BAB II

LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Berita Hoaks

Berita hoaks bukan fenomena baru, keberadaannya telah tercatat sejak seblum masehi (SM) [11]. Penyebarannya telah teridentifikasi sejak tahun 1439, menjadi semakin meluas melalui berbagai media tradisional seperti artikel surat kabar dan televisi [12]. Pertumbuhan pesat *World Wide Web* (WWW) pada pertengahan tahun 1990-an memicu ledakan dalam penyebaran berita palsu, dan dengan kenyamanan, keceparan, dan biaya rendanya, media sosial menjadi medium utama untuk interaksi manusia online dan transmisi informasi [13].

Berita hoaks, didefinisikan sebagai "kebenaran yang diubah," mencerminkan motivasi media yang kadang-kadang terdistorsi. Penyebaran berita palsu terlihat meningkat selama pemilihan presiden AS tahun 2016, dengan pembuatan 19 juta profil bot untuk menyebarkan informasi palsu tentang kandidat Trump dan Clinton. Meskipun ada 8.711.000 interaksi *shared*, *comments*, dan opini di *Facebook* tentang 20 artikel palsu yang paling banyak didiskusikan, berita palsu semakin meluas dan sering kali disebarkan melalui media sosial, mengalahkan platform tradisional [3].

Contohnya adalah kota Veles, Makedonia, di mana ratusan anak muda memproduksi informasi palsu yang disebarkan melalui media massa bersama, menghasilkan keuntungan melalui iklan *pay-per-click* selama pemilihan presiden AS [14]. Dampak berita hoaks tidak hanya terbatas pada individu yang lebih terlibat dalam percakapan politik dan nilai pasar global, tetapi juga dapat berdampak pada kesehatan psikologis, menyebabkan ketegangan dan ketakutan. Situasi ini lebih serius daripada jenis data lainnya, karena karakteristik berulangnya dapat membuat orang mempercayainya, terlepas dari kebenarannya.

Berita hoaks bukan hanya masalah teknis, tetapi juga melibatkan pandangan dan sikap yang dapat dibentuk oleh kondisi sosial. Dengan sengaja diciptakan untuk mengecoh dengan memanipulasi fakta dan data, berita hoaks sering kali meniru

pola keinginan untuk menipu publik. Identifikasi berita hoaks menjadi semakin sulit, meskipun upaya telah dilakukan melalui layanan pemeriksa fakta online dan pendekatan manual oleh para ahli. Volume besar materi palsu yang dihasilkan dan diterbitkan melalui jaringan sosial online juga menjadi tantangan tersendiri [15].

2.1.2 Penyebar Berita Hoaks

Penyebar berita hoaks merujuk pada individu atau kelompok yang sengaja menyebarkan informasi palsu atau disinformasi dengan tujuan tertentu. Penyebar berita hoaks bisa beroperasi melalui berbagai platform, termasuk media sosial, situs web, atau aplikasi pesan. Salah satu aplikasi yang paling sering digunakan oleh penyebar berita hoaks adalah aplikasi Twitter atau X. Beberapa ciri umum akun yang menyebar berita hoaks yaitu, memiliki pola posting yang sangat serupa atau bahkan identik dari waktu ke waktu, memiliki sedikit atau tidak ada informasi terverifikasi tentang pemiliknya, seringkali tidak memiliki riwayat atau sejarah yang dapat diverifikasi secara jelas, postingan cenderung sensasional atau provokstif, memiliki jumlah pengikut yang besar, tetapi keterlibatan (*engagement*) yang rendah dalam bentuk *like*, *retweet*, atau komentar, mengindikasikan bahwa jumlah pengikut mungkin tidak organik atau berasal dari aktivitas palsu, sering memposting informasi yang tidak diverifikasi, terlibat dalam menyebarkan konten yang kontroversial atau konspiratif, cenderung fokus pada satu tema atau jenis konten tertentu, seringkali terhubung dengan jaringan akun lain yang memiliki ciri-ciri serupa, dan lain sebagainya [16].

2.1.3 Data Mining

Data Mining adalah kegiatan penyaringan melalui kumpulan data yang besar untuk mengidentifikasi atau menyoroti korelasi dan pola minat. Hasil dari *data mining* memiliki potensi untuk meningkatkan model bisnis dengan melakukan analisis data yang mendalam, terutama melalui analisis data yang ekstensif. Perusahaan sering menggunakan pendekatan *data mining* untuk menggali informasi tambahan, membuat keputusan praktis, dan melakukan prediksi masa depan terkait trennya. Pendekatan ini khususnya efektif dalam skenario di mana data besar yang

memiliki nilai signifikan bagi organisasi perlu dievaluasi untuk mendukung pengambilan keputusan yang cepat.

Inti dari *data mining* adalah menemukan pengetahuan dalam data, mengatasi data yang bising, tidak lengkap, dan ambigu menggunakan algoritma kecerdasan buatan, serta menggunakan algoritma data mining untuk menggali potensi dan informasi tersembunyi yang bernilai. Tujuan utama dari *data mining* adalah mengekstrak data yang bernilai dan baru dari berbagai jenis data yang berbeda, serta menganalisis data lebih lanjut untuk memahami hubungan antar individu data dan membangun model pendukung keputusan. Proses dasar *data mining* melibatkan identifikasi masalah, pengumpulan informasi data, *preprocess data*, pelaksanaan *data mining*, dan akhirnya ekspresi serta interpretasi model [17].

2.1.4 Social Network Analysis

Social Network Analysis (SNA) adalah metode yang digunakna untuk menggambarkan dan menganalisis interaksi serta hubungan yang terjadi secara terus-menerus antar individu dalam suatu jaringan [18]. SNA, pada dasarnya, merupakan alat analisis yang memungkinkan penggambaran dan pemahaman lebih dalam terhadap struktur jaringan komunikasi [19]. Dengan menggunakan SNA, peneliti dapat mengidentifikasi dan mengevaluasi peran serta pengaruh aktor-aktor dalam topik atau konteks tertentu dalam jaringan tersebut. Dengan fokus pada hubungan dan pola interaksi, SNA memberikan wawasan yang berharga dalam menganalisis dinamika sosial dan struktur jaringan, memungkinkan pemahaman yang lebih mendalam tentang bagaimana individua tau entitas berinteraksi dalam suatu komunitas atau organisasi.

2.1.4.1Degree Centrality

Degree Centrality merupakan suatu ukuran yang menilai tingkat konektivitas atau keeratan suatu node di dalam jaringan (*network*). [20] mendefinisikan *degree centrality* sebagai jumlah koneksi yang dimiliki oleh sebuah node. Dalam konteks ini, aktor yang memiliki jumlah koneksi terbanyak dianggap sebagai aktor yang paling penting dalam jaringan. Dengan kata lain, node yang memiliki nilai *degree*

centrality terbedar merupakan node yang memiliki pengaruh yang signifikan dalam mempertahankan konektivitas dalam jaringan.

Formula matematis yang digunakan untuk menghitung *degree centrality* adalah sebagai berikut:

$$C_d(v) = \frac{degree\ of\ node\ v}{total\ number\ of\ nodes} \quad 2.1$$

$C_d(v)$ adalah *degree centrality* dari simpul (*node*) *v*,

degree of node v adalah jumlah tepi (*edges*) yang terhubung ke simpul *v*,

total number of nodes adalah jumlah total simpul dalam jaringan.

2.1.4.2Betweenness Centrality

Betweenness Centrality adalah sebuah ukuran yang menunjukkan sejauh mana suatu simpul (*node*) berperan sebagai mediator atau penghubung jaringan. Ukuran ini juga memperlihatkan tingkat kepentingan suatu *node* sebagai jembatan antar kelompok-kelompok dalam jaringan. Mirip dengan *closeness centrality*, *betweenness centrality* juga memiliki koefisien nilai yang berkisar antara 0 hingga 1. Semakin signifikan peran sebuah node sebagai penghubung antarkelompok dalam jaringan.

Formula matematis yang digunakan untuk menghitung *betweenness centrality* adalah sebagai berikut:

$$C_b(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad 2.2$$

$C_b(v)$ adalah *betweenness centrality* dari simpul v,

σ_{st} adalah jumlah jalur terpendek dari simpul *s* ke simpul *t*,

$\sigma_{st}(v)$ adalah jumlah jalur terpendek dari simpul *s* ke simpul t yang melibatkan simpul *v*.

Menurut [21], *betweenness centrality* memberikan indikasi bahwa sebuah node memiliki peran penting jika berfungsi sebagai *bottleneck* komunikasi. Ukuran ini dapat digunakan untuk mengidentifikasi *boundary spanners*, yaitu actor atau *node*

yang bertindak sebagai penghubung atau jembatan antara komunitas-komunitas dalam jaringan.

Melalui nilai *betweenness centrality*, peneliti dapat mengidentifikasi dan menilai peran penting suatu simpul sebagai penghubung atau mediator dalam mengelola aliran informasi antar kelompok jaringan.

2.1.4.3Closeness Centrality

Closeness Centrality merupakan suatu metrik yang mengukur sejauh mana suatu simpul (*node*) dalam jaringan mendekati simpul-simpul lainnya [22]. Koefisien dari *closeness centrality* berkisar antara 0 hingga 1. Semakin mendekati nilai 1, semakin dekat hubungan antar simpul dalam jaringan, yang menunjukkan tingkat kedekatan atau kepadatan yang tinggi. Dalam konteks ini, kedekatan yang tinggi menandakan bahwa suatu simpul memiliki jarak yang lebih singkat ke simpul-simpul lainnya dalam jaringan.

Formula matematis yang digunakan untuk menghitung *closeness centrality* adalah sebagai berikut:

$$C_c(v) = \frac{1}{\sum_u d(v,u)} \quad 2.3$$

$C_c(v)$ adalah *closeness centrality* dari simpul *v*,

d(*v*,*u*) adalah panjang jalur terpendek antara simpul *v* dan simpul *u*.

Dengan menggunakan *closeness centrality*, peneliti dapat mengevaluasi seberapa efisien suatu simpul dalam menyebarkan informasi ke seluruh jaringan, dan semakin tinggi nilai *closeness centrality*, semakin efisien simpul tersebut dalam berinteraksi dan menyebarkan informasi dalam konteks jaringan.

2.1.4.4Eigenvector Centrality

Eigenvector Centrality adalah suatu ukuran yang memberikan nilai *centrality* tertinggi pada sebuah simpul (*node*) yang terhubung dengan simpul-simpul lain yang juga memiliki *centrality* tinggi [23]. Nilai dari *eigenvector centrality* mendekati atau sama dengan 1, hal ini menunjukkan bahwa simpul tersebut memiliki keterkaitan yang kuat dengan banyak simpul (aktor) dalam jaringan.

Formula matematis yang digunakan untuk menghitung *eigenvector centrality* adalah sebagai berikut:

$$Ax = \lambda x \quad (2.4)$$

A adalah matriks ketetanggaan (*adjacency matrix*) dari jaringan,

x adalah vector *eigenvector centrality* yang ingin dihitung,

λ adalah nilai *eigen* yang sesuai dengan vector *eigenvector centrality* x.

Eigenvector centrality memberikan pandangan tentang sejauh mana suatu simpul berada dalam posisi yang strategis dalma jaringan dengan berhubungan dengan simpul-simpul penting lainnya. Semakin tinggi nilai *eigenvector centrality*, semakin kuat hubungan dan pengaruh simpul tersebut dalam konteks jaringan.

2.1.5 Machine Learning

Machine Learning (ML) merupakan salah satu bidang kunci dalam penelitian yang bertujuan membantu komputer belajar dari pengalaman dan membuat prediksi yang akurat terhadap peristiwa di masa depan. ML digunakan untuk mengajari mesin bagaimana mengelola data dengan lebih efisien. Terkadang, setelah melihat data, kita tidak dapat menginterpretasikan informasi yang diekstraksi dari data tersebut. Dalam situasi seperti ini, ML diterapkan. Dengan meningkatnya ketersediaan dataset, permintaan untuk ML semakin meningkat. Banyak industry menerapkan ML untuk mengekstrak data yang relevan. Tujuan dari ML adalah untuk belajar dari data. Sejumlah penelitian dilakukan tentang bagaimana membuat mesin belajar sendiri tanpa diprogram secara eksplisit.

ML bergantung pada berbagai algoritma untuk menyelesaikan masalah data. Ilmuwan data menekankan bahwa tidak ada satu algoritma yang cocok untuk semua jenis masalah. Jenis algoritma yang digunakan tergantung pada jenis masalah yang ingin diselesaikan, jumlah variabel, jenis model yang paling sesuai, dan faktor-faktor lainnya.

2.1.5.1 Supervised Learning

Pembelajaran terawasi (*supervised learning*) adalah paradigma pembelajaran mesin di mana algoritma diajarkan dan diberi petunjuk menggunakan data yang sudah diberi label. Dalam hal ini, setiap contoh dat apelatihan memiliki label atau

jawaban yang diketahui, dan algoritma menggunakan informasi ini untuk memahami hubungan antara input dan output. Tujuan dari *supervised learning* adalah untuk menghasilkan model atau fungsi yang dapat melakukan prediksi atau klasifikasi pada data baru yang belum pernah dilihat sebelumnya. Proses ini melibatkan pemahaman pola dari data pelatihan untuk membuat prediksi yang akurat pada data baru [24].

2.1.5.1.1 Algoritma Naïve Bayes

Naïve Bayes adalah teknik klasifikasi dalam *machine learning* yang didasarkan pada Teorema Bayes. Dalam konteks ini *naïve* mengindikasikan asumsi bahwa semua predictor atau fitur yang digunakan dalam model adalah independent satu sama lain. Dengan kata lain, teknik klasifikasi Naïve Bayes menganggap bahwa keberadaan suatu fitur tidak bergantung pada keberadaan fitur lain dalam suatu kelas. Pendekatan ini mempermudah perhitungan probabilitas karena mengasumsikan independensi antara fitur-fitur tersebut.

Pada dasarnya, metode ini memerlukan pemahaman probabilitas dasar untuk menghitung kemungkinana suatu kejadian terjadi berdasarkan kejadian lainnya. Meskipun asumsi kemandirian fitur dapat terlalu sederhana untuk beberapa situasi di dunia nyata, Naïve Bayes seringkali memberikan hasil yang baik dalam berbagai jenis tugas klasifikasi.

Formula dasar yang digunakan Naïve Bayes adalah Teorema Bayes:

$$P(B) = \frac{P(A) \times P(A)}{P(B)} \quad 2.5$$

P(A|B) adalah probabilitas posterior dari A terjadi jika B telah terjadi,

P(B|A) adalah probabilitas *likelihood* dari B terjadi jika A telah terjadi,

P(A) adalah probabilitas *prior* dari A,

P(B) adalah probabilitas *prior* dari B.

Secara umum, Naïve Bayes cocok digunakan ketika prediktor-prediktor dalam suatu masalah dapat dianggap sebagai independen, dan pendekatan ini telah berhasil diterapkan dalam berbagai konteks, termasuk pemrosesan bahas alami, klasifikasi dokumen, dan analisis sentimen. Meskipun sederhana, kecepatan dan

kinerja yang baik membuat Naïve Bayes menjadi pilihan yang popular dalam banyak aplikasi *machine learning* [25].

2.1.5.2.2 Algoritma Logistic Regression

Logistic Regression, meskipun disebut *regression*, sebenarnya merupakan metode klasifikasi yang digunakan untuk memodelkan hubungan antara variable independen dengan probabilitas kejadian pada kategori biner. Secara khusus, logistic regression cocok untuk memprediksi apakah suatu observasi termasuk dalam satu kategori atau yang lain.

Teknik ini memanfaatkan fungsi *logistic* atau *sigmoid* untuk menyatukan nilai-nilai input menjadi rentang antara 0 dan 1, yang dapat diinterpretasikan sebagai probabilitas. Meskipun logistic regression sering digunakan untuk kasus biner, yaitu klasifikasi dua kategori, adaptasinya yang lebih luas memungkinkan penggunaan pada masalah klasifikasi multikelas.

Fungsi *sigmoid* didefinisikan sebagai:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}} \quad 2.6$$

$P(y = 1 \mid x)$ adalah probabilitas bahwa variabel dependen y adalah 1 untuk nilai-nilai yang diberikan dari variabel independent x,

e adalah konstanra *Euler*,

$\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ adalah koefisien yang harus diestimasi dari data pelatihan,

$x_0, x_1, x_2, \ldots, x_n$ adalah nilai-nilai dari variabel independen.

Untuk memperkirakan nilai koefisien tersebut, digunakan metode seperti *Maximum Likelihood Estimation* (MLE) atau *Gradien Descent*. Tujuannya adalah untuk menemukan koefisien yang memaksimalkan kemungkinan munculnya data yang diamati berdasarkan model.

Kelebihan logistic regression meliputi efesiensinya dalam kasus kumpulan data besar, kemampuannya menangani korelasi antar variabel independen, dan interpretabilitas yang baik terkait dengan peluang hasil klasifikasi. Pada intinya, logistic regression memberikan cara yang sederhana namun kuat untuk memodelkan dan memahami hubungan antar variabel [25].

2.1.6 Evaluasi

2.1.6.1Time Processing

Time Processing adalah parameter kritis yang mencerminkan jumlah waktu yang diperlukan dalam melakukan pemrosesan suatu pemodelan menggunakan algoritma *machine learning* hingga mencapai tingkat akurasi prediksi yang diinginkan. Pemantauan dan evaluasi waktu *processing* membantu dalam penyesuaian strategi dan optimalisasi untuk memastikan bahwa penggunaan sumber daya waktu seefesien mungkin tanpa mengorbankan kualitas prediksi.

2.1.6.2Akurasi

Akurasi merupakan metrik penting dalam evaluasi kinerja model klasifikasi pada proyek *data mining*, diukur sebagai persentase prediksi yang tepat terhadap data uji [26]. Akurasi adalah parameter evaluasi yang paling intuitif, mewakili rasio observasi yang ada. Akurasi memberikan gambaran yang jelas tentang sejauh mana model mampu membuat prediksi yang benar, dan nilai tinggi pada metrik ini mengindikasikan tingkat keandalan model dalam mengklasifikasikan data uji. Evaluasi akurasi menjadi fundamental dalam memastikan bahwa model klasifikasi dapat diandalkan dalam mengidentifikasi dan mengelompokkan data dengan tingkat ketepatan yang optimal.

$$\text{Akurasi} = \frac{(TP+TN)}{(TP+FP+FN+TN)} \qquad 2.7$$

2.1.6.3Presisi

Presisi dihitung sebagai rasio antara jumlah sampel *True Positive* dengan jumlah total sampel yang diklasifikasikan sebagai positif baik benar atau salah. Presisi adalah ukuran yang menilai akurasi model dalam mengklasifikasikan sampel sebagai positif. Tujuan utama presisi adalah untuk memastikan bahwa dari semua sampel yang diklasifikasikan sebagai positif, sebanyak mungkin di antaranya adalah benar positif, sementara meminimalkan jumlah kesalahan mengklasifikasikan sampel negatif sebagai positif [26], [27]. Dengan kata lain, presisi berfokus pada ketepatan model dalam mengenali dan memisahkan dengan akurat sampel yang sesuai dengan kelas positif. Semakin tinggi nilai presisi,

semakin baik model dalam mencegah kesalahan dalam mengklasifikasikan sampel negatif sebagai positif.

$$\text{Presisi} = \frac{TP}{(TP+FP)} \quad 2.8$$

2.1.6.4Sensitivitas

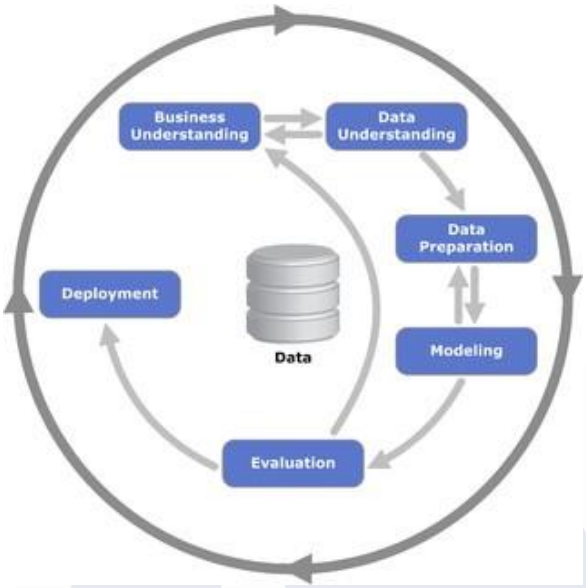
Sensitivitas, atau yang dikenal juga sebagai *recall*, merupakan ukuran yang mengindikasikan sejauh mana system berhasil dalam mendeteksi informasi yang positif. Sensitivitas dihitung dengan membandingkan jumlah sampel positif yang diklasifikasikan dengan benar sebagai positif dengan jumlah keseluruhan sampel positif yang sebenarnya. Dengan kata lain, sensitivitas mengukur kemampuan model untuk mengidentifikasi dan mendeteksi sampel positif dengan akurasi yang tinggi. Semakin tinggi nilai sensitivitas, semakin baik model dalam mendeteksi sampel positif, menunjukkan tingkat keberhasilan yang lebih besar dalam mengenali informasi yang dianggap relevan atau positif oleh sistem [26], [27].

$$\text{Sensitivitas} = \frac{TP}{(TP+FN)} \quad 2.9$$

2.2 Framework

2.2.1 CRISP-DM

Pada penelitian ini, peneliti menggunakan *framework* CRISP-DM (*Cross-Industry Standard Process for Data Mining*). CRISP-DM adalah model proses yang digunakan dalam *data mining*. Alasan penggunaan CRISP-DM daripada KDD adalah karena CRISP-DM memberikan kerangka kerja yang terstruktur dan berulang untuk mengelola proyek *data mining* dari awal hingga akhir. CRISP-DM (Gambar 2.2) sering digunakan di berbagai industri karena fleksibilitasnya dan dapat diterapkan pada berbagai jenis proyek *data mining*. Pada tahun 2007, 2014, dan 2020, CRISP-DM menduduki peringkat teratas dalam *KDNuggets Poll on Data Mining Methodology* dan survei *Data Science Project Management* di Amerika Serikat [28]. Hal ini menunjukkan bahwa model ini tetap menjadi salah satu kerangka kerja yang populer dan dihargai dalam praktik *data mining*. Tabel 2.1 Menunjukkan deskripsi singkat mengenai tiap fase.



Gambar 2.1 Proses CRISP-DM

Tabel 2.1 Deskripsi Fase CRISP-DM [29]

Fase	Deskripsi
Business Understanding (Pemahaman Bisnis)	Pada tahap ini, tujuan bisnis dari proyek *data mining* ditetapkan. Situasi bisnis harus dinilai untuk mendapatkan gambaran tentang sumber daya yang tersedia dan yang dibutuhkan. Tahap ini melibatkan pemahaman mendalam tentang masalah yang dihadapi dan cara solusi *data mining* dapat memberikan nilai tambah.
Data Understanding (Pemahaman Data)	Pada tahap ini, dilakukan serangkaian langkah untuk memperoleh pemahaman yang mendalam terhadap data yang akan digunakan dalam proyek. Langkah-langkah ini melibatkan pengumpulan data dari sumber-sumber yang relevan, ekplorasi data untuk mengidentifikasi pola atau tren awal, serta deskripsi menyeluruh untuk menggambarkan karakteristik data. Analisis statistik digunakan untuk merinci atribut-atribut kunci dan struktur data, memastikan bahwa data berkualitas dan sesuai dengan kebutuhan proyek. Tahap ini merupakan dasar penting dalam memahami konteks dan potensi data sebelum memasuki tahap berikutnya dalam proses *data mining*.
Data Preparation (Persiapan Data)	Pada tahap ini, penekanan diberikan pada pemilihan data yang cermat dengan menetapkan kriteria inklusi dan eksklusi yang sesuai. Proses ini melibatkan pemilahan data yang relevan dengan tujuan proyek, serta menangani kualitas data yang buruk melalui teknik pembersihan data. Metode pembersihan yang digunakan dapat bervariasi sesuai dengan jenis model

	yang diterapkan, memastikan bahwa data yang digunakan dalam proyek *data mining* sesuai dengan standar kualitas yang diperlukan untuk hasil yang akurat dan dapat diandalkan.
Modeling	Pada tahap ini, fokus utama dalah pada pemilihan teknik pemodelan yang paling sesuai untuk mencapai tujuan proyek. Proses ini mencakup pembangunan *test case* dan model menggunakan berbagai teknik *data mining* yang tersedia. Keseluruhan pemilihan teknik ini sangat tergantung pada sifat masalah bisnis dan karakteristik data yang ada. Dalam membangun model, parameter khusus harus ditetapkan untuk memastikan konsistensi dan performa yang optimal.
Evaluation	Pada tahap ini, hasil dari proses pemodelan dianalisis dan dievaluasi untuk memastikan kesesuaian dengan tujuan bisnis yang telah ditetapkan sejak tahap awal proyek. Evaluasi ini bertujuan untuk menilai kinerja model dan keefektifannya dalam mengatasi tantangan yang ditemui. Pemahaman mendalam tentang sejauh mana model dapat memenuhi kebutuhan bisnis menjadi fokus utama dalam langkah ini.

2.3 Tools Penelitian

2.3.1 X

Platform media sosial X menawarkan wadah untuk komunikasi online melalui komputer, membentuk struktur sosial yang terus berkembang. Dengan 1,3 miliar akun dan 336 juta pengguna aktif yang secara kolektif menghasilkan 500 juta tweet setiap harinya, platform ini menjadi salah satu wajah utama interaksi digital era saat ini.

Pengguna X memiliki kebebasan untuk membagikan pemikiran mereka melalui "*tweets*" yang awalnya dibatasi hingga 140 karakter sebelum Oktober 2018, dan kini diperluas menjadi 280 karakter. Tweet-tweet ini, kecuali ditetapkan pribadi, menjadi publik dan dapat menerima respons dari sesame pengguna. Respons tersebut dapat berupa "*retweet*" untuk membagikan tweet, menekan tombol *like*, menyebut nama pengguna lain, atau memberikan tanggapan langsung kepada penulis melalui *mention*.

Selain interaksi pengguna, X menyediakan *Application Programming Interface* (API) yang memungkinkan pengguna untuk mengumpulkan data dengan lebih

mudah. Untuk mengakses API, pengguna harus mengajukan akun pengembang dan setelah mendapatkan persetujuan aplikasi, mereka memperoleh akses ke empat kunci: *consumer key*, *consumer secret*, *access token*, dan *access secret*. Kunci-kunci ini berfungsi sebagai pengenal untuk mengautentikasi pengguna agar dapat mengakses data X termasuk tweet dan informasi pribadi.

API X, sebagai alat paling ampuh dalam hal pengumpulan data, menjadi jendela ke berbagai kategori demografi. Data yang dihasilkan melalui interaksi pengguna X sangat beragam, dan menjadi sumber daya yang kaya bagi peneliti dan pembuat kebijakan. Dengan demikian, X bukan hanya sekedar platform media social, tetapi juga merupakan sumber informasi berharga yang mencerminkan keragaman opini, tren, dan interaksi dalam masyarakat digital [30].

2.3.2 Python

Python adalah sebuah bahasa pemrograman tingkat tinggi yang serbaguna, mudah dibaca, dan memiliki sintaksis yang bersih. Dikembangkan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python dirancang untuk menjadi bahasa yang mudah dipahami dan digunakan. Nama Python berasal dari minat van Rossum pada acara komedi televisi Monty Python. Dalam beberapa tahun terakhir, bahasa pemrograman ini telah memainkan peran yang semakin penting dalam bidang pemrograman. Fenomena ini menunjukkan tren peningkatan kepopuleran Python di kalangan para praktisi *machine learning* dan pengembang perangkat lunak secara umum. Hal ini disebabkan oleh serangkaian keunggulan dan kemudahan yang dimiliki oleh Python dalam konteks pengembangan solusi *machine learning* yang kompleks.

Salah satu keunggulan utama Python adalah ketersediaan seperangkat alat dan perpustakaan yang sangat kuat, seperti NumPy, pandas scikit-learn, dan TensorFlow, yang mempercepat dan menyederhanakan pengembangan model *machine learning*. Alat-alat ini memberikan dukungan luas untuk berbagai tugas, termasuk *preprocessing* data, pemilihan model, evaluasi performa, dan visualisasi hasil. Tidak hanya itu, Python juga telah menjadi pusat pengembangan berbagai jenis jaringan syaraf tiruan yang canggih. Jaringan ini digunakan untuk menangani permasalahan yang sangat beragam, seperti pemrosesan bahasa alami, visi

computer, analisis teks, dan analisis sentiment. Keberhasilan Python dalam menghadirkan teknologi ini membantu menjadikannya bahasa yang sangat diandalkan dalam menyediakan solusi *machine learning* untuk tantangan yang semakin kompleks [31].

2.3.3 Google Colaboratory

Google Colaboratory atau yang sering disebut sebagai Colab, merupakan sebuah proyek yang bertujuan untuk memfasilitasi penyebaran pengetahuan dan penelitian dalam bidang *machine learning*. Platform ini dirancang berdasarkan pada Jupyter dan berperan sebagai dokumen Google yang dapat dibagikan, memungkinkan pengguna untuk berkolaborasi pada buku catatan yang sama. Colaboratory menyediakan *runtime* Python 2 dan 3 yang telah dikonfigurasi sebelumnya dengan Pustaka penting dalam *machine learning* dan *artificial intelligence,* seperti TensorFlow, Matplotlib, dan Keras. Colab memperkenankan pengguna untuk menjalankan kode, mengakses data, dan membuat visualisasi dalam lingkungan *cloud* tanpa perlu menginstal perangkat lunak tambahan secara local. Dengan menggunakan *notebook* Colab, pengguna dapat memanfaatkan infrastruktur *cloud* Google secara langsung

Satu fitur penting Colab adalah bahwa *virtual machine* (VM) yang digunakan akan dinonaktifkan setelah jangka waktu tertentu, sehingga pengguna perlu menyimpan dan mendukung dat amereka secara eksternal, misalnya ke Google Drive. Meskipun VM dinonaktifkan, *notebook* Colab tetap ada dan dapat diakses kembali. Infrastruktur Google Colaboratory dihosting di platform Google Cloud, memberikan keandalan dan aksesbilitas tinggi. Dengan kombinasi fitur-fitur ini, Colab menjadi alat yang efektif untuk *machine learning* dan penelitian di lingkungan *cloud* dengan dukungan perangkat keras yang kuat [32].

2.4 Penelitian Terdahulu

Tabel 2.2 Matriks Penelitian Terdahulu

Jurnal Penelitian	Judul Penelitian	Nama Peneliti	Isi Penelitian
ICECOS 2019 - 3rd International Conference on Electrical Engineering and Computer Science, Proceeding, (2019), 303-308	*Network Centralization Analysis Approach in the Spread of Hoax on Social Media* [33]	Dwi Fitri Brianna, Edi Surya Negara, Yesi Novaria Kunang	Penelitian ini bertujuan untuk menganalisis pola interaksi dan peran penting aktor dalam penyebaran berita hoaks di Twitter. Data di Twitter diolah menggunakan *Social Network Analysis* untuk menentukan aktor yang berperan kunci dalam penyebaran hoaks. *Centrality*, yang melibatkan *degree*, *betweenness*, dan *closeness* dalam jaringan, digunakan untuk mengukur pengaruh aktor-aktor ini. Melalui SNA dan *centrality*, penelitian ini berhasil mengidentifikasi aktor-aktor signifikan dalam penyebaran berita hoaks di platform Twitter.
International Journal of Electronics and Communication Engineering, (2020), 38-42, 14(2)	*Improving Fake News Detection Using K-Means and Support Vector Machine Approaches* [34]	Kasra Majbouri Yazdi, Adell Majbouri Yazdi, Saeid Khodayi, Jingyu Hou, Wanlei Zhou, Saeed Saedy	Penelitian ini mengusulkan pendekatan baru untuk mendeteksi berita hoaks dengan mengintegrasikan pengelompokkan K-Means dan Support Vector Machine (SVM) untuk mengurangi ukuran data dan meningkatkan kinerja klasifikasi. Penelitian ini menggunakan beberapa dataset dengan berbagai fitur untuk mngevaluasi metode yang diusulkan. Hasil evaluasi menunjukkan bahwa metode yang diusulkan berhasil mencapai hasil yang lebih baik dibandingkan dengan metode perbandingan yang menggunakan pendekatan ekstraksi fitur untuk mendeteksi berita hoaks.
(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 9, 2020	*An Empirical Comparison of Fake News Detection using different Machine Learning Algorithms* [35]	Abdulaziz Albahr, Marwan Albahar	Penelitian ini mengevaluasi sejumlah algoritma pembelajaran mesin dalam mendeteksi berita palsu, menyoroti pentingnya deteksi ini dalam mengatasi lonjakan sirkulasi berita palsu terutama dalam konteks peristiwa politik saat ini. Manusia dianggap kurang konsisten dalam mendeteksi berita palsu, sehingga peneliti berusaha mengotomatisasi proses identifikasi dengan menggunakan model pembelajaran mesin untuk menemukan pola bahasa yang membedakan berita nyata dan palsu. Hasil penelitian menunjukkan perbedaan kinerja antara algoritma Naïve Bayes, Random Forest, Neural Network, dan Decision Trees dalam mendeteksi berita palsu. Naïve Bayes misalnya memiliki tingkat akurasi 99%, sementara Decision Trees hanya 76%. Penelitian ini juga menampilkan analisis yang mendetail dari keempat algoritma menggunakan teknik Pemrosesan Bahasa Alami (Natural

			Language Processing/NLP) untuk mendeteksi berita palsu. Selain itu, penelitian membahas ketersediaan data latih berskala besar sebagai fondasi penting dalam deteksi berita palsu dan membandingkan kinerja model dengan dataset seperti ISOT, Kaggle, dan LIAR. Kontribusi utama penelitian ini adalah memberikan analisis mendalam dari berbagai algoritma pembelajaran mesin dalam deteksi berita palsu serta implementasi model untuk mendeteksi dan mengklasifikasikan berita palsu. Diharapkan penelitian ini memberikan wawasan penting dalam menangani masalah sirkulasi berita palsu di era digital.
2019 7th International Conference on Smart Computing and Communications, ICSCC 2019	*Detecting Fake News using Machine Learning and Deep Learning Algorithms* [36]	Abdullah-All-Tanvir, Ehesas Mia Mahir, Saima Akhter, Mohammad Rezwanul Huq	Penelitian ini fokus pada pengembangan model otomatis untuk mendeteksi berita palsu di Twitter, menggunakan dataset dari Cody Buntain yang terdiri dari 20.360 data dengan label H (h*arassment*) dan N (*non-harassment*). H melambangkan berita palsu, sementara N mewakili berita asli atau fakta. Penelitian menggunakan token input seperti kata-kata, karakter, n-gram, dan TF-IDF Vectors untuk menciptakan tiga jenis matriks representasi: Word Level TF-IDF, N-gram Level TF-IDF, dan Character Level TF-IDF. Hasil penelitian menunjukkan bahwa model yang diusulkan mampu secara efektif mendeteksi berita palsu di Twitter. Beberapa algoritma *Machine Learning* seperti SVM, NB, RNN, LR, dan LSTM diuji, dan SVM menunjukkan kinerja terbaik dalam klasifikasi. Temuan penelitian menegaskan bahwa meskipun menggunakan algoritma dasar dalam AI dan Machine Learning, model ini mampu mengatasi penyebaran berita palsu di platform media sosial. Kesimpulannya, model ini memiliki potensi untuk meningkatkan kepercayaan pengguna media sosial dalam memilih berita yang dapat dipercaya.
Measurement: Sensors, (2022), 100495, 24	*Effective Prediction of Fake News Using Two Machine Learning Algorithms* [37]	M. Sudhakar, K.P. Kaliyamurthie	Penelitian ini bertujuan mendeteksi berita hoaks dalam informasi politik dengan menggunakan algoritma *machine learning* yang lebih efektif. Mereka menguji dua algoritma yaitu Logistic Regression dan Naïve Bayes (NBA), pada lebih dari 44.000 data. Kedua algoritma ini berhasil dalam mengenali berita palsu dengan baik. Hasil penelitian menunjukkan bahwa Logistic Regression memiliki akurasi 98,7080%, sedangkan Naïve Bayes memiliki akurasi 94,8490%. Perbedaan statistik antara

			keduanya adalah 0,013. Uji statistik menunjukkan bahwa Logistic Regression memberikan peforma yang lebih baik dalam mendeteksi berita hoaks dalam informasi politik.
Journal of Information Systems and Informatics, (2023), 5(4), 1221-1239	*Empowering Pregnancy Risk Assessment: A Web-Based Classification Framework with K-Means Clustering Enhanced Models* [38]	Wongso, B., Johan, M., & Fianty, M.	Penelitian ini mengembangkan model prediksi untuk risiko kehamilan dengan menggunakan tiga algoritma klasifikasi utama dan pendekatan klasterisasi K-Means. Model yang dikembangkan terbukti efektif dengan mencapai akurasi sebesar 79,53% dan rata-rata F1-score 0.8. Eksplorasi dan *preprocessing data* menjadi tahap kunci dalam meningkatkan akurasi model prediksi.

Pada Tabel 2.2, terdapat beberapa pendekatan yang berfokus pada deteksi hoaks di media sosial menggunakan teknik-teknik seperti SNA, klasterisasi K-Means, algoritma SVM, dan berbagai model prediktif seperti Naïve Bayes dan Logistic Regression. Penelitian ini akan melanjutkan pendekatan menggunakan analisis pola interaksi dari SNA untuk mengidentifikasi peran aktor dalam penyebaran hoaks di jaringan X. Penelitian ini akan langsung membandingkan kinerja dua algoritma, yaitu Naïve Bayes dan Logistic Regression dalammendeteksi berita palsu. Hal ini akan memberikan pemahaman yang lebih mendalam tentang perbandingan langsung antara kedua algoritma dalam konteks jaringan media sosial tertentu.

Dari hasil penelitian terdahulu, terlihat bahwa dua algoritma, Naïve Bayes dan Logistic Regression, telah menonjol dalam performa deteksi berita palsu dalam konteks yang berbeda. Penelitian oleh M. Sudhakar dan K.P. Kaliyamurthie menunjukkan perbandingan kinerja yang jelas antara Naïve Bayes dan Logistic Regression dalam mendeteksi berita palsu dalam informasi politik. Data empiris dari penelitian terdahulu menunjukkan bahwa kedua algoritma ini menonjol dalam deteksi hoaks, dengan Logistic Regression memperlihatkan performa yang lebih unggul dalam beberapa kasus. Dengan demikian, pendekatan penelitian ini bertujuan untuk mendalami perbandingan langsung

antara Naïve Bayes dan Logistic Regression berdasarkan temuan yang sudah ada. Hal ini bisa memberikan wawasan yang lebih terperinci tentang keunggulan dan kelemahan masing-masing algoritma dalam konteks spesifik deteksi hoaks di jaringan media sosial, tanpa melibatkan algoritma lainnya yang mungkin memiliki performa yang lebih rendah dalam penelitian sebelumnya.