DETEKSI *PHISHING WEBSITE* MENGGUNAKAN *SUPPORT VECTOR MACHINE*, *GRADIENT BOOSTING*, DAN *NEURAL NETWORKS*

Vanyariska Indriani [1], Hersatoto Listiyono [2], Saefurrahman [3]
[1,3] Teknik Informatika, Universitas Stikubank
Jl. Tri Lomba Juang, Mugassari Semarang, Indonesia
[2] Teknologi Rekayasa Multimedia Grafis, Universitas Stikubank
JL. Kendeng, No. 5, Bendan Ngisor Semarang, Indonesia
vanyariskaindriani@gmail.com

ABSTRAK

Di era *digital*, *phishing* menjadi ancaman yang signifikan terhadap keamanan data pribadi pengguna internet. *Website phishing* yang meniru tampilan situs asli bertujuan untuk memperoleh informasi sensitif seperti data pribadi dan keuangan. Penelitian ini bertujuan untuk menguji efektivitas tiga algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *Gradient Boosting*, dan *Neural Networks* dalam mendeteksi URL *phishing*. *Dataset* PhiUSIIL yang terdiri dari 235.795 sampel data digunakan, yang mencakup URL *phishing* dan legitim dengan 54 fitur. Metode yang diterapkan meliputi *preprocessing* data, pembagian data latih dan uji, serta normalisasi menggunakan MinMaxScaler. Ketiga model dievaluasi berdasarkan akurasi, *F1-Score*, *Confusion Matrix*, dan waktu komputasi. Hasil penelitian menunjukkan bahwa model *Gradient Boosting* memperoleh akurasi dan *F1-Score* 100%, dengan waktu komputasi tercepat hanya 1 menit, menjadikannya pilihan terbaik untuk deteksi *phishing*. Sementara itu, SVM dan *Neural Networks* juga menunjukkan hasil yang sangat baik, masing-masing dengan akurasi 99,99%, namun dengan waktu komputasi yang lebih lama. Berdasarkan hasil tersebut, penelitian ini menyimpulkan bahwa *Gradient Boosting* adalah model yang paling efisien untuk mendeteksi *phishing* pada *dataset* yang digunakan.

Kata kunci : *Phishing, Support Vector Machine, Gradient Boosting, Neural Networks, Machine learning,* Deteksi *Website*

1. PENDAHULUAN

Di era teknologi saat ini, internet telah menjadi bagian penting dalam kehidupan sehari-hari. Dengan meningkatnya kebutuhan akan informasi, internet memberikan kemudahan di berbagai bidang, seperti komunikasi, edukasi, belanja, dan bisnis [1].

Salah satu sarana utama untuk mengakses informasi adalah melalui *website*. *Website* adalah kumpulan halaman yang menyajikan informasi dalam berbagai format, seperti teks, gambar, suara, animasi, video, atau kombinasi dari semuanya. *Website* dapat bersifat statis maupun dinamis, di mana setiap halaman saling terhubung dalam satu rangkaian dan ditautkan melalui jaringan untuk memberikan informasi yang dibutuhkan oleh pengguna [2].

Pada awalnya, *website* hanya digunakan sebagai media penyampaian informasi satu arah. Namun, seiring kemajuan teknologi, *website* kini berkembang menjadi *platform* interaktif yang mendukung komunikasi, kolaborasi, hingga transaksi secara daring [3].

Meskipun *website* membawa banyak manfaat, penggunaannya juga menghadirkan risiko, terutama munculnya oknum yang tidak bertanggung jawab yang melakukan tindakan merugikan banyak pihak [4].

Hal ini disebabkan oleh pelaku kejahatan yang melihat peluang untuk memanfaatkan popularitas sebuah *website* dalam melakukan tindakan kriminal yang dikenal sebagai *phishing* [5].

Salah satu tindakan yang dapat merugikan adalah *phishing*. *Phishing* merupakan tindakan penipuan yang dilakukan melalui *website* atau email palsu untuk mengelabui pengguna agar memberikan data sensitif, seperti informasi pribadi (nama, alamat, tanggal lahir), akun pengguna (*username* dan *password*), hingga data keuangan (informasi kartu kredit atau rekening) [6].

Website phishing adalah jenis ancaman siber yang mencoba meniru tampilan *website* resmi sebuah perusahaan atau institusi dengan tujuan memperoleh data pribadi korban [7].

Tindakan *phishing* kian meningkat seiring dengan popularitas internet dan *website* sebagai sarana utama untuk bertransaksi dan berbagi informasi. Berdasarkan laporan dari IDADX pada kuartal kedua tahun 2024, terdapat 14.093 laporan penyalahgunaan domain, di mana kategori *phishing* mendominasi dengan 1.598 laporan [8].

Data ini menunjukkan bahwa *phishing* merupakan ancaman signifikan yang memerlukan penanganan segera, baik dari sisi teknologi maupun edukasi kepada masyarakat.

Untuk mengatasi ancaman ini, berbagai metode deteksi telah dikembangkan, termasuk yang berbasis teknologi kecerdasan buatan (*Artificial Intelligence*). Salah satu pendekatan yang kini banyak digunakan adalah penggunaan metode *machine learning* untuk mendeteksi *website phishing*. Metode ini menawarkan solusi yang lebih cepat dan akurat dibandingkan dengan teknik tradisional. Oleh karena itu, pada penelitian ini mengkaji dan mengimplementasikan tiga algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *Gradient Boosting*, dan *Neural*

Networks, guna mendeteksi *website phishing*. SVM dipilih karena kemampuannya dalam menangani data berdimensi tinggi dan klasifikasi biner secara optimal. *Gradient Boosting* digunakan karena kekuatannya dalam menggabungkan beberapa model untuk meningkatkan akurasi prediksi. Sementara itu, *Neural Networks* mampu mengenali pola yang lebih kompleks melalui arsitektur jaringan saraf berlapis. Dengan membandingkan ketiga metode ini, penelitian ini bertujuan untuk menemukan pendekatan terbaik dalam mendeteksi website phishing secara efektif dan efisien, guna meningkatkan keamanan siber dan melindungi pengguna dari ancaman penipuan daring.

2. TINJAUAN PUSTAKA

Phishing merupakan salah satu bentuk kejahatan siber yang bertujuan untuk mencuri informasi sensitif, seperti kredensial *login*, data perbankan, dan informasi pribadi, dengan menggunakan situs *web* palsu yang menyerupai situs resmi. Penelitian sebelumnya telah mengembangkan berbagai metode untuk mendeteksi situs *phishing* menggunakan algoritma *machine learning*. Irawan, dkk (2021) [9] membandingkan performa empat algoritma klasifikasi, yaitu *Support Vector Machine* (SVM), *Decision Tree*, *Random Forest*, dan *Multilayer Perceptron* (*Neural Networks*), untuk mendeteksi *phishing website*. Hasil penelitian menunjukkan bahwa *Multilayer Perceptron* memiliki performa terbaik dengan akurasi 93.15% dan nilai AUC 0.976. Penelitian lainnya oleh Vinod, dkk. (2023) [10] menggunakan algoritma *Random Forest* dan *Decision Tree* dalam mendeteksi situs *phishing* berdasarkan URL. Hasilnya, *Random Forest* memberikan akurasi tertinggi sebesar 97.31% dan telah diimplementasikan dalam bentuk ekstensi browser *Chrome* untuk membantu pengguna mengenali situs *phishing* secara *real-time*. Arifin, dkk. (2024) [11] mengembangkan sistem berbasis *web* dengan metode *Certainty Factor* untuk mendiagnosis serangan *phishing* berdasarkan gejala tertentu. Sistem ini menunjukkan akurasi 85.1% dan berhasil membantu pengguna mengidentifikasi potensi serangan *phishing*. Berdasarkan penelitian – penelitian sebelumnya, pendekatan *machine learning* terbukti efektif untuk mendeteksi *phishing website*. Namun, penelitian lebih lanjut diperlukan untuk meningkatkan akurasi dan mengatasi tantangan, seperti manipulasi URL yang semakin kompleks. Oleh karena itu, penelitian ini mengkaji penerapan algoritma SVM, *Gradient Boosting*, dan *Neural Networks* untuk meningkatkan akurasi dalam mendeteksi *phishing website* secara komprehensif.

2.1. Phising

Phishing adalah tindakan penipuan yang bertujuan mencuri informasi sensitif, seperti *username*, *password*, dan data kartu kredit, dengan menyamar sebagai entitas terpercaya atau organisasi resmi. Serangan ini biasanya dilakukan melalui komunikasi elektronik, seperti email atau *website* palsu [11].

Karakteristik *phishing* dapat diklasifikasikan ke dalam empat kategori utama, yaitu [12] :

a. *Address Bar – based Features* : Menyasar manipulasi elemen pada address bar.
b. *Abnormal – based Features* : Mengidentifikasi elemen yang tidak sesuai dengan standar atau abnormal.
c. *HTML and JavaScript – based Features* : Memanfaatkan kode HTML dan JavaScript untuk menyembunyikan atau memalsukan informasi.
d. *Domain – based Features* : Menggunakan nama domain palsu untuk meniru *website* resmi.

2.2. Keamanan Siber

Keamanan siber mencakup serangkaian kebijakan, prosedur, dan teknologi yang bertujuan melindungi, mendeteksi, memperbaiki, dan mempertahankan sistem informasi dari kerusakan, akses tidak sah, dan eksploitasi [13].

Salah satu aspek penting dalam keamanan siber adalah perlindungan data, yang diatur oleh undang-undang untuk menjaga privasi dan keamanan data individu [14].

2.3. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah algoritma *machine learning* yang sering digunakan untuk tugas klasifikasi dan regresi. SVM bekerja dengan mencari hyperplane yang secara optimal memisahkan dua kelas data sehingga dapat menghasilkan model prediksi yang akurat [15].

Salah satu keunggulan SVM adalah kemampuannya untuk menangani data dengan dimensi tinggi dan hasil klasifikasi yang baik meskipun *dataset* kecil. Proses penentuan support vector dalam SVM dapat dirumuskan sebagai masalah pemrograman kuadratik (*Quadratic Programming*, QP) sehingga implementasinya relatif sederhana [16].

2.4. *Gradient Boosting*

Boosting adalah algoritma *machine learning* yang membangun model prediktif secara bertahap dengan menggabungkan beberapa model keputusan (*Decision Tree*) lemah menjadi model yang lebih kuat. Metode ini menggunakan pendekatan iteratif untuk meminimalkan kesalahan prediksi pada setiap langkah. *Gradient Boosting* sangat efektif untuk menangani *dataset* yang kompleks dan sering digunakan dalam tugas klasifikasi dan regresi [17].

2.5. *Neural Networks*

Neural Networks adalah salah satu pendekatan dalam *deep learning* yang terinspirasi dari cara kerja otak manusia. Algoritma ini menggunakan sejumlah lapisan *neuron* yang saling terhubung untuk mengenali pola dalam data yang tidak linear dan kompleks. *Neural Networks* sangat efektif dalam tugas klasifikasi, seperti mendeteksi *phishing website*,

dengan memanfaatkan fitur mendalam dari URL, struktur, dan elemen *website* untuk memberikan hasil prediksi yang akurat [18].

3. METODE PENELITIAN

3.1. Pengumpulan Data

Penelitian ini menggunakan *dataset* PhiUSIIL *Phishing* URL *Dataset*, yang terdiri atas total 235.795 sampel URL, dengan 134.850 URL *valid* dan 100.945 URL *phishing*. Fitur – fitur yang diekstraksi dari *dataset* ini mencakup karakteristik URL dan kode sumber halaman *web*, seperti *CharContinuationRate* (tingkat kelanjutan karakter dalam URL), *URLTitleMatchScore* (kesamaan antara judul halaman dan URL), *URLCharProb* (probabilitas kemunculan karakter dalam URL), dan *TLDLegitimateProb* (probabilitas dari domain tingkat atas yang sah). Fitur-fitur ini memungkinkan pemisahan URL *phishing* dari URL yang sah berdasarkan pola dan karakteristik spesifik yang terdapat dalam data.

3.2. *Preprocessing* Data

Proses *preprocessing* data dilakukan dalam dua tahap utama :

a. Normalisasi Data
   Data numerik dinormalisasi menggunakan metode *Min – Max Scaling* untuk memastikan semua fitur berada dalam rentang 0 hingga 1, sehingga mengurangi bias akibat skala fitur yang berbeda.
b. Pembagian Data
   Dataset dibagi menjadi data latih (80%) dan data uji (20%) menggunakan metode *train – test split*. *Pembagian* ini memastikan model dilatih pada data yang cukup besar dan diuji pada data yang terpisah untuk mengukur performa generalisasi.

3.3. Implementasi Model

Penelitian ini menggunakan tiga model *machine learning* dengan detail implementasi sebagai berikut :

a. *Support Vector Machine* (SVM)
   Model *ini* bekerja dengan menemukan *hyperplane* optimal yang memisahkan dua kelas data. Proses implementasi SVM melibatkan :
   - Pencarian parameter optimal menggunakan *Grid Search* dengan *cross-validation* untuk menemukan kombinasi parameter terbaik seperti C dan gamma.
   - Model dilatih menggunakan dataset latih yang telah dinormalisasi.
   - Evaluasi model menggunakan dataset uji untuk mengukur performanya berdasarkan metrik evaluasi.
b. *Gradient Boosting*
   Algoritma ini bekerja dengan menggabungkan beberapa *weak learners* (*decision trees*) untuk meningkatkan performa klasifikasi. Implementasi dilakukan melalui :
   - Model dilatih dengan parameter default sebagai baseline.
   - Model diperbaiki melalui iterasi berulang dengan menyesuaikan bobot kesalahan dari model sebelumnya.
   - Evaluasi performa model diukur dengan metrik evaluasi.
c. *Neural Networks*
   Model ini menggunakan jaringan saraf tiruan dengan beberapa lapisan tersembunyi. Prosesnya meliputi :
   - Desain arsitektur model menggunakan dua lapisan tersembunyi dengan aktivasi ReLU.
   - Regularisasi menggunakan *dropout layer* untuk mencegah *overfitting*.
   - Pelatihan model dengan *Adam optimizer* dan fungsi *loss binary crossentropy*.
   - Model diuji menggunakan dataset uji.

3.4. Evaluasi Model

Evaluasi model dilakukan menggunakan data uji dengan metrik berikut :

a. Akurasi (Accuracy) : Mengukur persentase prediksi yang benar terhadap total data. Semakin tinggi nilai akurasi, semakin baik model dalam mengklasifikasikan data dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + \ FN} \quad (1)$$

b. *F1-Score* : Metrik yang mempertimbangkan keseimbangan antara presisi dan recall, terutama berguna jika data tidak seimbang. Jika *F1-Score* mendekati 1, berarti model memiliki keseimbangan baik antara precision dan recall.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Reacall} \quad (4)$$

   Keterangan :
   TP = Jumlah sampel positif yang diprediksi benar.
   TN = Jumlah sampel negatif yang diprediksi benar.
   FP = Jumlah sampel negatif yang diprediksi salah sebagai positif.
   FN = Jumlah sampel positif yang diprediksi salah sebagai negatif
c. *Confusion Matrix* : Menyediakan rincian prediksi benar dan salah untuk tiap kelas, sehingga memberikan gambaran lebih rinci tentang performa model.

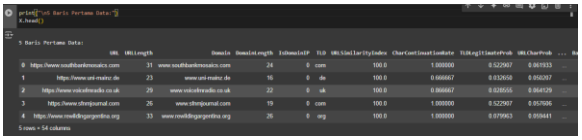
3.5. Analisis Hasil

Hasil evaluasi dari ketiga model dibandingkan untuk menentukan metode terbaik berdasarkan *F1 – Score*, *Confusion Matrix*, dan waktu komputasi.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan Data

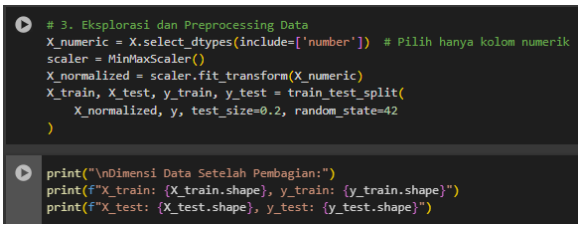
Dengan melakukan pencarian pada situs UCI *Repository*, diperoleh *dataset* PhiUSIIL *Phishing* URL. *Dataset* ini terdiri atas 235.795 sampel data, yang mencakup 134.850 URL legitim dan 100.945 URL *phishing*. Data *phishing* URL diperoleh dari arsip terbaru situs *phishing*, sedangkan data URL legitim dikumpulkan dari berbagai sumber tepercaya. *Dataset* ini memiliki 54 fitur yang mencakup tipe *real*, kategorikal, dan *integer*. Format *dataset* dapat dilihat pada gambar 1 di bawah ini.



Gambar 1. Datase PhiUSILL

4.2. Pelatihan dan Pengujian Data

Tahapan pelatihan dan pengujian dimulai dengan eksplorasi dan *preprocessing* data. Pada tahap ini, hanya kolom numerik yang dipilih menggunakan metode seperti pada gambar 2.



```
# 3. Eksplorasi dan Preprocessing Data
X_numeric = X.select_dtypes(include=['number'])  # Pilih hanya kolom numerik
scaler = MinMaxScaler()
X_normalized = scaler.fit_transform(X_numeric)
X_train, X_test, y_train, y_test = train_test_split(
    X_normalized, y, test_size=0.2, random_state=42
)

print("\nDimensi Data Setelah Pembagian:")
print(f"X_train: {X_train.shape}, y_train: {y_train.shape}")
print(f"X_test: {X_test.shape}, y_test: {y_test.shape}")
```

Gambar 2. *Preprocessing* Data

Berikut hasil dari *preprocessing* dan pembagian data adalah sebagai berikut:

a. X_*train* : Berisi fitur untuk data latih dengan dimensi (188636, 50)
b. y_*train* : Berisi target untuk data latih dengan dimensi (188636, 1)
c. X_*test* : Berisi fitur untuk data uji dengan dimensi (47159, 50)
d. y_*test* : Berisi target untuk data uji dengan dimensi (47159, 1)

Proses pembagian data ini dilakukan untuk memastikan bahwa model dilatih pada data yang cukup besar, sehingga mampu mempelajari pola secara optimal, dan menguji pada data yang terpisah untuk mengukur performa generalisasi.

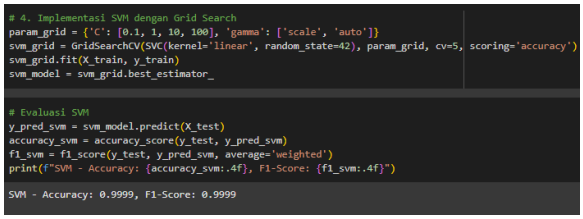
Selanjutnya, data fitur (X) yang digunakan dalam pelatihan dinormalisasi menggunakan *MinMaxScaler* agar berada dalam rentang nilai [0, 1]. Langkah ini dilakukan untuk mengurangi bias akibat skala fitur yang berbeda dan mempercepat proses pelatihan model.

4.3. Implementasi Model

Pada tahap implementasi model, dilakukan eksperimen menggunakan beberapa algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *Gradient Boosting*, dan *Neural Networks*.

4.4. *Support Vector Machine* (SVM)

Model SVM diimplementasikan dengan pencarian hyperparameter menggunakan *GridSearchCV* untuk menemukan parameter optimal yang meningkatkan performa klasifikasi. Setelah diperoleh parameter terbaik, model dilatih menggunakan dataset latih dan diuji menggunakan dataset uji untuk mengukur akurasi dan efektivitas prediksi. Pengimplementasiannya dapat dilihat pada gambar 3.

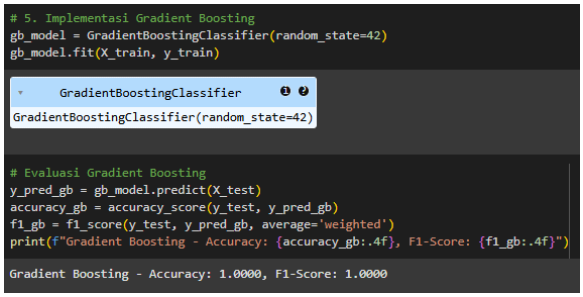


```
# 4. Implementasi SVM dengan Grid Search
param_grid = {'C': [0.1, 1, 10, 100], 'gamma': ['scale', 'auto']}
svm_grid = GridSearchCV(SVC(kernel='linear', random_state=42), param_grid, cv=5, scoring='accuracy')
svm_grid.fit(X_train, y_train)
svm_model = svm_grid.best_estimator_

# Evaluasi SVM
y_pred_svm = svm_model.predict(X_test)
accuracy_svm = accuracy_score(y_test, y_pred_svm)
f1_svm = f1_score(y_test, y_pred_svm, average='weighted')
print(f"SVM - Accuracy: {accuracy_svm:.4f}, F1-Score: {f1_svm:.4f}")

SVM - Accuracy: 0.9999, F1-Score: 0.9999
```

Gambar 3. Implementasi SVM

4.5. *Gradient Boosting*

Model *Gradient Boosting* diterapkan tanpa *hyperparameter tuning* untuk mendapatkan *baseline* performa. Model ini dilatih menggunakan dataset latih, kemudian diuji dengan dataset uji untuk menilai distribusi prediksi yang dihasilkan. Proses ini membantu mengidentifikasi kemungkinan kesalahan prediksi serta peluang peningkatan model di tahap selanjutnya. Pengkodeannya terlihat pada gambar 4.



```
# 5. Implementasi Gradient Boosting
gb_model = GradientBoostingClassifier(random_state=42)
gb_model.fit(X_train, y_train)

GradientBoostingClassifier
GradientBoostingClassifier(random_state=42)

# Evaluasi Gradient Boosting
y_pred_gb = gb_model.predict(X_test)
accuracy_gb = accuracy_score(y_test, y_pred_gb)
f1_gb = f1_score(y_test, y_pred_gb, average='weighted')
print(f"Gradient Boosting - Accuracy: {accuracy_gb:.4f}, F1-Score: {f1_gb:.4f}")

Gradient Boosting - Accuracy: 1.0000, F1-Score: 1.0000
```

Gambar 4. Implementasi *Gradient Boosting*

4.6. *Neural Networks*

Model *Neural Networks* dibangun menggunakan arsitektur sederhana dengan Keras *Sequential* API. Arsitektur jaringan terdiri dari dua lapisan tersembunyi dengan aktivasi ReLU, serta penggunaan *dropout layer* untuk mengurangi risiko *overfitting*. Model dilatih menggunakan *optimizer Adam* dengan fungsi *loss binary crossentropy*. Setelah pelatihan selesai, model diuji pada dataset uji untuk mengevaluasi akurasi Pengempementasian dapat dilihat pada gambar 5.

```
# 6. Implementasi Neural Networks
nn_model = Sequential([
    Dense(64, activation='relu', input_dim=X_train.shape[1]),
    Dropout(0.2),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
nn_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
nn_model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=0)

<keras.src.callbacks.history.History at 0x7da72801fb80>

# Evaluasi Neural Networks
y_pred_nn = (nn_model.predict(X_test) > 0.5).astype(int)
accuracy_nn = accuracy_score(y_test, y_pred_nn)
f1_nn = f1_score(y_test, y_pred_nn, average='weighted')
print(f"Neural Networks - Accuracy: {accuracy_nn:.4f}, F1-Score: {f1_nn:.4f}")

1474/1474 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step
Neural Networks - Accuracy: 0.9999, F1-Score: 0.9999
```

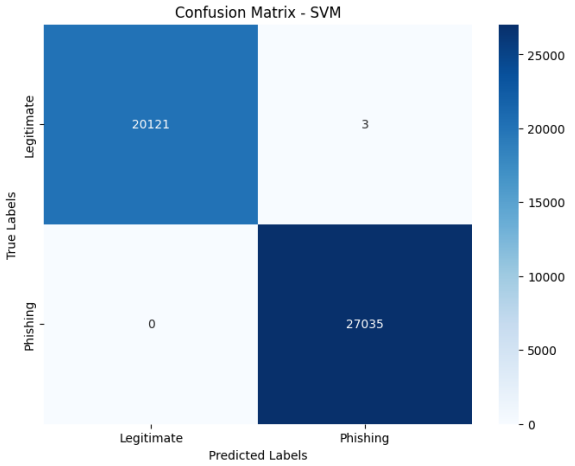
Gambar 5. Implementasi *Neural Networks*

4.7. Analisis Hasil Model

Pada penelitian ini, tiga algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *Gradient Boosting*, dan *Neural Networks*, digunakan untuk mendeteksi URL *phishing*. Evaluasi dilakukan menggunakan data uji dengan metrik akurasi, *F1 – Score*, *Confusion Matrix*, dan waktu komputasi. Berikut adalah hasil analis dari proses yang telah dilakukan

4.8. *Support Vector Machine* (SVM)

Support Vector Machine (SVM) menunjukkan performa yang sangat baik dalam mendeteksi URL *phishing* dengan akurasi sebesar 99,99% dan *F1 – Score* sebesar 99,99%. Berdasarkan *Confusion Matrix* pada gambar 6, model ini mampu mengklasifikasikan sebagian besar data dengan benar, dengan jumlah *True Positive* sebesar 27.035 dan *True Negative* sebesar 20.121, serta hanya terdapat 3 *False Positive* dan 0 *False Negative*.

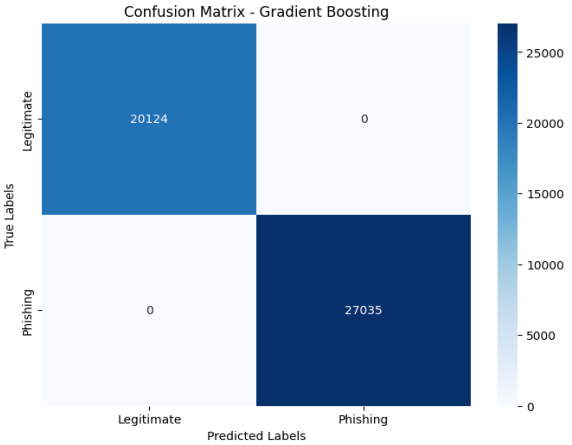


Gambar 6. *Confusion Matrix* SVM

Namun, waktu komputasi yang dibutuhkan untuk melatih dan mengevaluasi model ini adalah 5 menit, yang merupakan waktu terlama dibandingkan model lainnya. Meskipun memiliki akurasi yang tinggi, waktu komputasi yang relatif panjang membuat SVM kurang efisien untuk digunakan pada *dataset* yang sangat besar atau dalam skenario dengan kebutuhan *real-time*.

4.9. *Gradient Boosting*

Gradient Boosting memberikan hasil terbaik di antara ketiga model yang diuji, dengan akurasi sempurna sebesar 100% dan *F1-Score* sebesar 100%. Selain itu, waktu komputasi untuk model ini sangat efisien, hanya membutuhkan 1 menit. *Confusion Matrix* pada gambar 7 menunjukkan bahwa model ini berhasil mengklasifikasikan seluruh data dengan benar tanpa adanya kesalahan, dengan *True Positive* sebesar 27.035 dan *True Negative* sebesar 20.124, serta tidak ada *False Positive* maupun *False Negative*.

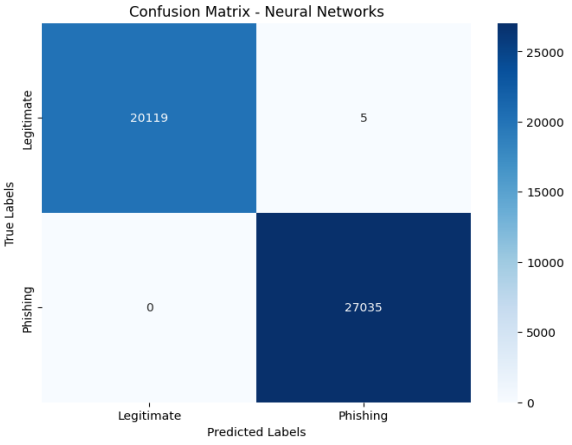


Gambar 7. *Confusion Matrix Gradient Boosting*

Dengan hasil ini, *Gradient Boosting* menjadi solusi yang sangat andal dan efisien untuk mendeteksi URL *phishing* pada *dataset* yang digunakan.

4.10. *Neural Networks*

Neural Networks juga menunjukkan performa yang sangat tinggi, dengan akurasi sebesar 99,99% dan *F1-Score* sebesar 99,99%. Waktu komputasi model ini hanya 1 menit, sama seperti *Gradient Boosting*, yang menunjukkan efisiensi dalam proses pelatihan dan evaluasi. Berdasarkan *Confusion Matrix* pada gambar 8 , model ini mencatat *True Positive* sebesar 27.035 dan *True Negative* sebesar 20.119, dengan hanya 5 *False Positive* dan 0 *False Negative*.



Gambar 8. *Confusion Matrix Neural Networks*

Meskipun terdapat beberapa kesalahan klasifikasi pada data sah yang dianggap sebagai *phishing*, jumlahnya sangat kecil dan tidak signifikan. *Neural Networks* menjadi pilihan yang kompetitif, terutama untuk *dataset* dengan pola yang kompleks.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, implementasi algoritma *Support Vector Machine* (SVM), *Gradient Boosting*, dan *Neural Networks* pada *dataset PhiUSIIL* menunjukkan bahwa *Gradient Boosting* memberikan performa terbaik dengan akurasi sempurna 100% dan waktu komputasi tercepat 1 menit, menjadikannya solusi optimal untuk deteksi phishing. Model SVM mencapai akurasi 99,99%, namun memiliki waktu komputasi lebih lama yaitu 5 menit, sehingga kurang efisien untuk skenario *real – time*. Sementara itu, *Neural Networks* juga menunjukkan performa tinggi dengan akurasi 99,99%, tetapi masih memiliki beberapa kesalahan klasifikasi minor.

Secara keseluruhan, penelitian ini membuktikan bahwa *Gradient Boosting* merupakan model terbaik untuk mendeteksi phishing pada *dataset* yang digunakan. Evaluasi lebih lanjut dengan *dataset* yang lebih beragam dan teknik optimasi *hyperparameter* diharapkan dapat meningkatkan generalisasi model untuk aplikasi nyata.

DAFTAR PUSTAKA

[1] M. A. G. Al Ghifani, B. Hananto, and B. T. Wahyono, "Implementasi Ekstensi Google Chrome Dalam Mendeteksi Situs Web Phishing Menggunakan Algoritma Random Forest," *Semin. Nas. Mhs. Ilmu Komput. dan Apl.*, vol. 3, pp. 179–189, 2022, [Online]. Available: https://conference.upnvj.ac.id/index.php/senamika/article/view/2152%0Ahttps://conference.upnvj.ac.id/index.php/senamika/article/download/2152/1744

[2] S. Diantika, "Penerapan Teknik Random Oversampling Untuk Mengatasi Imbalance Class Dalam Klasifikasi Website Phishing Menggunakan Algoritma Lightgbm," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 19–25, 2023, doi: 10.36040/jati.v7i1.6006.

[3] A. Ferdita Nugraha, R. F. A. Aziza, and Y. Pristyanto, "Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing," *J. Infomedia*, vol. 7, no. 1, p. 39, 2022, doi: 10.30811/jim.v7i1.2959.

[4] N. B. Putri and A. W. Wijayanto, "Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing," *Komputika J. Sist. Komput.*, vol. 11, no. 1, pp. 59–66, 2022, doi: 10.34010/komputika.v11i1.4350.

[5] D. B. Suwarno, M. Hardjianto, F. T. Informasi, U. B. Luhur, and K. J. Selatan, "DETEKSI WEBSITE PHISHING DARI ANALISIS URL PHISHING WEBSITE DETECTION FROM URL ANALYSIS USING RANDOM FOREST ALGORITHM," vol. 21, no. 2, pp. 145–152, 2024.

[6] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, and A. Setiawan, "Deteksi Website Phishing Menggunakan Teknik Filter Pada Model Machine Learning," *Inf. Syst. J.*, vol. 6, no. 01, pp. 39–43, 2023, doi: 10.24076/infosjournal.2023v6i01.1268.

[7] A. D. Sulistyo, B. D. Wicaksono, R. N. Saputra, and R. Ramadhani, "Strategi Penanggulangan Serangan Phishing di Media Sosial," pp. 385–396, 2024.

[8] IDADX, "LAPORAN AKTIVITAS ABUSE DOMAIN .ID," 2024. [Online]. Available: https://api.idadx.id/documents/uploads/1724725529_Laporan Q2 2024.pdf.pdf

[9] 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.

[10] D. Vinod, M., Vaishnavi, M., Tejasree, "Phishing Website Detection," *Indian J. Data Min.*, vol. 4, no. 1, pp. 38–41, 2023, doi: 10.54105/ijdm.a1642.04010524.

[11] A. Arifin and H. Rahmawati, "Sistem Pakar Diagnosa Phising Dengan Metode Certainty Factor Berbasis Web: Sistem Pakar," *J. SIGN J. Ilm. Sist. Inf. dan Inform.*, vol. 3, no. 1, pp. 11–21, 2024.

[12] T. F. Handoyo, M. Pajar, and K. Putra, "Optimasi Bobot Kelas LSTM untuk Deteksi URL Phishing pada Dataset Tidak Berimbang," vol. 10, no. 1, pp. 20–36, 2025, doi: 10.30591/jpit.v10i1.8128.

[13] R. Diseria and S. Fadhlain, "Strategi Komunikasi DISKOMINSA Kabupaten Simeulue dalam Melaksanakan Sosialisasi Persandian dan Keamanan Siber," *COMSERVA J. Penelit. dan Pengabdi. Masy.*, vol. 3, no. 09, pp. 3596–3605, 2024, doi: 10.59141/comserva.v3i09.1161.

[14] G. Wibisono, R. A. G. Gultom, and T. Mantoro, "Strategi Peningkatan Kapabilitas Satuan Siber Dispamsanau Melalui Pemanfaatan Artificial Intelligence Pada Keamanan Siber Berdasarkan National Institute of Standards and Technology Cybersecurity Framework Version 1.1," *J. Rev. Pendidik. dan Pengajaran*, vol. 7, no. 1, pp. 968–975, 2024.

[15] M. Vebriani and W. Yustanti, "Klasifikasi Deteksi Link Phising DANA Kaget Menggunakan Metode Support Vector Machine Berbasis Website," *J. Informatics Comput. Sci.*, vol. 06, pp. 411–412, 2024, [Online]. Available: https://danakagetvezridd.

[16] R. W. Dwinanto, A. S. S. A, and R. Ardianto, "KLASIFIKASI BERISIKO STUNTING PADA BALITA : PERBANDINGAN K- NEAREST

NEIGHBOR , NAÏVE BAYES , SUPPORT VECTOR MACHINE," vol. 8, no. 2, pp. 264–273, 2024.

[17] W. A. Naseer, S. Sarwido, and B. B. Wahono, "Gradient Boosting Optimization with Pruning Technique for Prediction of Bmt Al-hikmah Permata Customer Data," *Jinteks*, vol. 6, no. 3, pp. 719–727, 2024.

[18] C. M. Bachri and W. Gunawan, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Deteksi Email Spam menggunakan Algoritma Convolutional Neural Network (CNN)," *Edukasi dan Penelit. Inform.*, vol. 10, no. 1, pp. 88–94, 2024.