi adalah algoritma C 4.5 yaitu sebesar 89,66% , diikuti oleh 88,92% Neural Network, 87,08% KNN, dan 83,89% Naive Bayes[5].
2	Prediksi Website Pemancing Informasi Penting Phishing Menggunakan Support Vector Machine (SVM)	Zuhri Halim	Nilai akurasi tertinggi adalah algoritma metode Decision Tree yaitu sebesar 91,84%, lalu diikuti oleh metode Naïve Bayes sebesar 74,07% dan Support Vector Machine sebesar 92,34%[6].
3	Penerapan Algoritma Naïve Bayes Classifier Untuk Meningkatkan Keamanan Data Dari Website Phishing	Agus Fatkhurohman, Eli Pujastuti	Akurasi algoritma Naive Bayes yang didapatkan adalah sebesar 92.98%[7].

2.2 *Naïve Bayes*

Naïve Bayes merupakan salah satu penerapan teorema Bayes. *Naïve Bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu[8]. Untuk mendapatkan nilai probabilitas pada sebuah sampel diberikan sebuah teorema *Bayes,* seperti pada persamaan 1:

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

Dimana P(H) adalah nilai probabilitas prior dari

hipotesis pada sebuah sampel, biasa disebut dengan priori. P(X) merupakan *evidence* dari probabilitas data pelatihan. P(H|X) adalah nilai probabilitas H yang mempengaruhi X (*posterior density*), sedangkan P(X|H) merupakan probabilitas x kepada h yang disebut dengan *likelihood*.

Untuk menjelaskan teorema *Naive Bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Oleh karena itu, teorema *Bayes* di atas disesuaikan seperti persamaan 2.

$$P(C|F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)} \quad (2)$$

Dimana Variabel C merepresentasikan kelas, sementara variabel F1 ... Fn merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (*Posterior*) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik karakteristik sampel pada kelas C (disebut juga *likelihood*), dibagi dengan peluang kemunculan karakteristik sampel secara global (disebut juga *evidence*). Karena itu, persamaan 2 dapat pula ditulis secara sederhana seperti persamaan 3

$$Posterior = \frac{Prior \times likelihood}{evidence} \quad (3)$$

Dari persamaan diatas dapat disimpulkan bahwa asumsi independensi naif tersebut membuat syarat peluang menjadi sederhana, sehingga perhitungan menjadi mungkin untuk dilakukan. Selanjutnya, penjabaran $P(C|F_1 \dots F_n)$ dapat disederhanakan menjadi persamaan 4 dan 5.

$$P(C|F_1 \dots F_n) = P(C)P(F_1|C)P(F_2|C) \dots \quad (4)$$

$$P(C|F_1 \dots F_n) = P(C) \prod_{i=1}^{n} P(F_i|C) \quad (5)$$

Persamaan 4 dan 5 merupakan model dari teorema *Naive Bayes* yang selanjutnya akan digunakan dalam proses klasifikasi. Untuk klasifikasi dengan data kontinyu digunakan rumus Densitas Gauss, seperti persamaan 6.

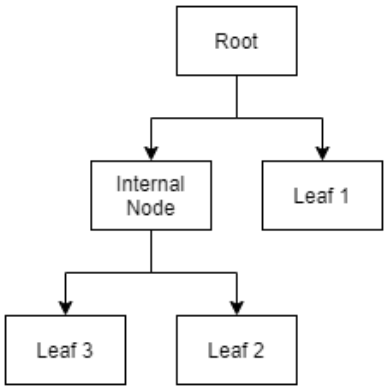
$$P(X_i = x_i|Y = y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} e^{-\frac{(x_i-\mu_i)^2}{2\sigma_{ij}^2}} \quad (6)$$

Keterangan :
P : Peluang
X_i : Atribut ke i
x_i : Nilai atribut ke i
Y : Kelas yang dicari
y_i : Sub kelas Y yang dicari
π : Mean, menyatakan rata rata dari seluruh atribut
σ : Deviasi standar, menyatakan varian dari seluruh atribut

2.3 *Decision Tree*

Decision tree adalah sebuah struktur pohon, dimana setiap *node* internal (*non-leaf*) merepresentasikan pengujian atribut, setiap cabang merupakan suatu pembagian hasil uji, dan *node* daun (*leaf*) merepresentasikan kelompok kelas tertentu[9]. Tingkat *node* teratas dari sebuah decision tree adalah node akar (*root*) yang biasanya berupa atribut yang paling berpengaruh pada suatu kelas tertentu. Pada umumnya *Decision Tree* melakukan strategi pencarian secara *top-down* untuk solusinya. Pada proses mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari *node* akar (*root*) sampai *node* akhir (*leaf*) dan kemudian akan diprediksi kelas yang dimiliki oleh suatu data baru tertentu. *Decision Tree* dapat dengan mudah dirubah menjadi aturan klasifikasi, dapat terlihat pada gambar 1.

Decision tree merupakan salah satu teknik klasifikasi data mining yang paling populer. *Decision tree* sesuai digunakan untuk kasus yang memiliki ciri-ciri sebagai berikut[10]: 1. Data atau contoh dinyatakan dengan pasangan atribut dan nilainya; 2. Label atau output data biasanya bernilai diskrit; 3. Data mempunyai *missing value*



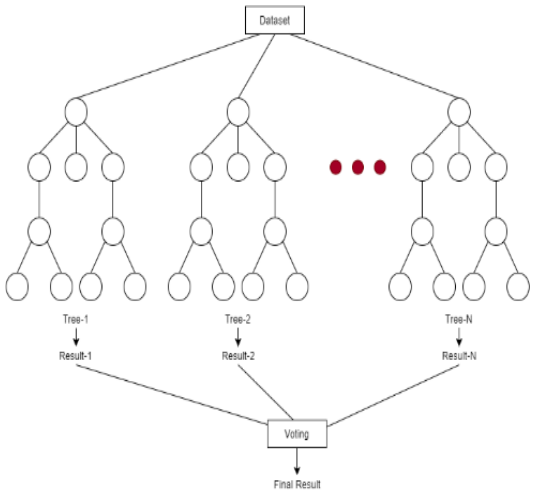
Gambar 1. *Decision Tree*

2.4 *Random Forest*

Metode *Random Forest* adalah pengembangan dari metode CART, yaitu dengan menerapkan metode *bootstrap aggregating* (*bagging*) dan *random feature selection*[11]. Dalam *Random Forest*, banyak pohon ditumbuhkan sehingga terbentuk hutan (*forest*), kemudian analisis dilakukan pada kumpulan pohon tersebut. Pada gugus data yang terdiri atas n amatan dan p peubah penjelas, *Random Forest* dilakukan

dengan cara: 1) Lakukan pengambilan contoh acak berukuran n dengan pemulihan pada gugus data. Tahapan ini merupakan tahapan bootstrap. 2) Dengan menggunakan contoh bootstrap, pohon dibangun sampai mencapai ukuran maksimum (tanpa pemangkasan). Pada setiap simpul, pemilihan pemilah dilakukan dengan memilih m peubah penjelas secara acak, dimana m << p. Pemilah terbaik dipilih dari m peubah penjelas tersebut. Tahapan ini adalah tahapan *random feature selection*. 3) Ulangi langkah 1 dan 2 sebanyak k kali, sehingga terbentuk sebuah hutan yang terdiri atas k pohon.

Respons suatu amatan diprediksi dengan menggabungkan (*aggregating*) hasil prediksi k pohon. Pada masalah klasifikasi dilakukan berdasarkan *majority vote* (suara terbanyak). *Error* klasifikasi *random forest* diduga melalui *error* OOB yang diperoleh dengan cara: 1) Lakukan prediksi terhadap setiap data OOB pada pohon yang bersesuaian. Data OOB (*out of bag*) adalah data yang tidak termuat dalam contoh bootstrap. 2) Secara rata-rata, setiap amatan gugus data asli akan menjadi data OOB sebanyak sekitar 36% dari banyak pohon. Oleh karena itu, pada langkah 1, masing-masing amatan gugus data asli mengalami prediksi sebanyak sekitar sepertiga kali dari banyaknya pohon. Jika a adalah sebuah amatan dari gugus data asli, maka hasil prediksi *Random Forest* terhadap a adalah gabungan dari hasil prediksi setiap kali a menjadi data OOB. 3) Error OOB dihitung dari proporsi misklasifikasi hasil prediksi *Random Forest* dari seluruh amatan gugus data asli,seperti terlihat pda gambar 2.



Gambar 2. *Random Forest*

2.5 *Support Vector Machine* (SVM)

Support Vector Machine adalah sebuah metode seleksi dengan membandingkan parameter standar seperangkat nilai diskrit yang disebut kandidat set, dan mengambil salah satu yang memiliki akurasi klasifikasi terbaik[12]. *Support Vector Machine* (SVM) adalah pengklasifikasi linier berdasarkan prinsip memaksimalkan *margin*[13]. SVM menggunakan *hyperplane* secara optimal untuk mengklasifikasikan data menjadi dua kelompok data dalam ruang dimensi yang lebih tinggi[14]. Margin adalah jarak antara *hyperplane* dan data terdekat dari setiap kelas[13]. Data terdekat disebut vektor dukungan. *Hyperplane* adalah pemisah terbaik antara dua kelas yang telah ditentukan. Prinsip dasar SVM adalah pengklasifikasi linier kemudian dikembangkan untuk mengerjakan soal-soal non linier[15]. Dengan memasukkan konsep trik *kernel* dalam ruang kerja berdimensi tinggi. *Kernel* SVM yang digunakan dalam penelitian ini adalah *kernel* RBF untuk proses transformasi dari ruang input menjadi ruang fitur.

Metode SVM memiliki konsep sentral dalam mengklasifikasikan data, yaitu mencari *hyperplane* terbaik untuk memisahkan antara dua kelas yang telah ditentukan. *Hyperplane* terbaik diperoleh dengan memaksimalkan vektor dukungan margin. Proses memaksimalkan margin vektor pendukung dapat dilakukan dengan meminimalkan lagrangian dan direduksi menjadi w dan b yang terdapat pada persamaan 7.

$$Lp = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i (w.x_i + b) - 1 \quad (7)$$

$$w = \sum_{i=1}^{N} \alpha_i y_i \quad b = y_i - w.x_i$$

Karena nilai α tidak diketahui, maka nilai w dan b tidak dapat ditentukan. Nilai α dicari dengan memaksimalkan pengali Lagrangian dengan kondisi optimal untuk dualitasnya menggunakan batasan Karush-Kuhn-Tucker (KKT). Penggunaan batasan KKT membuat nilai *Lagrange multiplier* (α) sama dengan jumlah data latih. Proses memaksimalkan pengali Lagrangian masih memiliki banyak kemungkinan nilai w, b, dan α. Berdasarkan permasalahan tersebut, maka proses maksimalisasi pengali Lagrange harus ditransformasikan menjadi dualitas pengali *Lagrange* pada persamaan 8 dengan batasan 1 dan 2.

$$Maks\ Ld = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \quad (8)$$

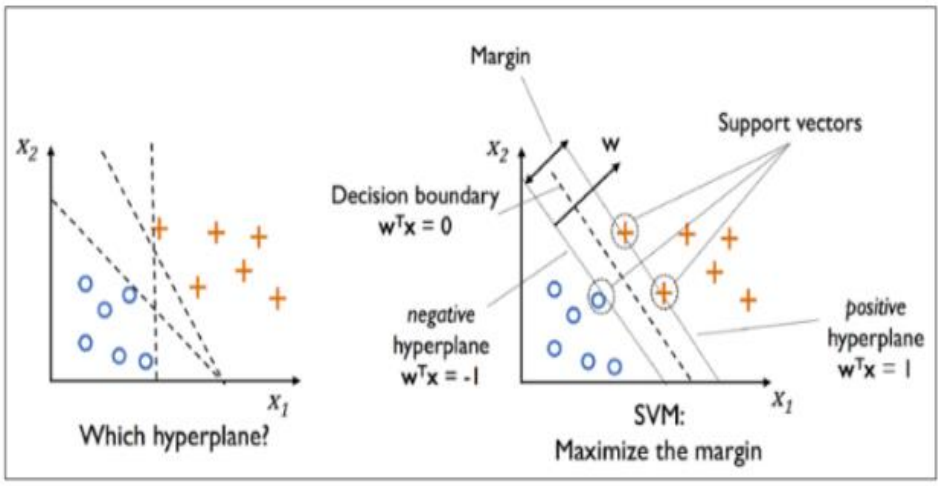
$$\sum_{i=1}^{N} \alpha_i y_i = 0; \quad 0 \le \alpha_i \le C; \quad i = 1,2,\dots,N$$

Setelah diperoleh nilai w, b, dan α, selanjutnya dilakukan penentuan label menggunakan model SVM.

$$F(\Phi(x)) = sign\ (w.\Phi(x) + b) \quad (9)$$

Jika nilai f(x) yang dihasilkan adalah f(x) > 0 maka data diklasifikasikan ke dalam kelas positif (+1), jika f(x) < 0 maka data tersebut diklasifikasikan ke dalam

kelas negatif (-1).

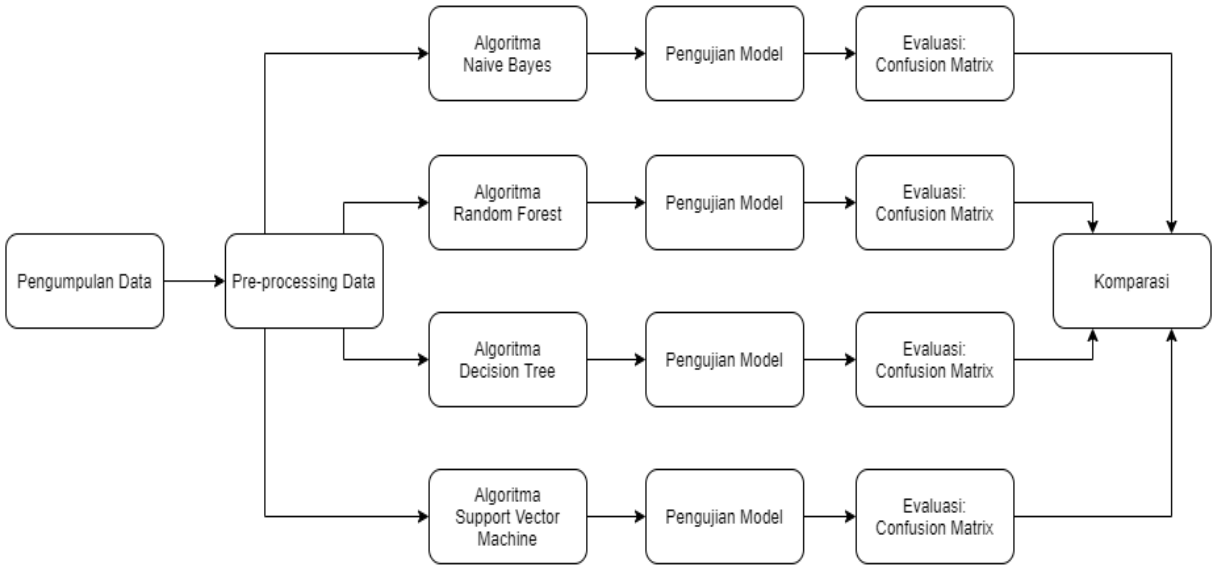


Sumber : https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781787125933/3/ch03lvl1sec21/maximum-margin-classification-with-support-vector-machines

Gambar 3. Support Vector Machine [21]

2.6 METODE

Diagram Alur Tahapan Klasifikasi Website Phishing, dapat dilihat pada gambar 4.



Gambar 4. Diagram Alur Tahapan Klasifikasi Website Phishing

2.6.1 Pengumpulan Data

Penelitian ini menggunakan data *website* sebanyak 1.353 data yang terdiri dari 702 data situs bukan *phishing*, 103 data situs mencurigakan, dan 548 data situs *phishing*. Dataset ini memiliki 10 atribut seperti yang dijelaskan pada tabel 2.

2.6.2 Pengujian Model dan Evaluasi

1. *Cleaning Data*

Cleaning data adalah proses menghilangkan *noise* dan data yang tidak konsisten agar dapat dianalisis[16]. Data yang diubah tersebut adalah data yang salah, rusak, tidak akurat, tidak lengkap, atau salah format. Hal ini bertujuan untuk meningkatkan kualitas data tersebut.

2. *Split Validation*

Split Validation merupakan teknik validasi dengan cara membagi data secara acak menjadi dua bagian yaitu *training data* dan *testing data*[17]. *Training data* merupakan suatu data yang sudah terklasifkasi lalu diolah untuk menemukan sutau pola. Dari *training data*, dengan menggunakan metode tertentu akan diperoleh suatu model klasifikasi yang kemudian akan digunakan untuk penentuan kelas terhadap *testing data*[3].

Tabel 2. Atribut Dataset Website Phishing

No	Atribut	Keterangan
1	SFH	Domain pemrosesan Server Form Handler (polinominal). Nilai : -1 , 0, 1
2	popUpWindow	Penggunaan popUpWindow untuk meminta user mengisi data mereka (biner). Nilai : -1 , 0, 1
3	SSLfinal_State	Memiliki sertifikat SSL dimana sertifikat yang dipercaya berasal dari penyedia ternama (polinominal). Nilai : -1 , 0, 1
4	Request_URL	Persentase permintaan url ekternal dari keseluruhan (polinominal). Nilai : -1 , 0, 1
5	URL_of_anchor	Persentase penggunaan tag yang mengarah selain ke domain yang sama dari keseluruhan (polinominal). Nilai : -1 , 0, 1
6	Web_traffic	Rank lalu lintas website dalam basis data Alexa (polinominal). Nilai : -1 , 0, 1
7	URL_Length	Panjang url (polinominal). Nilai : -1 , 0, 1
8	age_of_domain	Umur domain (biner). Nilai : -1 , 1
9	having_IP_Address	Adanya IP address sebagai domain pada url (biner). Nilai : 0, 1
No	**Atribut**	**Keterangan**
10	Result	Hasil identifikasi website (biner). Nilai : -1 , 0, 1

3. *Confusion Matrix*

Evaluasi model klasifikasi didasarkan pada pengukuran terhadap kinerja dari model klasifikasi untuk menggambarkan seberapa baik sistem dalam mengklasifikasikan data[18]. Salah satu metode yang dapat digunakan untuk mengukur kinerja model klasifikasi adalah dengan *confusion matrix*[18]. *Confusion Matrix* mengandung suatu informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya[19]. Setiap sel berisi angka yang menunjukkan berapa banyak kasus yang sebenarnya dari kelas yang diamati untuk diprediksi. *Confusion Matrix* dapat dilihat pada tabel 3.

Hasil klasifikasi dapat dihitung tingkat akurasinya berdasarkan kinerja matriks. Untuk menghitung tingkat akurasi pada matriks[20] digunakan:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (9)$$

$$Sensitivity = TP_{rate} = \frac{TP}{TP+FN} \times 100\% \quad (10)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (11)$$

Tabel 3. Confusion Matrix

	Terklasifikasi Positif	**Terklasifikasi Negatif**
Positif	TP *(True Positive)*	FN *(False Negative)*
Negatif	FP *(False Positive)*	TN *(True Negative)*

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, peneliti membagi dataset menjadi dua bagian dengan proporsi 80% data sebagai *training data* dan 20% data sebagai *testing data*. Dengan demikian, terdapat 1093 *website* sebagai *training data* dan 260 *website* lainnya sebagai *testing data*. Peneliti menggunakan software RStudio untuk mengklasifikasikan *website phising*.

Tabel 4 merupakan rangkuman hasil komparasi dari pengujian model yang telah dilakukan. Peneliti mengambil tiga variabel sebagai alat ukur dalam komparasi keempat model yang telah didapatkan. Ketiga variabel tersebut adalah akurasi, presisi, dan sensitivitas.

Tabel 4. Hasil Pengujian Model

	NB	**RF**	**DT**	**SVM**
Accuracy	82,31%	90,77%	85,77%	86,25%
Precision	82,54%	87,90%	78,42%	83,58%
Sensitivity	91,23%	95,61%	95,61%	92,88%

Dari tabel diatas dapat kita ketahui bahwa algoritma *Random Forest* memiliki akurasi tertinggi diantara model algoritma lain yaitu sebesar 90,77%. Tidak hanya akurasi, tetapi presisi dan sensitivitas dari algoritma *Random Forest* juga memiliki nilai tertinggi dari model algoritma lainnya.

4. KESIMPULAN

Dari hasil penelitian dengan dataset *website phishing* sebanyak 1.353 data dan terdiri dari 10 variabel yang kemudian dievaluasi dengan *confusion matrix*, didapatkan bahwa algoritma *Random Forest* menghasilkan model dengan nilai akurasi yang lebih baik dari pada algrotima *Naïve Bayes, Decision Tree,* dan *Support Vector Machine* yaitu sebesar 90,77%. Sedangkan algoritma *Naïve Bayes* menghasilkan model yang memiliki nilai akurasi terendah yaitu sebesar 82,31%. Presisi dan sensitivitas yang dimiliki model dari algoritma *Random Forest* juga memiliki nilai tertinggi daripada algoritma lainnya, dengan masing-masing bernilai 87,90% dan 95,61%. Dengan demikian model klasifikasi sudah tepat digunakan untuk mengklasfikasi *website phishing* untuk mencegahan pencurian data dari sebuah ancaman *website phishing*.

Untuk penelitian selanjutnya dapat dilakukan dengan kasus yang sama namun dengan algoritma lainnya agar dapat menemukan algoritma yang lebih baik.

DAFTAR PUSTAKA

[1] P. Febry Eka, "Model Klasifikasi Untuk Deteksi Situs Phishing Di Indonesia". *Masters thesis. Institut Teknologi Sepuluh Nopember*, 2017.

[2] P. Kaur, M. Singh, and G. S. Josan. "Classification and Prediction Based Data Mining Algorithms to Predict Slow Learners in Education Sector". *Procedia Computer Science*, 2015.

[3] A. Indriani, "Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier". *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Vol 1, no.1, hal. G-5 – G-10, 2014.

[4] S. Azani Cempaka. "PengenalanTeknologi Informasi : Mengenal Apa itu Phishing Penyebab, dan Mengatasinya", [Online]. https://socs.binus.ac.id/2018/11/29/pengenalanteknologi-informasi-mengenal-apa-itu-phishing-penyebab-dan-mengatasinya/#:~:text=Phishing%20adalah%20suatu%20metode%20untuk,akun%20korban%20untuk%20maksud%20tertentu. [Diakses 30 Desember 2020].

[5] E. Nanda, I. Istikomah, N.A. Amari, Y. Pristyanto, "Perbandingan Klasifikasi Algoritma K-Nn, Neural Network, Naïve Bayes, C 4.5 untuk Mendeteksi Web Phishing". *Jurnal Teknologi Informasi dan Ilmu Komputer : FAHMA,* Vol. 16 (3), hal. 33-42, 2018.

[6] Z. Halim, "Prediksi Website Pemancing Informasi Penting Phishing Menggunakan Support Vector Machine (SVM)". *Information System For Educators And Professionals,* Vol. 2 (1), hal . 71-82, 2017.

[7] A. Fatkhurohman, E. Pujastuti, "Penerapan Algoritma Naïve Bayes Classifier Untuk Meningkatkan Keamanan Data Dari Website Phishing". *Jurnal Teknologi Informasi*, Vol. 14 (1), hal. 115 – 124, 2019.

[8] A. Primajaya, B.N. Sari, "Random Forest Algorithm for Prediction of Precipitation. *Indonesian Journal of Artificial Intelligence and Data Mining*, *1*(1), 27-31, 2018.

[9] N.Frastian, S. Hendrian., V.H. Valentino, "Komparasi Algoritma Klasifikasi Menentukan Kelulusan Mata Kuliah Pada Universitas". *Faktor Exacta,* Vol.11 (1). 65-74, 2018.

[10] Santoso, Budi. "Data Mining Teknik Pemanfaatan Data Untuk Keperluan Bisnis, 1st ed", Yogyakarta : Graha Ilmu, 2017.

[11] N.K. Dewi, U.D. Syafitri, S.Y. Mulyadi, "Penerapan Metode Random Forest dalam Driver Analysis". *Forum Statistika dan Komputasi*. Vol.16 (1). 35-43, 2011.

[12] Y. Dong, Z. Xia, M. Tu, and G. Xing, "An Optimization Method For Selecting Parameters In Support Vector Machines". *In Sixth International Conference on Machine Learning and Applications (ICMLA 2007), IEEE,* pp. 1-6, 2007.

[13] A. Kowalczyk. "Support Vector Machines Succinctly". Book, vol. Alexandre, 2017.

[14] R. Wijayanti and A. Arisal. "Ensemble Approach for Sentiment Polarity Analysis in User-Generated Indonesian Text". *International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, IEEE, pp. 158-163, 2017.

[15] F. Xhafa, S. Patnaik and M. Tavana. "Advances in Intelligent, Interactive Systems and Applications". *Proceedings of the 3rd International Conference on Intelligent, Interactive Systems and Applications (IISA2018), Spinger*, Vol. 885, 2019.

[16] F. Marisa. "Educational Data Mining (Konsep Dan Penerapan)". *Jurnal Teknologi Informasi: Teori, Konsep, dan Implementasi* , Vol.4(2), Hal. 90-97, 2013.

[17] E. Fajrila. "Perbandingan Klasifikasi Ketepatan Waktu Kelulusan Mahasiswa Menggunakan Regresi Logistik Biner Dan Naïve Bayes Classifier". *Skripsi Universitas Islam Indonesia*, 2018.

[18] N. Hadianto, H.B. Novitasari, dan A. Rahmawati. "Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network".

Jurnal PILAR Nusa Mandiri, Vol. 15(2), hal. 163 – 170, 2019.

[19] Prasetyo, E., " Data Mining Konsep dan Aplikasi Menggunakan Matlab". Penerbit Andi Offset, Yogyakarta, 2012.

[20] Nugroho, Kuncahyo Setyo. "Confusion Matrix untuk Evaluasi Model pada Supervised Learning", [Online]. https://medium.com/@ksnugroho/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f. [Diakses 30 Desember 2020]

[21] Packt. "Maximum margin classification with support vector machines", [Online]. https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781787125933/3/ch03lvl1sec21/maximum-margin-classification-with-support-vector-machines. [Diakses 02 Januari 2021] .