BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Bab ini menyajikan berbagai teori umum yang menjadi landasan utama penelitian ini. Penjelasan mengenai konsep-konsep dasar tersebut bertujuan untuk memberikan pemahaman mendalam tentang algoritma *naïve bayes* untuk klasifikasi email spam.

2.1.1 Keamanan Jaringan

Keamanan jaringan merupakan salah satu cabang ilmu komputer yang mencakup berbagai aspek, seperti penggunaan kriptografi, pertimbangan dalam desain dan konfigurasi sistem, serta aspek praktis jaringan, termasuk deteksi intrusi komputer dan analisis lalu lintas data (Misra et al., 2021). Tujuan utama dari keamanan jaringan adalah menjaga kerahasiaan, keutuhan, dan ketersediaan data.

Untuk melindungi jaringan, ada beberapa teknologi yang sering digunakan, seperti *firewall*, sistem deteksi serangan (*Intrusion Detection System*), dan enkripsi. *Firewall* membantu membatasi akses dari jaringan luar yang tidak aman, sedangkan enkripsi digunakan untuk memastikan data yang dikirim tidak mudah dibaca oleh orang yang tidak berhak.

Saat ini, ancaman terhadap jaringan semakin meningkat, seperti serangan phishing, penyebaran virus, atau serangan *Denial of Service (DoS).* Oleh karena itu, perusahaan atau organisasi perlu menjaga keamanan jaringan mereka dengan baik,

misalnya dengan membuat kebijakan keamanan, melatih pengguna, dan memantau aktivitas jaringan secara berkala.

2.1.2 Email

Surat elektronik (email) adalah sarana komunikasi yang tersedia di internet, digunakan untuk berbagai tujuan, seperti berbagi informasi melalui lampiran file, berdiskusi dalam grup mailing list, hingga dimanfaatkan sebagai media promosi oleh perusahaan (Wibisono, 2023). Email memungkinkan seseorang untuk mengirim pesan atau informasi dalam bentuk teks, gambar, atau dokumen melalui internet dalam waktu yang sangat cepat. Dibandingkan dengan cara komunikasi konvensional seperti surat pos, email menawarkan kecepatan dan kemudahan yang jauh lebih tinggi, sehingga mempermudah komunikasi baik dalam urusan pribadi maupun profesional.

Email terdiri dari dua komponen utama, yaitu email client (aplikasi untuk mengirim dan menerima email) dan email server (server yang menyimpan dan mendistribusikan pesan). Setiap email memiliki alamat pengirim dan penerima yang unik, yang biasanya mengikuti format standar seperti *"username@domain.com"*. Selain itu, email juga dilengkapi dengan fitur-fitur seperti subject, attachment, dan body, yang memudahkan pengorganisasian pesan dan lampiran yang dikirim.

Dalam dunia bisnis dan organisasi, email menjadi alat komunikasi yang sangat penting karena memungkinkan pertukaran informasi secara langsung dan efisien. Namun, meskipun email menawarkan banyak keuntungan, penggunaannya

juga membawa berbagai tantangan, terutama terkait dengan ancaman seperti email spam dan penipuan.

2.1.3 Spam

Spam merupakan istilah yang mengacu pada pengiriman pesan secara massal kepada banyak penerima, di mana pesan tersebut sering kali tidak diminta dan tidak diinginkan (Wibisono, 2023). Dalam konteks email, spam sering kali berupa iklan, promosi produk, atau bahkan penipuan yang bertujuan untuk mengelabui penerima agar memberikan informasi pribadi atau keuangan.

Selain email, spam juga dapat berupa pesan teks, komentar di media sosial, atau bentuk komunikasi digital lainnya. Pesan-pesan ini biasanya tidak relevan dengan kebutuhan atau minat penerima, sering kali dikirim oleh individu atau organisasi yang tidak dikenal. Konten spam dapat bervariasi, mulai dari promosi barang dan jasa hingga upaya penipuan yang bertujuan mencuri data pribadi atau menyebarkan malware. Meskipun filter spam terus berkembang untuk meminimalkan dampaknya, spam tetap menjadi tantangan dalam dunia digital, karena tidak hanya mengganggu penerima, tetapi juga dapat membebani infrastruktur jaringan dan meningkatkan risiko ancaman keamanan siber.

2.1.4 Machine Learning

Menurut (Fathurohman, 2021), *machine Learning* merupakan bidang yang mempelajari berbagai algoritma yang memungkinkan mesin untuk mempelajari suatu hal dan menjalankan tugas-tugas tertentu secara otomatis, seperti yang biasa dilakukan oleh manusia.

Sedangkan menurut (Munir et al., 2024), menyebutkan bahwa *machine learning* merupakan bidang ilmu yang menawarkan alat dan teknik untuk secara otomatis mengenali pola dari data yang tersedia, meramalkan kejadian di masa depan berdasarkan pola tersebut, atau membuat keputusan dalam situasi yang tidak pasti.

Selain itu, (Taye, 2023) menerangkan bahwa dalam *Machine Learning*, sebuah program komputer diberikan sejumlah tugas untuk diselesaikan, dan dikatakan bahwa mesin tersebut telah "belajar" dari pengalamannya jika kemampuannya yang terukur dalam menyelesaikan tugas-tugas tersebut meningkat seiring waktu, sejalan dengan bertambahnya latihan yang dilakukan untuk menyelesaikan tugas-tugas tersebut.

2.1.5 Algoritma machine learning

Terdapat beberapa macam algoritma dalam machine learning. algoritma ini berperan sebagai prosedur yang mengarahkan model untuk mengidentifikasi pola data dan mengambil keputusan atau tindakan berdasarkan pola data tersebut. beberapa jenis algoritma pada *machine learning* adalah sebagai berikut.

2.1.5.1 SVM

Support Vector Machine (*SVM*) merupakan salah satu metode dalam *machine learning* yang menggunakan data masukan dalam bentuk vektor untuk menyelesaikan masalah klasifikasi dan regresi (Kumari et al., 2023). Dengan prinsip kerjanya yang sederhana namun efektif, *SVM* mampu memanfaatkan data dalam bentuk vektor untuk menyelesaikan berbagai masalah klasifikasi dan regresi,

sekaligus memberikan solusi yang akurat dan efisien melalui penggunaan dua hyperplane paralel untuk memisahkan kelas data.

SVM bekerja dengan mempelajari pola dari data dan menggunakan dua hyperplane paralel untuk memisahkan kelas-kelas data, yang membantu memberikan hasil klasifikasi yang akurat, mengatasi masalah *overfitting*, dan menangani data berdimensi tinggi dengan baik. Selain itu, *SVM* terkenal karena kecepatan dalam pelatihan dan pengujian serta efisiensi penggunaan memori (Akinola et al., 2024).

2.1.5.2 KNN

K-Nearest Neighbors (KNN) adalah algoritma yang mengklasifikasikan data berdasarkan kedekatan dengan tetangga terdekatnya, sehingga mampu meningkatkan performa dalam hal akurasi dan presisi (Vahedifar et al., 2023). KNN dikenal sebagai algoritma yang tidak bergantung pada data pelatihan, sehingga disebut sebagai *lazy learning* sehingga mudah untuk diimplementasikan (Zhang & Li, 2023). Prinsip utama algoritma *KNN* adalah membandingkan data baru dengan data yang sudah ada berdasarkan jarak tertentu seperti jarak euclidean, untuk menentukan hasil prediksi.

2.1.5.3 Random Forest

Random Forest (RF) adalah algoritma *machine learning* berbasis metode ensemble yang menggabungkan hasil dari beberapa pohon keputusan untuk menghasilkan satu output akhir, menjadikannya lebih tangguh dan kompleks dengan kebutuhan daya komputasi yang lebih besar tetapi memberikan performa yang lebih optimal (Ramos Collin et al., 2024). Algoritma ini dibangun dari

kumpulan pohon keputusan (*decision trees*) yang bekerja secara independen. Setiap pohon dilatih dengan subset data dan subset fitur yang dipilih secara acak, sehingga meningkatkan keberagaman model (Kurniawan, 2022).

2.1.5.4 Naïve bayes

Naïve Bayes adalah algoritma klasifikasi berbasis probabilitas yang bekerja dengan menghitung probabilitas berdasarkan frekuensi dan kombinasi nilai dalam suatu dataset tertentu (sahni, 2021). Dengan menekankan kesederhanaan dan efisiensinya, *Naïve Bayes* mengandalkan asumsi "naif" tentang independensi fitur sebagai prinsip intinya (Kumar et al., 2024). Keunggulan dari *Naive Bayes* adalah kemampuannya untuk menangani dataset besar dengan cepat dan sederhana. Hal ini membuatnya sangat cocok untuk tugas-tugas seperti klasifikasi teks, termasuk deteksi spam pada email.

Ciri khas *Naive Bayes* berasal dari anggapan bahwa setiap atribut bersifat independen. Salah satu upaya untuk mengatasi masalah ini adalah melalui pendekatan pemilihan atribut, yang merupakan metode penting dalam mengurangi ketergantungan pada asumsi tersebut (Chen et al., 2020). Algoritma *Naive Bayes* bekerja dengan cara menghitung probabilitas suatu kategori berdasarkan fitur-fitur tertentu yang ada pada data. Fitur-fitur ini bisa berupa kata-kata atau karakteristik lain dari email yang dianalisis. Setelah probabilitas untuk setiap kategori dihitung, algoritma akan memilih kategori dengan probabilitas tertinggi sebagai hasil klasifikasi. Rumus algoritma *naïve bayes* pada penelitian ini dapat dilihat pada persamaan berikut.

$$P(C|X) = \frac{P(X|C) \times P(C)}{PX}$$ **Rumus 1**

Penjelasan:

P(C|X) : Probabilitas posterior C (spam/non spam) pada X (teks email)

P(X|C) : Probabilitas likelihood untuk mendapatkan fitur (X) pada C

P(C) : Probabilitas prior C

P(X) : Probabilitas fitur X

Mengacu pada rumus diatas, langkah-langkah perhitungan algoritma naïve bayes adalah sebagai berikut.

1. Hitung probabilitas prior: Probabilitas prior ialah peluang bahwa sebuah data termasuk dalam sebuah kelas tertentu sebelum mempertimbangkan fitur (kata-kata) didalamnya, misalnya spam dan non spam. Rumus menghitung probabilitas prior adalah sebagai berikut:

   $$P(kelas) = \frac{jumlah\ data\ dalam\ kelas\ tertentu}{total\ data}$$ **Rumus 2**

2. Hitung probabilitas *likelihood*: Probabilitas *likelihood* ialah peluang munculnya sebuah kata dalam sebuah kelas tertentu. *Likelihood* membantu menentukan apakah kata tertentu lebih cenderung muncul di email spam atau ham. Rumusnya adalah sebagai berikut:

   $$P(kata|kelas) = \frac{jumlah\ kemunculan\ kata}{total\ jumlah\ kata}$$ **Rumus 3**

3. Probabilitas posterior: *Naive Bayes* kemudian menghitung probabilitas *posterior* untuk masing-masing kelas dengan mengalikan probabilitas *prior* dan *likelihood*. Rumus yang digunakan adalah:

$$P(kelas|kata) = \frac{P(kelas) \times P(kata\ n|kelas)}{P(kata)}$$ **Rumus 4**

4. Tentukan kelas dengan probabilitas posterior tertinggi: setelah menghitung probabilitas posterior untuk kelas spam dan non spam, algoritma *naïve bayes* mengklasifikasikan data email berdasarkan nilai tertinggi antara probabilitas posterior spam dan non spam.

2.1.6 Confusion Matrix dan Metrix Evaluasi

Confusion matrix bertujuan untuk mengukur performa algoritma naïve bayes dalam mengklasifikasikan email spam atau non spam. Matrix ini memberikan gambaran keseluruhan performa model dengan menampilkan distribusi TP, TN, FP, dan FN. Pada konteks klasifikasi email spam, tabel confusion matrix dapat dijelaskan seperti dibawah ini.

Tabel 2. 1 Tabel Confusion Matrix

	Aktual spam	Aktual non spam
Prediksi spam	TP	FP
Prediksi non spam	FN	TN

Sumber: penelitian 2025

Penjelasan:

1. TP (*True Positive*): jumlah data yang sebenarnya spam dan diprediksi hasilnya benar sebagai spam

2. FP (*False Positive*): jumlah data yang sebenarnya non spam tetapi salah diprediksi sebagai spam
3. FN (*False Negative*): jumlah data yang sebenarnya spam tetapi salah diprediksi sebagai non spam
4. TN (*True Negative*): Jumlah data yang sebenarnya non spam, dan diprediksi hasilnya benar sebagai non spam.

Selain menggunakan *confusion matrix,* performa model juga dihitung dengan beberapa metrix evaluasi seperti akurasi, presisi, recall, dan F1-Score. Penjelasan masing-masing dapat dijelaskan seperti dibawah ini.

1. Akurasi: Akurasi mengukur proporsi prediksi yang benar terhadap total data. Rumus akurasi adalah sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$ **Rumus 5**

2. Presisi: Presisi mengukur seberapa akurat prediksi untuk kelas positif (spam). Rumusnya adalah sebagai berikut:

$$Presisi = \frac{TP}{TP + FP}$$ **Rumus 6**

3. Recall: *Recall* mengukur kemampuan model dalam memprediksi semua data positif. Rumusnya adalah sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$ **Rumus 7**

4. F1-Score: F1-Score merupakan rata-rata harmonis dari presisi dan *recall*. Rumusnya dijelaskan seperti berikut ini.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$ **Rumus 8**

2.2 Software Pendukung

Terdapat beberapa *software* pendukung yang digunakan dalam penelitian ini untuk mengoptimalkan proses pengembangan dan implementasi model *machine learning*. Pada bagian ini, akan dibahas secara mendalam mengenai perangkat lunak yang digunakan dalam penelitian ini, mencakup bahasa pemrograman, *framework*, serta alat bantu lainnya yang mendukung penerapan algoritma machine learning, khususnya dalam konteks klasifikasi email spam.

2.2.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang kuat, dikenal karena kemudahan pembelajarannya dan penerapannya dalam pengembangan dunia nyata, menjadikannya pilihan ideal untuk proses pembelajaran yang cepat serta pengembangan aplikasi praktis (Sharma et al., 2020). *Python* memungkinkan pengembang untuk menulis kode yang bersih dan efisien dengan sedikit baris kode. *Python* memiliki ekosistem yang sangat luas dan mendukung berbagai paradigma pemrograman, seperti pemrograman prosedural, berorientasi objek, dan fungsional.

Salah satu keunggulan utama *Python* adalah kemampuannya untuk digunakan di berbagai bidang, mulai dari pengembangan web, analisis data, kecerdasan buatan, hingga otomatisasi tugas (Kurniawan, 2022). *Python* memiliki berbagai pustaka, seperti *NumPy*, *Pandas*, *Matplotlib*, dan *TensorFlow*, yang memungkinkan pengembang untuk menangani berbagai jenis tugas tanpa perlu menulis banyak kode dari awal.

Pustaka yang luas ini juga berguna pada pengembangan *machine learning* karena dapat mendukung pengembangan model secara efisien. Contohnya, pustaka seperti *pandas* dan *numpy* memungkinkan kita memanipulasi data secara cepat dan efisien. Untuk membantu memahami pola data sebelum melatih model, dapat menggunakan *matplotlib* dan *seaborn* untuk membuat grafik, diagram, dan *heatmap*, yang penting dalam eksplorasi data. Selain itu pustaka seperti *scikit learn* menyediakan implementasi berbagai algoritma *machine learning*, mulai dari regresi, klasifikasi, hingga *clustering*.

2.2.2 Pycharm

PyCharm dikembangkan oleh perusahaan *JetBrains*, *PyCharm* menawarkan berbagai fitur yang mempermudah proses pengembangan, debugging, dan pengujian aplikasi *Python*. *IDE* ini sangat populer di kalangan pengembang *python* karena memberikan dukungan lengkap untuk pengembangan perangkat lunak, terutama dalam proyek-proyek skala besar dan kompleks.

Salah satu fitur utama *PyCharm* adalah kemampuannya untuk menyediakan auto-completion atau penyelesaian otomatis kode, yang memungkinkan pengembang menulis kode dengan lebih cepat dan mengurangi kemungkinan kesalahan penulisan. Selain itu, *PyCharm* juga dilengkapi dengan fitur debugging yang sangat canggih, memungkinkan pengembang untuk menganalisis dan memperbaiki masalah pada kode secara efisien. Fitur refactoring di *PyCharm* juga memungkinkan pengembang untuk dengan mudah mengubah struktur kode tanpa merusak fungsionalitasnya.

Pycharm tersedia dalam dua versi, yaitu versi gratis dan versi berbayar. *PyCharm* sangat fleksibel serta dapat disesuaikan dengan kebutuhan pengembangan berbagai jenis aplikasi. Dengan fitur-fitur lengkap, *PyCharm* telah menjadi pilihan utama bagi banyak pengembang *Python* dalam mengelola proyek mereka secara efisien.

2.2.3 Jupyter Notebook

Jupyter Notebook adalah aplikasi web *open-source* yang digunakan untuk membuat dan berbagi dokumen yang berisi kode, visualisasi, dan teks naratif. *Jupyter* memungkinkan pengguna untuk menggabungkan pemrograman dengan dokumentasi secara interaktif, yang menjadikannya alat yang sangat populer di kalangan ilmuwan data, peneliti, dan pengembang. *Jupyter* awalnya dikembangkan sebagai bagian dari proyek *Python*, dan kini mendukung berbagai bahasa pemrograman selain *Python*, seperti *R*, *Julia*, dan banyak lainnya.

Salah satu fitur utama dari *Jupyter Notebook* adalah kemampuannya untuk menjalankan kode secara langsung dalam sel-sel terpisah. Setiap sel dapat berisi kode yang dapat dieksekusi secara terpisah, dan outputnya akan langsung ditampilkan di bawah sel tersebut. Hal ini memungkinkan pengguna untuk bereksperimen dengan kode dan melihat hasilnya secara langsung tanpa perlu menjalankan seluruh program atau skrip.

Jupyter Notebook mendukung berbagai visualisasi data, seperti grafik, tabel, dan gambar, yang sangat berguna untuk analisis data dan pemodelan statistik. Pustaka seperti *Matplotlib*, *Seaborn*, dan *Plotly* memungkinkan pengguna untuk

menyajikan data secara visual, membantu dalam pemahaman dan komunikasi hasil analisis.

2.3 Penelitian Terdahulu

Berikut beberapa penelitian terdahulu yang memiliki keterkaitan langsung dengan topik yang dibahas pada penelitian ini.

Tabel 2. 2 Penelitian Terdahulu

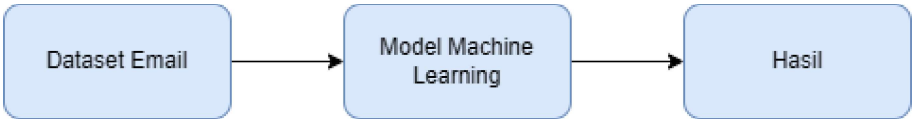
No	Author	Judul	Algoritma	hasil	kesimpulan
1	(Bachri & Gunawan, 2024)	Deteksi email spam menggunakan algorima *CNN*	*CNN*	Hasilnya menunjukkan keefektivitasan CNN yang signifikan dalam mengklasifikasikan email dengan tingkat akurasi tinggi sebesar 99.67% untuk data uji 20%, 99.64% untuk data uji 30%, dan 99.63% untuk data uji 40%.	dapat disimpulkan bahwa model *CNN* yang dirancang berhasil mempelajari representasi yang bermakna dari teks email dan mengklasifikasikannya ke dalam kategori spam dan ham dengan akurasi yang tinggi,
2	(Jaiswal et al., 2021)	*Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU*	*Naïve Bayes*	Hasil menunjukkan akurasi pelatihan 99,67% dan pengujian 99,03%. Waktu pelatihan di gpu 1,361 detik dan cpu 2,029. Sedangkan hasil pengujian di GPU 1,978 detik dan CPU 2,280 detik.	Kesimpulannya adalah *GPU NVIDIA* P100 dapat mempercepat proses pelatihan dan pengujian dan mendapat akurasi lebih tinggi daripada algoritma naïve bayes konvensional.

3	(Sari & Mahalisa, 2024)	*Naive Bayes Classifier* Untuk Deteksi Email Spam	*Naïve Bayes*	Hasil evaluasi menunjukkan bahwa model yang dibangun mampu mencapai akurasi sebesar 97% dalam mengklasifikasikan email sebagai spam atau non-spam.	algoritma *Naive Bayes* dapat menjadi pilihan yang baik untuk deteksi email spam, namun perlu dilakukan penelitian lebih lanjut untuk mengatasi tantangan seperti evolusi teknik spammer dan ketidakseimbangan data.
4	(Qodariyah Fitriyah & Oktaviant o, 2019)	Deteksi Spam Pada Email Berbasis Fitur Konten Menggunakan *Naïve Bayes*	*Naïve Bayes*	Algoritma *Naive Bayes* menghasilkan akurasi 84,8%, dengan 3903 email terklasifikasi benar dan 698 salah. *Precision* dan *recall* masing-masing 0,86 dan 0,85.	*Naive Bayes* efektif untuk klasifikasi spam email, dengan performa terbaik pada k=9 menggunakan *k-fold cross-validation*.
5	(Firmansyah et al., 2025)	Penerapan Algoritma *Naive Bayes* Dengan *Chi-Square* Untuk Klasifikasi Spam Email Berbasis Kata Dan Frekuensi	*Naïve Bayes*	Algoritma *Naive Bayes* dengan *Chi-square* mencapai akurasi 81.00%, *precision* 100%, *recall* 65%, dan *F1-score* 79%. Nilai *AUC* sebesar 0.91	*Naive Bayes* dengan *Chi-square* efektif untuk klasifikasi spam email, dengan precision tinggi yang menghindari kesalahan positif. Namun, recall perlu ditingkatkan untuk mendeteksi lebih banyak spam.

Sumber: data penelitian 2025

2.4 Kerangka Pemikiran

Kerangka pemikiran merupakan sebuah diagram atau alur logis yang menggambarkan langkah-langkah dalam menyelesaikan suatu masalah atau mencapai tujuan penelitian. Kerangka pemikiran dirancang untuk memberikan gambaran yang sistematis tentang bagaimana proses pemecahan masalah dilakukan. Kerangka pemikiran pada penelitian ini adalah sebagai berikut:



Gambar 2. 1 Kerangka Pemikiran

Sumber: data penelitian 2025

Secara sistematis, penelitian ini diawali dengan tahap input, yaitu penggunaan dataset email yang telah dikategorikan sebagai spam dan non-spam. Dataset ini berfungsi sebagai sumber utama bagi model *machine learning* dalam mempelajari karakteristik serta pola yang membedakan kedua jenis email tersebut.

Pada tahap proses, berbagai teknik pemrosesan data diterapkan. Proses ini mencakup *preprocessing*, seperti tokenisasi teks, pembersihan dari karakter tidak relevan, penghapusan kata-kata umum yang tidak memiliki makna signifikan, serta lemmatisasi. Selanjutnya, teks yang telah diproses dikonversi menjadi representasi numerik menggunakan metode *TF-IDF*. Setelah itu, model dilatih menggunakan algoritma *Naïve Bayes* untuk membangun sistem klasifikasi email berdasarkan pola yang diperoleh dari dataset.

Tahap akhir adalah output, yaitu hasil klasifikasi email berdasarkan model yang telah dikembangkan. Model ini akan menghasilkan prediksi terhadap kategori email yang diterima, apakah termasuk spam atau non-spam. Selain itu, evaluasi model dilakukan menggunakan berbagai metrik untuk mengukur tingkat akurasi, presisi, dan recall, sehingga efektivitas model dapat dinilai secara objektif.