BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Table 2. 1 Penelitian Terdahulu

Penulis	Judul Jurnal	Nama Jurnal	Metode	Hasil
Ahmad et al. [6]	Phishing Website Detection Based on URL Features Using SVM	Journal of Cybersecurity	SVM, URL Features	Akurasi 95.6%, fitur panjang URL, simbol, SVM unggul dibanding Naïve Bayes dan Decision Tree.
Lertwatechakul et al. [9]	Hybrid SVM Model for Phishing Detection with Whitelist Features	International Journal of Information Technology	SVM, Hybrid Whitelist	Akurasi 92.4%, false positive rate rendah; Decision Tree lebih rendah akurasinya.

UNIVERSITAS MULTIMEDIA NUSANTARA

Penulis	Judul Jurnal	Nama Jurnal	Metode	Hasil
Kumar et al. [10]	Lightweight Phishing Detection via SVM with IP Presence & URL Structure	Neural Computing and Applications	SVM, URL IP Presence, Feature Analysis	Akurasi 93.5%, waktu deteksi lebih cepat, dibandingkan dengan Naïve Bayes yang 89%.
Chen et al. [7]	Application of Whitelist and Feature-Based Detection Methods in SVM	Journal of Security and Privacy	SVM, Whitelist Filtering	Akurasi 90.2%, fitur whitelist efektif; performa dibandingkan dengan Decision Tree yang 85%.
Anand & Patil [11]	Highly Accurate Phishing URL Detection Using Machine Learning	Cybersecurity	SVM, Random Forest, Feature Extraction	SVM mencapai akurasi 97.3%, Random Forest 95.8%; Naïve Bayes 92%.

Penulis	Judul Jurnal	Nama Jurnal	Metode	Hasil
Guo et al. [12]	URL Phishing Detection Using Deep Learning and SVM	Future Internet	SVM, Deep Learning	Akurasi 95.2%, SVM lebih cepat dalam waktu komputasi dibandingkan Decision Tree yang 91.5%.
Alkawaz & Steven [13]	Feature Extraction-Based Phishing URL Detection	Advances in Intelligent Systems	SVM, Whitelist, Feature Extraction	Akurasi 94.7%, fitur URL berperan besar; Decision Tree mencapai 90.3%, Naïve Bayes 89%.
Wella [14]	Personalized Learning Models Using Decision Tree Algorithm	International Journal on Informatics Visualization	Decision Tree	Studi tentang model pembelajaran personalisasi berbasis algoritma Decision Tree; akurasi 90%.
Erick Fernando [15]	Comparative Analysis of KNN and Decision Tree for Financial Data Prediction	Journal of Information Systems and Informatics	Decision Tree, KNN	Decision Tree akurasi 87%, dibandingkan KNN yang lebih rendah pada dataset keuangan.

MULTIMEDIA NUSANTARA

Penelitian yang tercantum dalam tabel 2.1 secara konsisten menunjukkan keunggulan Support Vector Machine (SVM) dalam mendeteksi situs phishing,

terutama dalam hal akurasi yang lebih tinggi dibandingkan dengan algoritma lain seperti Naïve Bayes dan Decision Tree. Ahmad et al. melaporkan bahwa SVM yang menggunakan fitur berbasis URL, seperti panjang URL dan simbol tertentu, mencapai akurasi sebesar 95.6%, yang secara signifikan lebih unggul dibandingkan Naïve Bayes dan Decision Tree [6]. Temuan ini menegaskan kemampuan SVM untuk memanfaatkan pola spesifik dalam data URL guna menghasilkan klasifikasi yang sangat akurat.

Penelitian yang dilakukan oleh Chen et al. juga memperkuat dominasi SVM, di mana penerapan *whitelist filtering* menghasilkan akurasi sebesar 90.2%, lebih tinggi daripada Decision Tree yang hanya mencapai akurasi 85% [7]. Hal ini menunjukkan fleksibilitas SVM dalam mengintegrasikan teknik pendukung untuk meningkatkan akurasi deteksi.

Lebih lanjut, studi Kumar et al. menambahkan bukti empiris tentang efektivitas SVM dengan memanfaatkan fitur tambahan, seperti kehadiran IP dan struktur URL. Dengan pendekatan ini, SVM mencapai akurasi sebesar 93.5%, mengungguli Naïve Bayes yang hanya mencatat akurasi 89%. Selain itu, SVM juga menunjukkan efisiensi waktu yang lebih baik dalam proses deteksi [16].

Guo et al. mengembangkan SVM dengan mengintegrasikan teknologi deep learning dan berhasil mencapai akurasi sebesar 95.2%, lebih tinggi daripada Decision Tree yang hanya mencapai akurasi 91.5%. Selain akurasi, SVM juga unggul dalam waktu komputasi, menjadikannya pilihan yang lebih efisien untuk aplikasi deteksi phishing berbasis machine learning [12].

Anand & Patil semakin memperkokoh keunggulan SVM melalui eksplorasi model ensembel, yang menggabungkan SVM dengan metode lain untuk meningkatkan performa. Dengan metode ekstraksi fitur tingkat lanjut, SVM mencapai akurasi tertinggi di antara semua model yang diuji, yaitu 97.3%, mengungguli Random Forest (95.8%) dan Naïve Bayes (92%) [11]. Temuan ini

mengukuhkan SVM sebagai algoritma terbaik untuk deteksi phishing, baik dalam hal akurasi maupun fleksibilitas penerapannya.

Secara keseluruhan, berbagai hasil penelitian ini menunjukkan secara kuat bahwa SVM merupakan algoritma yang unggul untuk mendeteksi situs phishing. Algoritma ini menonjol berkat akurasi tinggi yang konsisten, kemampuan adaptasi yang baik dalam mengintegrasikan fitur tambahan seperti *whitelist filtering*, serta efisiensi dalam waktu komputasi, sehingga menjadi pilihan yang sangat tepat untuk sistem deteksi phishing berbasis machine learning.

2.2 Tinjauan Teori

2.2.1 URL Phishing

URL phishing adalah teknik manipulatif yang digunakan oleh penjahat siber untuk mencuri informasi pribadi pengguna dengan mengarahkan mereka ke situs web palsu yang tampak seperti situs asli. Penyerang biasanya mengirimkan email atau pesan yang mengandung link URL yang tampak sah, yang ketika diklik, mengarahkan korban ke situs yang meniru situs resmi [17]. Situs ini dirancang untuk mengelabui korban agar memasukkan informasi sensitif seperti nama pengguna, kata sandi, atau detail kartu kredit. Serangan ini memanfaatkan kepercayaan pengguna terhadap entitas yang dikenal, seperti bank atau penyedia layanan, sehingga meningkatkan kemungkinan keberhasilan serangan.

Metode URL phishing sangat efektif karena situs web palsu biasanya memiliki tampilan dan nuansa yang sangat mirip dengan situs resmi. Penyerang menggunakan berbagai teknik untuk membuat URL terlihat sah, termasuk menggunakan subdomain yang menyerupai domain asli atau menambahkan karakter khusus yang sulit dikenali oleh mata pengguna [18]. Selain itu, pesan phishing sering kali mengandung elemen yang mendesak, seperti ancaman penangguhan akun atau permintaan pembaruan informasi yang cepat, untuk

memanipulasi korban agar bertindak tanpa berpikir panjang. Penyerang juga dapat menggunakan metode seperti pharming, di mana mereka mengarahkan pengguna ke situs palsu bahkan ketika korban mengetik URL asli di peramban mereka [19].

URL phishing memiliki dampak yang signifikan jika berhasil. Informasi pribadi yang dicuri dapat digunakan untuk berbagai bentuk penipuan, termasuk pencurian identitas, transaksi keuangan yang tidak sah, dan pengambilalihan akun. Di samping kerugian finansial langsung, serangan phishing juga dapat mengakibatkan kerusakan reputasi bagi individu dan organisasi yang terpengaruh. Perusahaan yang menjadi target atau korban serangan phishing dapat kehilangan kepercayaan pelanggan, yang pada gilirannya dapat berdampak pada keuntungan bisnis mereka. Oleh karena itu, penting untuk selalu waspada terhadap tanda-tanda phishing dan mengadopsi praktik keamanan yang baik, seperti memverifikasi URL sebelum mengklik link dan menggunakan alat keamanan yang dapat mendeteksi dan memblokir situs phishing [20].

2.2.2 Mekanisme Serangan URL Phishing [21]

1. Pengiriman Email atau Pesan Phishing

   Penyerang mengirimkan email atau pesan yang tampak seperti berasal dari sumber tepercaya, seperti bank, penyedia layanan online, atau rekan kerja. Pesan ini sering kali mencakup permintaan mendesak untuk mengambil tindakan, seperti memperbarui informasi akun atau mengonfirmasi transaksi.

2. Pengarahan ke Situs Palsu

Link yang disertakan dalam pesan tersebut mengarahkan korban ke situs web yang telah disiapkan oleh penyerang. Situs ini meniru desain dan branding situs asli untuk menipu korban agar merasa aman.

3. Pengumpulan Informasi Sensitif

Ketika korban memasukkan informasi mereka di situs palsu tersebut, data tersebut dikirimkan langsung ke penyerang. Informasi ini kemudian dapat digunakan untuk berbagai tujuan jahat, seperti pencurian identitas, akses tidak sah ke akun, atau penipuan finansial.

2.2.3 Jenis-jenis URL Phishing

1. Spear Phishing [21]

Menargetkan individu atau organisasi tertentu dengan pesan yang dipersonalisasi. Penyerang menggunakan informasi yang dikumpulkan tentang target untuk membuat email atau pesan yang lebih meyakinkan dan relevan dengan target.

2. Whaling

Menargetkan individu dengan otoritas tinggi dalam organisasi, seperti eksekutif atau CEO. Serangan ini biasanya lebih canggih dan sering kali melibatkan peniruan sebagai rekan bisnis atau pihak berwenang.

3. Clone Phishing

Penyerang menduplikasi email sah yang sebelumnya diterima oleh korban, lalu mengganti link atau lampiran dengan yang berbahaya. Email ini kemudian dikirim ulang seolah-olah berasal dari pengirim asli.

2.2.4 Ciri-ciri URL Phishing

1. URL yang Tidak Biasa[21]

   URL mengandung ejaan yang salah, tambahan angka atau karakter yang tidak lazim, atau domain yang berbeda dari situs web asli. Misalnya, menggunakan "www.bcaa,co,id" alih-alih "www.bca.co.id".

2. Desain Situs yang Kurang Profesional

   Situs phishing sering kali memiliki tampilan yang mirip dengan situs asli tetapi dengan kualitas desain yang lebih rendah, kesalahan tata bahasa, atau gambar yang buram.

3. Ketidakcocokan Domain

   Nama domain atau subdomain mungkin tidak sesuai dengan nama perusahaan atau layanan yang sah. Misalnya, "support-login.banksecure.com" alih-alih "login.bank.com".

4. Permintaan Informasi Pribadi

   Situs atau pesan phishing biasanya meminta informasi sensitif dengan cara yang tidak biasa, seperti meminta password atau informasi keuangan secara langsung.

5. Tidak Ada HTTPS

   Situs web phishing mungkin tidak menggunakan protokol HTTPS yang aman, yang ditandai dengan ikon gembok di sebelah URL di browser.

2.2.5 Bahaya Jika Terkena URL Phishing

1. Pencurian Identitas[21]

   Informasi pribadi yang dicuri dapat digunakan untuk membuka rekening bank, mengajukan kredit, atau melakukan penipuan lainnya atas nama korban.

2. Kehilangan Finansial

   Informasi keuangan seperti nomor kartu kredit dapat digunakan untuk melakukan transaksi tanpa sepengetahuan korban, menyebabkan kerugian finansial yang signifikan.

3. Akses Tidak Sah ke Akun

   Akun email, media sosial, atau layanan lainnya dapat diambil alih oleh penyerang, memungkinkan mereka untuk mengakses data sensitif atau menyebarkan lebih banyak serangan phishing.

4. Kerugian Bisnis

   Jika akun bisnis yang penting terkena phishing, ini dapat menyebabkan kebocoran data penting, kerugian finansial, dan kerusakan reputasi perusahaan.

5. Infeksi Malware

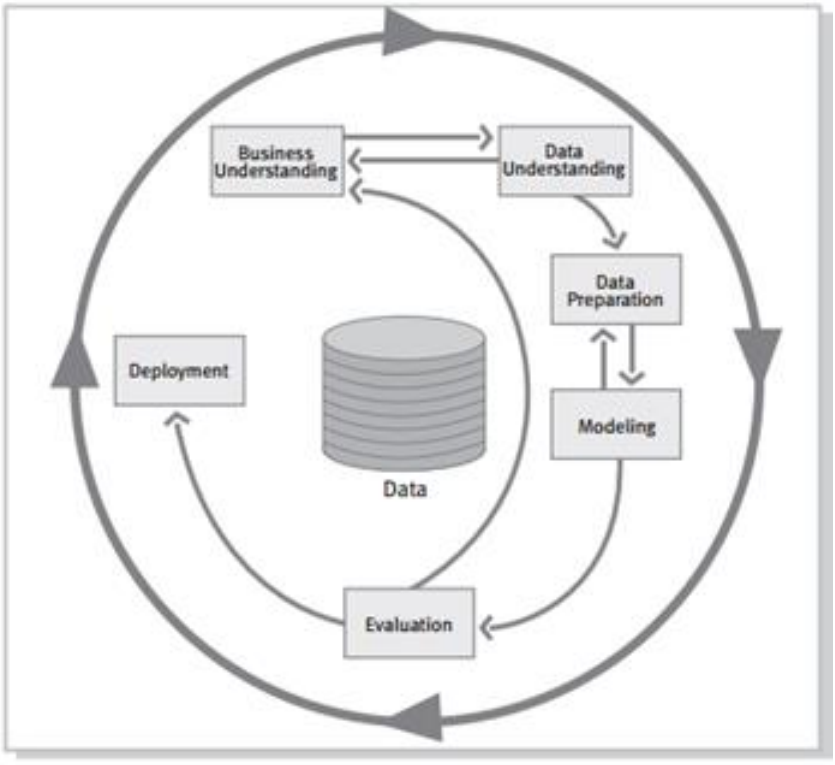
   URL phishing sering kali digunakan untuk menyebarkan malware yang dapat menginfeksi perangkat korban, memungkinkan penyerang untuk mencuri data lebih lanjut atau mengendalikan perangkat dari jarak jauh.

2.3 Framework dan Algoritma

2.3.1 Framework

2.3.1.1 CRISP DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) adalah kerangka kerja yang dibentuk pada tahun 1990-an oleh lima perusahaan: SPSS, Daimler AG, NCR, OHRA, dan TeraData [22]. CRISP-DM terdiri dari enam tahapan: pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, dan penerapan. Gambar 2.1 menunjukkan alur kerja CRISP-DM.



Gambar 2. 1Crisp DM [23]

1) Pemahaman Bisnis (Business Understanding)

Tahap pertama dalam CRISP-DM adalah pemahaman bisnis, di mana tujuan proyek didefinisikan dari perspektif bisnis [24]. Ini sangat penting untuk menetapkan langkah-langkah berikutnya dalam proses.

Memahami kebutuhan dan tujuan bisnis memastikan bahwa analisis data yang dilakukan relevan dan memberikan nilai tambah.

2) Pemahaman Data (Data Understanding)

Tahap ini melibatkan pengumpulan data dan identifikasi masalah dalam data tersebut. Kualitas data dievaluasi, dan insight tersembunyi diidentifikasi. Tahap ini penting untuk memastikan bahwa data yang digunakan akurat dan memadai untuk analisis lebih lanjut .

3) Persiapan Data (Data Preparation)

Pada tahap persiapan data, data dibersihkan dan ditransformasikan untuk digunakan dalam analisis dan pengujian selanjutnya. Ini mencakup penanganan data yang hilang, pengubahan format data, dan pembuatan fitur baru jika diperlukan. Tujuan utama tahap ini adalah untuk menyiapkan data yang optimal untuk pemodelan.

4) Pemodelan (Modeling)

Tahap pemodelan melibatkan pembuatan model dan pengembangan teknik yang akan digunakan. Algoritma yang dipilih diterapkan pada data yang telah dipersiapkan. Data biasanya dibagi menjadi data pelatihan dan pengujian untuk memvalidasi kualitas model. Model ini diuji untuk memastikan kinerja yang baik sebelum melanjutkan ke tahap berikutnya .

5) Evaluasi (Evaluation)

Pada tahap evaluasi, model yang dibuat ditinjau dan hasil analisis diinterpretasikan dalam konteks tujuan bisnis. Berbagai metrik digunakan untuk menilai kualitas model. Jika model tidak memenuhi harapan, peninjauan lebih lanjut dan penyesuaian mungkin diperlukan. Evaluasi ini memastikan bahwa model siap untuk diterapkan.

6) Penerapan (Deployment)

Tahap penerapan adalah tahap akhir dari CRISP-DM, di mana model diterapkan pada proyek nyata. Ini bisa berupa implementasi sistem atau

pembuatan dashboard. Model yang dipilih pada tahap evaluasi digunakan untuk menghasilkan nilai bisnis yang nyata [24].

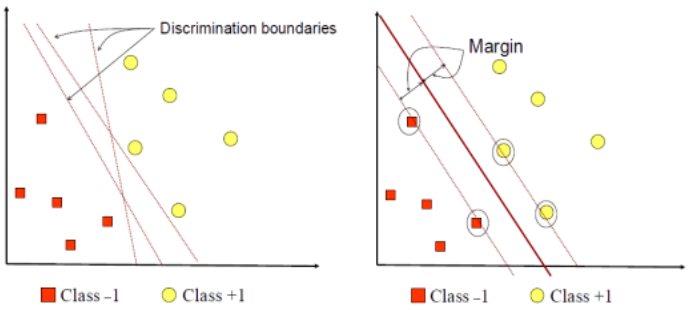
2.3.2 Algoritma

2.3.2.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) adalah algoritma pembelajaran mesin berbasis supervised learning yang digunakan terutama untuk tugas klasifikasi dan regresi. Algoritma ini pertama kali diperkenalkan pada tahun 1992 dan sejak itu menjadi salah satu metode yang populer dalam bidang klasifikasi karena performanya yang kuat dalam memisahkan kelas data yang berbeda [25]. SVM bekerja dengan menentukan garis pemisah atau hyperplane yang optimal, yang mampu memisahkan data ke dalam dua kelas berbeda dengan margin yang maksimum. Dalam penerapannya, SVM sering digunakan pada berbagai aplikasi, seperti pengenalan wajah, klasifikasi teks, dan bioinformatika [26].

Konsep utama dalam SVM adalah mencari hyperplane yang optimal, yaitu garis atau bidang yang mampu memisahkan kelas positif (+1) dan kelas negatif (-1) dengan margin maksimum. Hyperplane tersebut tidak hanya memisahkan kelas, tetapi juga memastikan bahwa data dari kedua kelas berada pada jarak terjauh dari hyperplane. Garis pemisah yang optimal ini akan memaksimalkan margin antara titik data terdekat dari setiap kelas, yang dikenal sebagai support vectors. Dengan memilih hyperplane yang memiliki margin maksimum, SVM bertujuan untuk meningkatkan generalisasi model pada data baru [27].

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 2. 2 Proses SVM dalam Menemukan Hyperline [27]

Berbagai studi menunjukkan bahwa SVM seringkali memberikan hasil yang unggul dalam kasus klasifikasi biner, terutama pada data dengan margin pemisahan yang jelas. SVM juga banyak dibandingkan dengan algoritma lain seperti Neural Networks dan Decision Trees, dan pada beberapa kasus menunjukkan hasil yang lebih akurat dan stabil [28]. Gambar 2.2 menunjukkan ilustrasi dari hyperplane optimal dengan margin maksimum, di mana SVM berupaya untuk mengoptimalkan batas pemisah untuk menghasilkan klasifikasi yang optimal.

Dalam SVM, hyperplane optimal yang memisahkan dua kelas didefinisikan secara matematis sebagai:

$$wTx + b = 0$$

Rumus 2. 1 Rumus hyperplane

Di mana:

- w adalah vektor bobot yang tegak lurus terhadap hyperplane.
- x adalah vektor fitur (data input).
- b adalah bias (intersep), yang menentukan posisi hyperplane relatif terhadap asal koordinat.

Tujuan utama SVM adalah memaksimalkan margin, yaitu jarak antara hyperplane dengan titik-titik data terdekat dari kedua kelas (yang disebut support vectors). Margin dihitung sebagai:

$$Margin = \|w\|2$$

Analisis Perbandingan Algoritma Machine Learning Untuk Deteksi Url Phishing Dengan Pengembangan Aplikasi Berbasis Web Pada PT Bank Central Asia, Tbk ,Michael Owen Kohar , Universitas Multimedia Nusantara

Penggunaan hyperplane dengan margin terbaik ini sangat penting untuk mendapatkan hasil prediksi yang akurat pada data yang belum pernah dilihat oleh model [29].

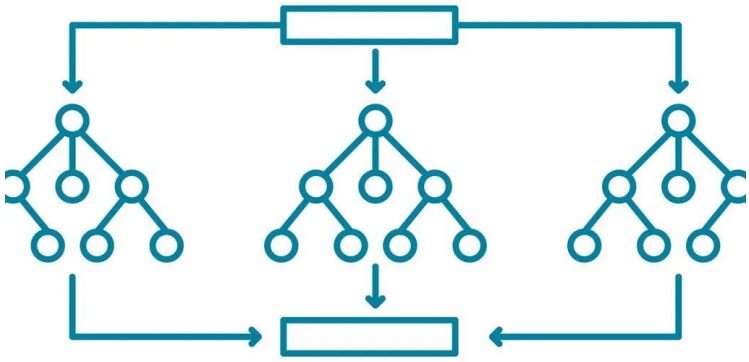
Untuk data yang tidak linier, SVM menggunakan fungsi kernel $K(xi, xj$) untuk memetakan data ke ruang fitur berdimensi lebih tinggi, sehingga memungkinkan pemisahan linier. Fungsi kernel yang umum digunakan meliputi:

- Linear kernel: $K(xi, xj) = xiTxj$
- Polynomial kernel: $K(xi, xj) = (xiTxj + c)d$
- Radial Basis Function (RBF) kernel: $K(xi, xj) = exp(-\gamma \parallel xi - xj \parallel 2)$

Dengan kernel ini, SVM dapat menangani dataset yang tidak terpisahkan secara linier, menghasilkan klasifikasi yang lebih akurat pada dataset kompleks.

2.3.2.2 Random Forest

Random Forest adalah salah satu algoritma pembelajaran mesin berbasis ensemble yang populer digunakan untuk tugas klasifikasi dan regresi [30]. Dikembangkan oleh Leo Breiman dan Adele Cutler, algoritma ini memanfaatkan konsep ensemble learning dengan menggabungkan prediksi dari sejumlah pohon keputusan (decision trees) yang dibangun secara independen pada subset data yang berbeda-beda. Pendekatan ini memungkinkan Random Forest untuk mencapai tingkat akurasi yang lebih tinggi dan stabilitas yang lebih baik dibandingkan dengan pohon keputusan Tunggal [31].



Gambar 2. 3 Random Forest [31]

Gambar 2.3 menggambarkan ensemble learning dalam pembelajaran mesin, khususnya metode bagging atau boosting yang melibatkan penggabungan beberapa model decision tree. Diagram menunjukkan beberapa pohon keputusan independen yang dibangun dari subset data, kemudian hasil prediksi dari setiap pohon digabungkan untuk membentuk prediksi akhir. Dalam bagging, model dibangun secara independen dari bootstrap samples dan prediksi akhir diperoleh dengan metode agregasi (misalnya, voting untuk klasifikasi atau rata-rata untuk regresi).

Rumus untuk prediksi akhir (klasifikasi):

$$y\hat{} = mode(\{fm(x)\}m = 1M)$$

Rumus 2. 2 Rumus untuk prediksi akhir (klasifikasi)

Rumus untuk prediksi akhir (regresi):

$$y\hat{} = M1m = 1\sum Mfm(x)$$

Rumus 2. 3 Rumus untuk prediksi akhir (regresi)

Di mana:

- $y\hat{}$ adalah prediksi akhir.

- M adalah jumlah model (pohon keputusan).
- $fm(x)$ adalah prediksi dari pohon ke-mmm

Sementara dalam metode boosting (misalnya, AdaBoost atau Gradient Boosting), pohon-pohon dilatih secara berurutan, di mana model selanjutnya fokus pada kesalahan yang dibuat oleh model sebelumnya. Teknik ini efektif untuk meningkatkan akurasi prediksi, mengurangi risiko overfitting, dan meningkatkan stabilitas model.

Rumus prediksi akhir untuk boosting:

$$y\hat{} = sign(m = 1\sum M\alpha mfm(x))$$

Rumus 2. 4 Rumus prediksi akhir untuk boosting

Di mana:

- αm adalah bobot untuk model ke-mmm, yang biasanya ditentukan berdasarkan tingkat kesalahan.
- $fm(x)$ adalah prediksi dari model ke-mmm (biasanya $\{-1,+1\}$ untuk klasifikasi biner).
- Sign digunakan untuk mengembalikan label akhir dalam klasifikasi.

Kedua teknik ini bertujuan untuk menggabungkan kekuatan dari beberapa pohon keputusan agar menghasilkan model yang lebih akurat, stabil, dan memiliki kemampuan generalisasi yang baik.[32].

2.3.2.3 Naïve bayes

Algoritma Naive Bayes merupakan salah satu metode dalam pembelajaran mesin yang didasarkan pada Teorema Bayes. Algoritma ini mengasumsikan bahwa semua fitur atau atribut dalam dataset saling independen atau tidak saling bergantung satu sama lain terhadap hasil prediksi [33]. Dalam konteks pembelajaran mesin, pendekatan ini disebut

sebagai “naive” atau sederhana karena asumsi independensi jarang terjadi dalam kasus dunia nyata, terutama pada data yang kompleks[34]. Namun, walaupun asumsi ini cenderung tidak akurat dalam banyak kasus, Naive Bayes tetap menunjukkan performa yang baik dalam beberapa aplikasi, terutama pada pengklasifikasian teks.

Algoritma Naive Bayes memiliki beberapa varian yang disesuaikan untuk berbagai jenis data. Salah satu varian utama adalah *Multinomial Naive Bayes*, yang cocok digunakan untuk data diskrit seperti teks, di mana jumlah kemunculan suatu kata dalam dokumen dapat dihitung.

Rumus probabilitas posterior dalam MNB untuk memprediksi kelas ccc diberikan oleh:

$$P(c \mid x) \propto P(c)i = 1\prod nP(xi \mid c)$$

Rumus 2. 5 Rumus Multinomial Naïve Bayes

Dimana

- P(c): Probabilitas prior dari kelas ccc.
- $P(xi \mid c)$: Probabilitas fitur xi muncul dalam kelas ccc.
- xi : Frekuensi atau jumlah kemunculan fitur iii dalam sampel input.
- n : Jumlah total fitur.

Selanjutnya, *Gaussian Naive Bayes* digunakan untuk data kontinu dan mengasumsikan bahwa setiap fitur mengikuti distribusi normal atau Gaussian. Ada juga *Bernoulli Naive Bayes*, yang biasanya diterapkan pada data biner yang menyatakan kehadiran atau ketidakhadiran suatu fitur dalam data, seperti analisis dokumen biner [35]

Distribusi Gaussian dinyatakan sebagai:

$$P(xi \mid c) = 2\pi\sigma c21exp(-2\sigma c2(xi - \mu c)2).$$

Rumus 2. 6 Rumus Gaussian

Dimana

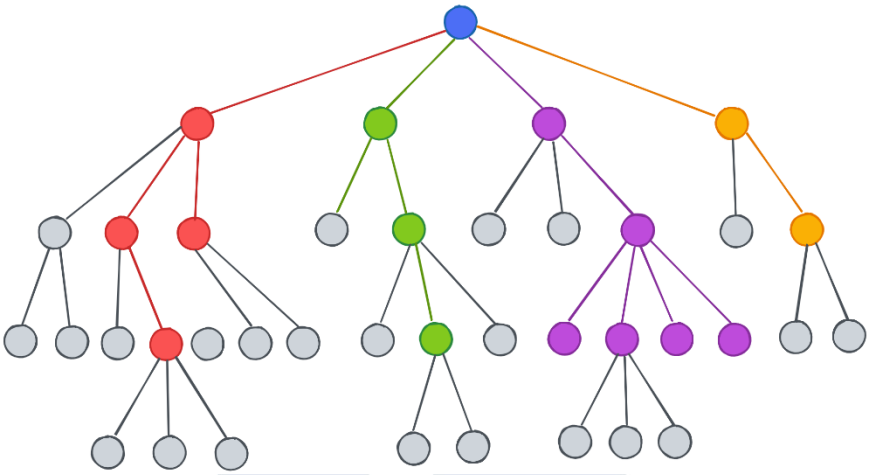
- xi : Nilai fitur ke-i.
- μc: Mean (rata-rata) dari fitur xi pada kelas c.
- $\sigma c2$: Varians dari fitur xi pada kelas c.

Salah satu kelebihan utama dari algoritma Naive Bayes adalah kemampuannya untuk bekerja cepat bahkan pada dataset yang besar. Algoritma ini juga sering menunjukkan performa yang baik pada masalah klasifikasi teks seperti pengenalan spam email dan analisis sentimen. Namun, kelemahan utama dari Naive Bayes adalah asumsi independensi antara fitur-fitur yang ada, yang sering kali tidak valid dalam kasus dunia nyata. Ketika fitur-fitur saling berkorelasi, performa Naive Bayes cenderung menurun dan menghasilkan prediksi yang kurang akurat [36].

2.3.2.4 Decision Tree

Decision Tree adalah salah satu algoritma machine learning yang digunakan dalam metode supervised learning untuk keperluan klasifikasi dan regresi. Algoritma ini berbasis pohon keputusan yang dibentuk dari serangkaian aturan yang berurutan dan bercabang-cabang. Setiap cabang dalam pohon ini mewakili kondisi tertentu, sedangkan setiap simpul daun (leaf node) menunjukkan hasil akhir atau klasifikasi dari data tersebut[37]. Decision Tree banyak digunakan karena mampu memetakan proses pengambilan keputusan secara visual yang mudah dimengerti .

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 2. 4 Decision Tree [37]

Pada gambar 2.4 Decision Tree berfungsi dengan cara membagi dataset secara berulang ke dalam subset-subset berdasarkan fitur atau atribut tertentu, hingga terbentuk pohon keputusan dengan beberapa cabang yang merepresentasikan hasil klasifikasi atau prediksi akhir [38]. Setiap pembagian atau pemisahan (split) data bertujuan untuk mengoptimalkan keputusan atau prediksi di setiap langkahnya. Algoritma ini umumnya menggunakan metode seperti *Gini Index yang sering digunakan sebagai kriteria pemisahan dalam Decision Tree. Gini Index mengukur probabilitas bahwa dua data yang dipilih secara acak memiliki kelas yang berbeda* [16]..

Rumus Gini Index adalah:

$$Gini(S) = 1 - i = 1 \sum cpi2$$

Rumus 2. 7 Rumus Gini Index

- *c: Jumlah kelas dalam dataset.*
- *pi: Proporsi data yang termasuk ke kelas iii.*

Lalu ada Entropy yang digunakan untuk mengukur **t**ingkat ketidakpastian dalam suatu himpunan data. Semakin rendah nilai entropy, semakin "murni" suatu himpunan data.

Rumus entropy untuk satu node adalah:

$$H(S) = -i = 1\sum cpilog2(pi)$$

Rumus 2. 8 Rumus Entropy

- *S: Himpunan data pada node tertentu.*
- *c: Jumlah kelas dalam dataset.*
- *pi : Probabilitas suatu data termasuk ke kelas iii.*

Dan terakhir *Information Gain* yang digunakan untuk mengukur penurunan ketidakpastian (entropy) setelah membagi data berdasarkan suatu atribut. Atribut dengan Information Gain tertinggi akan dipilih untuk memecah node.

Rumus Information Gain adalah:

$$IG(S, A) = H(S) - v \in V\sum | S || Sv | H(Sv)$$

Rumus 2. 9 Rumus Information Gain

- H(S) : Entropy sebelum pemisahan.
- A : Atribut yang digunakan untuk pemisahan.
- V : Nilai-nilai unik dari atribut A.
- Sv : Subset data SSS untuk nilai atribut vvv.
- $| S || Sv |$: Proporsi data pada subset Sv .

Decision Tree digunakan secara luas dalam berbagai aplikasi, seperti prediksi kredit, diagnosis medis, analisis data pelanggan, dan deteksi penipuan. Dalam aplikasi deteksi phishing, misalnya, Decision Tree dapat digunakan untuk memisahkan URL phishing dan URL asli dengan menganalisis pola data yang berbeda, seperti karakteristik URL atau

metadata halaman web. Kombinasi dari berbagai kriteria ini memungkinkan Decision Tree untuk mengenali pola yang spesifik dan relevan dalam proses klasifikasi phishing [39].

2.4 Tools

2.4.1 Python

Python adalah salah satu bahasa pemrograman yang banyak digunakan dalam proyek pengolahan data dan pengembangan aplikasi. Dikembangkan oleh Guido van Rossum di Belanda pada tahun 1990, Python telah menjadi pilihan utama di kalangan industri maupun akademisi [40]. Bahasa ini dikenal karena kemudahan penggunaannya dan sifatnya yang gratis, serta menyediakan berbagai struktur data tingkat lanjut, seperti array, dynamic binding, class, exceptions, list, dan fitur lainnya [41]. Logo Python dapat ditemukan pada gambar 2.5.

Gambar 2. 5 Logo Bahasa Pemrograman Python [41]

Python memiliki sejumlah keunggulan yang menjadikannya berbeda dari bahasa pemrograman lainnya, di antaranya [42]:

- Python menawarkan sintaks yang ringkas dan intuitif, menyerupai struktur Bahasa Inggris, sehingga memudahkan pemahaman dan pembelajaran, terutama

bagi pemula .

- Python bersifat open-source dan dapat dijalankan di berbagai sistem operasi utama, termasuk Windows, Linux, dan macOS, menjadikannya fleksibel untuk berbagai kebutuhan pengembangan.

- Pustaka dan modul yang beragam:Python memiliki ekosistem pustaka yang luas serta modul yang mencakup berbagai bidang, seperti analisis teks, manipulasi tipe data, komputasi numerik, pengembangan perangkat lunak, hingga pembelajaran mesin, yang mendukung implementasi dalam berbagai proyek dan penelitian.

2.4.2 Visual Studio Code

Visual Studio Code (VSCode) adalah salah satu alat pengembangan perangkat lunak yang banyak digunakan untuk menulis dan mengelola kode program dalam berbagai bahasa, termasuk Python, HTML, CSS, PHP, dan JavaScript [43]. Dikembangkan oleh Microsoft, VSCode pertama kali dirilis pada tahun 2015 dan dirancang untuk memenuhi kebutuhan pengembang modern dengan fitur-fitur yang mendukung produktivitas dan efisiensi dalam pengembangan perangkat lunak [44]. Editor ini kompatibel dengan berbagai sistem operasi, seperti Linux, macOS, dan Windows, sehingga dapat diakses oleh berbagai pengguna di lingkungan yang berbeda.

UNIVERSITAS MULTIMEDIA NUSANTARA

Gambar 2. 6 Gambar Logo Visual Studio Code [44]

Keunggulan utama VSCode terletak pada fleksibilitas dan kemampuan penyesuaiannya yang tinggi. Dengan dukungan ekstensi yang luas, pengguna dapat menyesuaikan VSCode untuk memenuhi kebutuhan spesifik dalam proyek mereka. Misalnya, pengguna dapat menambahkan ekstensi untuk pengembangan web, pemrograman data, atau pengembangan aplikasi mobile. Selain itu, VSCode memiliki performa yang ringan dan tidak membebani sistem, sehingga pengguna dapat bekerja dengan lancar, bahkan pada proyek besar. VSCode juga menyediakan pembaruan rutin dan perbaikan bug yang memastikan editor ini tetap relevan dan optimal seiring perkembangan teknologi [45].

Meskipun memiliki banyak kelebihan, VSCode juga memiliki beberapa keterbatasan. Pertama, penggunaan ekstensi yang berlebihan dapat mempengaruhi kinerja editor, terutama pada komputer dengan spesifikasi yang rendah. Selain itu, karena sifatnya sebagai editor teks dan bukan IDE penuh, VSCode mungkin memerlukan lebih banyak konfigurasi untuk mendukung beberapa fitur yang biasanya ditemukan di IDE lengkap seperti

Visual Studio atau IntelliJ IDEA. Selain itu, beberapa pengguna pemula mungkin merasa sulit menavigasi berbagai ekstensi dan fitur konfigurasi yang luas, sehingga membutuhkan waktu untuk penyesuaian.

VSCode digunakan secara luas oleh pengembang dari berbagai disiplin ilmu, termasuk pengembangan web, analisis data, ilmu data, pengembangan aplikasi desktop, dan pengembangan perangkat lunak secara umum. Di bidang pengembangan web, VSCode populer karena mendukung HTML, CSS, dan JavaScript secara langsung, serta menyediakan ekstensi untuk framework seperti React, Angular, dan Vue. Di bidang data science, VSCode digunakan dengan ekstensi untuk Jupyter Notebook dan pustaka seperti pandas atau NumPy, memungkinkan para ilmuwan data untuk mengelola dan menganalisis data secara efisien. Selain itu, VSCode juga digunakan dalam pengembangan backend dan API, terutama ketika digabungkan dengan pustaka dan framework seperti Flask, Django, atau Express.js [46].

2.4.3 Flask

Flask adalah framework web berbasis Python yang bersifat ringan dan mudah digunakan untuk mengembangkan aplikasi web. Framework ini dirancang untuk memfasilitasi pembuatan aplikasi web dengan cepat dan sederhana, dengan menyediakan struktur yang fleksibel serta berbagai fitur dasar yang dibutuhkan untuk membangun aplikasi web [47]. Flask dikenal sebagai "micro-framework" karena tidak menyertakan komponen yang biasanya ada pada framework besar, seperti ORM (Object-Relational Mapping) atau otentikasi bawaan. Hal ini membuat Flask lebih ringan dan memungkinkan pengembang untuk memilih komponen yang sesuai dengan kebutuhan proyek mereka.

Gambar 2. 7 Gambar Logo Flask [47]

Flask memiliki beberapa keunggulan yang membuatnya menjadi framework yang populer di kalangan pengembang. Pertama, fleksibilitas dalam penggunaan berbagai modul eksternal membuatnya cocok untuk aplikasi yang beragam, mulai dari aplikasi kecil hingga besar. Kedua, Flask mendukung ekstensi yang dapat diintegrasikan dengan berbagai komponen tambahan seperti database, otentikasi pengguna, dan sesi. Ketiga, Flask mudah untuk dipelajari dan digunakan, karena dokumentasi yang lengkap serta komunitas yang aktif, menjadikannya pilihan yang ideal bagi pengembang pemula maupun berpengalaman [48].

Salah satu kekurangan Flask adalah ketergantungannya pada komponen eksternal. Karena Flask tidak menyertakan banyak fitur bawaan, pengembang sering kali harus mengandalkan ekstensi pihak ketiga untuk menambahkan fungsionalitas tertentu. Hal ini dapat menyebabkan ketergantungan pada beberapa pustaka eksternal yang mungkin tidak selalu diperbarui atau memiliki standar yang berbeda-beda. Selain itu, dalam proyek besar yang membutuhkan struktur kode yang kompleks, pengembangan dengan Flask bisa menjadi lebih rumit dibandingkan dengan framework yang lebih terstruktur seperti Django [49].

2.4.4 Html

HTML (HyperText Markup Language) adalah bahasa markup standar yang digunakan untuk membuat dan menyusun halaman web. HTML

memungkinkan pengembang untuk menentukan struktur dasar dari sebuah halaman web, termasuk elemen seperti teks, gambar, tautan, dan multimedia. HTML pertama kali dikembangkan pada tahun 1991 oleh Tim Berners-Lee dan sejak saat itu mengalami berbagai perkembangan hingga kini menjadi salah satu bahasa dasar dalam pengembangan web modern [50].

HTML berfungsi sebagai fondasi utama dari halaman web. Setiap elemen dalam HTML disebut "tag" yang memberitahu browser bagaimana menampilkan berbagai jenis konten pada halaman. Tag HTML, seperti <h1>, <p>, dan <a>, memungkinkan pengembang untuk menyusun teks, membuat judul, menghubungkan halaman, dan membentuk struktur hierarki dokumen. Dengan menggunakan HTML, pengembang dapat memastikan bahwa informasi disusun dengan cara yang mudah dibaca dan diakses oleh pengguna, serta memungkinkan browser untuk menampilkan konten sesuai dengan standar web yang ditetapkan.

HTML memiliki beberapa keunggulan utama, yaitu kemudahan penggunaan dan aksesibilitas yang tinggi. Dengan struktur yang sederhana, HTML dapat dipelajari dengan mudah oleh pemula. Selain itu, HTML adalah bahasa dasar yang diakui secara luas dan dapat diakses oleh berbagai browser dan perangkat. Namun, HTML juga memiliki keterbatasan, seperti kurangnya kemampuan untuk menangani fungsionalitas dinamis. Untuk itu, HTML sering dikombinasikan dengan bahasa lain, seperti CSS untuk mempercantik tampilan dan JavaScript untuk menambahkan interaktivitas [51].

HTML adalah dasar dari hampir semua halaman web dan menjadi langkah pertama dalam pengembangan web. Setiap website menggunakan HTML untuk menyusun kontennya, mulai dari website sederhana hingga aplikasi web kompleks. Dalam kombinasi dengan CSS dan JavaScript, HTML menjadi bagian penting dari pengembangan aplikasi web interaktif, website bisnis, blog, e-commerce, dan masih banyak lagi.

2.4.5 CSS

CSS (Cascading Style Sheets) adalah bahasa desain yang digunakan untuk mengatur tampilan dan tata letak elemen dalam halaman web. CSS memungkinkan pengembang web untuk memisahkan struktur konten dari desain visual, sehingga dapat mengubah tampilan tanpa mengubah struktur HTML yang mendasarinya. Dalam model client-server, CSS diunduh dari server web ke browser pengguna, yang kemudian menginterpretasikan kode CSS dan mengaplikasikannya ke elemen-elemen HTML di dalam halaman tersebut [52]. Pemisahan ini memberikan keuntungan dalam pengelolaan situs web yang lebih efektif, terutama dalam desain antarmuka yang konsisten dan responsif di berbagai perangkat. Dengan CSS, pengembang dapat mengatur aspek visual seperti warna, font, margin, padding, posisi, dan ukuran elemen, yang semuanya mendukung pengalaman pengguna yang lebih baik.

Selain itu, CSS bekerja melalui hirarki atau "cascade" dari aturan-aturan gaya yang diterapkan pada elemen-elemen dalam dokumen HTML, di mana prioritas ditentukan berdasarkan spesifisitas dan urutan aturan. Misalnya, aturan yang lebih spesifik, seperti yang ditujukan untuk suatu elemen dengan kelas atau ID tertentu, akan menimpa aturan umum. CSS juga memungkinkan penggunaan konsep seperti inheritance, di mana beberapa gaya dapat diwariskan dari elemen induk ke elemen anak. Dengan kemampuannya ini, CSS memudahkan pembuatan desain yang konsisten, terstruktur, dan fleksibel, sehingga dapat disesuaikan dengan berbagai tampilan layar atau perangkat, menjadikan CSS sebagai komponen penting dalam desain web yang responsif dan ramah pengguna [53].