

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian sebelumnya yang dijadikan acuan dalam studi ini disusun dan disajikan dalam bentuk tabel untuk memudahkan pemahaman. Tabel ini merangkum berbagai aspek penting dari studi terdahulu, seperti latar belakang penelitian, metode yang digunakan, serta hasil utama yang relevan dengan topik penelitian. Penyajian ini bertujuan untuk menunjukkan posisi dan kontribusi penelitian saat ini dalam memperluas kajian yang telah ada sebelumnya.

Tabel 2. 1 Penelitian Terdahulu

No	Ref	Latar Belakang	Metode	Hasil dan Kesimpulan
1.	[11]	Depresi merupakan masalah kesehatan di seluruh dunia yang memerlukan strategi diagnosis dini yang lebih efisien. Meskipun ada banyak data yang tersedia untuk analisis di <i>platform</i> media sosial seperti Twitter, hubungan antara konten yang menyedihkan dan metrik keterlibatan seperti <i>like</i> dan <i>share</i> terkadang diabaikan. Untuk meningkatkan akurasi identifikasi, penelitian ini menggunakan algoritma pembelajaran mesin untuk mengklasifikasikan informasi yang menandakan depresi dan memadukan teknik pemrosesan bahasa alami dengan metode <i>Grey Relational Grade</i> (GRG).	<i>Logistic Regression</i> , SVM, <i>Random Forest</i> , XGBoost	Penelitian ini menggunakan <i>machine learning</i> untuk mendeteksi depresi berdasarkan 1.6 juta <i>tweet</i> yang diproses dengan NLP. Dalam klasifikasi sentimen yang dilakukan, menggunakan empat model utama yang digunakan yaitu <i>Logistic Regression</i> (Akurasi 96%, <i>Recall</i> tertinggi 0.962), <i>Support Vector Machine</i> (SVM) (Akurasi 95%, <i>Precision</i> tertinggi 0.963), <i>Random Forest Classifier</i> (Akurasi 97%), dan <i>XGBoost Classifier</i> (Akurasi 96%, <i>F1-score</i> tertinggi 0.966). Dalam hal ini, XGBoost menunjukkan performa terbaik dalam menyeimbangkan <i>precision</i> dan <i>recall</i> , sementara <i>Logistic Regression</i> unggul dalam akurasi dan <i>recall</i> . <i>Grey Relational Grade</i> (GRG) digunakan untuk menganalisis hubungan antara interaksi media sosial dan stres, dengan nilai 0.9736 untuk dataset penuh dan 0.9436 untuk subset kecil, hal ini menunjukkan korelasi yang kuat. Kombinasi dari GRG dan XGBoost memberikan wawasan lebih dalam terhadap deteksi kesehatan mental berbasis media sosial [11].

No	Ref	Latar Belakang	Metode	Hasil dan Kesimpulan
2.	[17]	Penelitian ini dilatarbelakangi oleh maraknya penggunaan TikTok sebagai sumber informasi kesehatan mental, terutama di kalangan mahasiswa. Tingginya kepercayaan terhadap konten video yang tersebar di platform tersebut menimbulkan kekhawatiran akan kecenderungan pengguna melakukan <i>self-diagnose</i> tanpa dukungan profesional.	-	Penelitian ini menunjukkan adanya hubungan positif antara persepsi keandalan informasi dalam video TikTok dengan kecenderungan pengguna melakukan <i>self-diagnose</i> terhadap gangguan mental. Semakin tinggi tingkat kepercayaan terhadap isi video, semakin besar pula kecenderungan pengguna untuk mengaplikasikan informasi tersebut pada diri mereka sendiri. Temuan ini mengindikasikan bahwa persepsi terhadap kredibilitas konten digital memainkan peran penting dalam membentuk perilaku pengguna, khususnya dalam konteks kesehatan mental. Penelitian ini juga menyoroti risiko yang ditimbulkan oleh informasi yang tidak tervalidasi, serta perlunya pendekatan yang lebih ketat dalam pengawasan konten serupa di media sosial.
3.	[18]	Maraknya konten kesehatan mental seperti depresi dan kecemasan di TikTok, yang merupakan platform populer di kalangan remaja dan dewasa muda, menimbulkan kekhawatiran mengenai potensi penyebaran informasi yang tidak tepat serta kecenderungan pengguna untuk melakukan <i>self-diagnose</i> .	-	Studi ini menemukan bahwa mayoritas video bertagar depresi dan kecemasan di TikTok berasal dari pengalaman pribadi pengguna, bukan dari profesional. Video tersebut memiliki tingkat keterlibatan tinggi dan sering menampilkan gejala emosional yang mudah dikenali, sehingga berisiko mendorong <i>self-diagnose</i> dan <i>overpathologizing</i> . Algoritma TikTok memperkuat paparan terhadap konten serupa, terutama di kalangan perempuan yang lebih dominan sebagai kreator konten. Meskipun platform ini bisa menjadi ruang dukungan, minimnya literasi digital meningkatkan potensi penyebaran informasi keliru tentang gangguan psikologis.
4.	[19]	TikTok menjadi platform yang berpengaruh besar terhadap kesehatan mental remaja. Algoritmanya yang mudah menyebarkan	-	Hasil menunjukkan bahwa TikTok berperan dalam memperkuat fenomena <i>self-diagnose</i> di kalangan remaja, dengan konten-konten populer

No	Ref	Latar Belakang	Metode	Hasil dan Kesimpulan
		informasi tidak akurat mendorong praktik self-diagnose dan memperkuat stereotip gangguan mental, serta mengurangi kepercayaan terhadap profesional dan penderita yang sesungguhnya.		yang banyak berasal dari non-profesional dan memiliki daya tarik tinggi. Paparan terus-menerus terhadap video terkait depresi dan kecemasan menyebabkan sebagian pengguna mengidentifikasi diri mereka dengan gangguan tersebut, meski hanya berdasarkan gejala ringan. Algoritma TikTok juga memicu efek gema (echo chamber), memperkuat persepsi keliru dan membuat gangguan mental tampak lebih umum atau bahkan “menarik.” Meski disarankan adanya perbaikan sistem filter dan promosi konten profesional, penelitian menekankan pentingnya riset lanjutan untuk memahami dampak jangka panjang fenomena ini.
5.	[23]	Pandemi COVID-19 berdampak signifikan pada sektor penerbangan, yang menyebabkan banyak pelancong menggunakan <i>platform</i> media sosial seperti Twitter untuk menyuarkan keluhan mereka dengan nada mengejek. Proyek ini bertujuan untuk membuat kerangka kerja guna mengidentifikasi dan mengkategorikan sikap sarkastik dan non-sarkastik secara tepat guna membantu sektor penerbangan dalam memahami keluhan konsumen dengan lebih baik, karena sarkasme sering kali mengaburkan makna sebenarnya dari sentimen pengguna.	SVM, GRU	Penelitian ini menerapkan SVM dengan TF-IDF dan GRU untuk menganalisis opini Twitter terkait dengan dampak dari COVID-19 pada industri penerbangan. SVM dengan TF-IDF standar mencapai nilai <i>precision</i> 0.89, <i>train accuracy</i> 0.97, dan <i>test accuracy</i> 0.88, lalu setelah optimasi, <i>precision</i> meningkat ke 0.90 (negatif) dan 0.87 (positif), <i>train accuracy</i> 0.99, dan <i>test accuracy</i> 0.89. GRU mencapai akurasi 84.94% dengan <i>loss score</i> 0.342. Dalam hal ini, SVM menunjukkan performa yang lebih baik [23].
6.	[24]	Kasus <i>cyberbullying</i> semakin sering ditemukan di <i>platform</i> media sosial, terutama Twitter, di mana banyak unggahan yang mengandung ujaran negatif. Penelitian ini dilakukan untuk	SVM dan <i>Information Gain</i> (IG)	Hasil penelitian menunjukkan bahwa metode SVM berhasil mengklasifikasikan <i>tweet</i> yang mengandung <i>cyberbullying</i> dengan performa yang cukup baik, yakni akurasi 80%, <i>precision</i> 75,1%, <i>recall</i> 96%, dan <i>f-measure</i> 85%. Sementara

No	Ref	Latar Belakang	Metode	Hasil dan Kesimpulan
		mendeteksi unggahan yang mengarah pada tindakan <i>cyberbullying</i> dengan memanfaatkan algoritma klasifikasi.		itu, penggunaan <i>Information Gain</i> sebagai teknik seleksi fitur juga menghasilkan kinerja tinggi dengan akurasi 86%, <i>precision</i> 81%, <i>recall</i> 95%, dan <i>f-measure</i> 87%. Kedua metode terbukti efektif dalam mendeteksi konten perundungan di Twitter.
7.	[25]	Diagnosis gangguan depresi umumnya dilakukan melalui wawancara medis dan evaluasi klinis, yang membutuhkan tenaga profesional serta memakan waktu dan biaya. Untuk meningkatkan efisiensi dan akurasi, penelitian ini mengeksplorasi penggunaan data biomarker dari sampel besar masyarakat Belanda, diproses melalui teknik <i>machine learning</i> .	XGBoost, dengan pendekatan penyeimbangan data melalui <i>oversampling</i> , <i>undersampling</i> , dan teknik ROSE	Model XGBoost berhasil mencapai akurasi, <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> yang paling tinggi sebesar 97%. Hasil ini menunjukkan bahwa pendekatan ini sangat efektif dalam mendeteksi depresi berdasarkan data biologis, bahkan ketika proporsi kelas tidak seimbang.

Tabel 2.1 memuat sejumlah studi terdahulu yang mendasari penelitian ini. Sejumlah penelitian telah menunjukkan bahwa media sosial memiliki pengaruh signifikan terhadap persepsi dan perilaku pengguna terkait isu kesehatan mental, khususnya dalam konteks self-diagnose. Salah satu studi menemukan adanya hubungan positif antara persepsi keandalan informasi dalam video TikTok dengan kecenderungan pengguna untuk mengidentifikasi diri mengalami gangguan mental. Pendekatan kuantitatif yang digunakan, seperti kuesioner dan korelasi, mampu memberikan gambaran umum, tetapi belum menjelaskan secara lebih mendalam bagaimana persepsi tersebut terbentuk melalui representasi teks atau narasi dalam konten [17]. Penelitian lain menunjukkan bahwa mayoritas konten bertema depresi dan kecemasan di TikTok berasal dari pengalaman pribadi pengguna, bukan dari tenaga profesional. Video-video ini sering kali menampilkan gejala emosional yang mudah dikenali oleh pengguna, yang dapat memicu proses *overpathologizing* dan dorongan untuk melakukan *self-diagnose*. Terlebih, algoritma TikTok memperkuat eksposur terhadap jenis konten serupa, terutama pada perempuan sebagai kreator dominan. Walaupun pendekatan analisis

konten yang digunakan memberikan wawasan penting, studi ini belum mengeksplorasi respons pengguna di kolom komentar yang juga memiliki potensi besar sebagai indikator persepsi dan interpretasi emosional [18].

Studi lainnya mengungkap bahwa TikTok berperan dalam memperkuat fenomena self-diagnose di kalangan remaja karena mudahnya penyebaran informasi yang tidak tervalidasi serta rendahnya literasi digital pengguna. Konten-konten dengan narasi populer dan daya tarik tinggi yang berasal dari non-profesional berkontribusi terhadap stereotip gangguan mental dan berkurangnya kepercayaan terhadap profesional serta penderita yang sebenarnya. Meskipun menggunakan pendekatan esai analitis, studi ini tidak memberikan bukti empiris yang dapat digunakan untuk mengukur seberapa besar pengaruh konten tersebut secara kuantitatif [19]. Di sisi lain, beberapa studi telah menerapkan algoritma pembelajaran mesin untuk mengklasifikasikan isu kesehatan mental melalui pendekatan berbasis teks di media sosial. Salah satu penelitian memproses 1,6 juta tweet dengan NLP untuk mendeteksi depresi menggunakan empat model utama: Logistic Regression, *Support Vector Machine* (SVM), *Random Forest*, dan XGBoost. Hasilnya menunjukkan bahwa *Random Forest* memiliki akurasi tertinggi (97%), sementara XGBoost unggul dalam *f1-score* (0.966) dan keseimbangan *precision-recall*. Sementara itu, *Logistic Regression* menunjukkan keunggulan pada recall. Penelitian ini juga memadukan metode Grey Relational Grade (GRG) untuk mengukur korelasi antara interaksi digital dengan stres, menghasilkan nilai korelasi yang sangat kuat, yakni 0.9736 pada dataset penuh dan 0.9436 pada subset kecil [11].

Studi lain menggunakan model SVM dan GRU dengan pendekatan TF-IDF untuk menganalisis opini masyarakat terhadap dampak COVID-19 pada industri penerbangan. SVM dengan TF-IDF berhasil mencapai *precision* 0.89, *train accuracy* 0.97, dan *test accuracy* 0.88. Setelah optimasi, nilai *precision* meningkat hingga 0.90 (negatif) dan 0.87 (positif), dengan *test accuracy* mencapai 0.89. GRU mencatat akurasi sebesar 84.94% dengan *loss score* 0.342, namun SVM tetap menunjukkan performa yang lebih optimal [23]. Selain itu, algoritma SVM dan *Information Gain* juga diterapkan untuk mendeteksi ujaran kebencian di Twitter. SVM berhasil mengklasifikasikan tweet bermuatan negatif dengan

akurasi 80%, *precision* 75.1%, *recall* 96%, dan *f-measure* 85%. Penggunaan *Information Gain* menghasilkan performa lebih tinggi, yakni akurasi 86%, *precision* 81%, *recall* 95%, dan *f-measure* 87% [24]. Temuan ini menegaskan bahwa metode ini efektif dalam mengidentifikasi ekspresi emosional, sehingga relevan untuk konteks isu *mental health*. Penggunaan algoritma XGBoost juga terbukti efisien dalam menangani klasifikasi pada data tidak seimbang. Salah satu studi menggunakan data biomarker dari masyarakat Belanda untuk mendeteksi depresi dengan pendekatan *oversampling*, *undersampling*, dan teknik ROSE. Hasilnya menunjukkan bahwa XGBoost mampu mencapai akurasi, *precision*, *recall*, dan *f1-score* sebesar 97% secara keseluruhan, meskipun distribusi data tidak seimbang [25]. Temuan ini menyoroti keunggulan XGBoost dalam menangani tantangan klasifikasi yang kompleks.

Meskipun berbagai pendekatan telah dilakukan, belum banyak studi yang secara khusus menerapkan analisis sentimen terhadap komentar pengguna TikTok yang berkaitan dengan isu *self-diagnose*. Kolom komentar menyimpan ekspresi emosional, validasi sosial, dan persepsi yang tidak dapat ditangkap hanya dari isi video semata. Oleh karena itu, penerapan analisis sentimen dengan pendekatan berbasis teks dapat memberikan gambaran yang lebih utuh mengenai bagaimana pengguna merespons dan menginternalisasi konten kesehatan mental di TikTok. Penggunaan algoritma SVM dan XGBoost dalam konteks ini menjadi relevan karena keduanya telah terbukti efektif dalam mengklasifikasikan teks dengan performa tinggi pada domain sosial dan kesehatan mental. Penelitian ini hadir untuk menjembatani ruang eksplorasi yang belum banyak disentuh sebelumnya, yaitu pemahaman terhadap dinamika persepsi publik yang tercermin dalam komentar terhadap konten kesehatan mental berbasis video pendek.

2.2 Teori tentang Topik Skripsi

2.2.1 Analisis Sentimen

Analisis sentimen, juga dikenal sebagai penambangan opini, adalah metode komputasi yang berupaya memahami pikiran, perasaan, dan emosi individu sebagaimana yang direpresentasikan dalam bentuk teks. Proses ini mengevaluasi isi teks untuk menentukan apakah opini yang

disampaikan bersifat positif atau negatif, baik dalam bentuk dokumen lengkap maupun kalimat tertentu [26]. Analisis sentimen umumnya digunakan untuk memahami serta mengumpulkan pendapat masyarakat yang diungkapkan pada postingan di Blog, Twitter, Facebook, dan *platform* lainnya [27]. Dalam analisis sentimen, terdapat tiga pendekatan utama yang sering digunakan, yaitu *machine learning*, berbasis kamus (*lexicon-based*), dan pendekatan *hybrid*. Pendekatan *machine learning* mengandalkan algoritma seperti SVM atau Naive Bayes untuk membangun model dari data yang telah diberi label. Sementara itu, pendekatan berbasis kamus memanfaatkan daftar kata yang memiliki makna sentimen, seperti kata positif atau negatif, tanpa perlu data berlabel. Adapun pendekatan *hybrid* menggabungkan keduanya untuk memperoleh hasil yang lebih akurat dan stabil, karena mampu menggabungkan kemampuan pembelajaran dari data dengan pengetahuan kata-kata yang bermuatan emosi [28].

2.2.2 Pelabelan Data dengan *Rule-Based*

Pelabelan merupakan tahapan dalam pengolahan data yang bertujuan untuk menetapkan kategori sentimen pada setiap data yang ada, seperti apakah sebuah teks termasuk ke dalam sentimen positif, negatif, atau netral [29]. *Rule-Based Classifier* adalah pendekatan klasifikasi yang menggunakan seperangkat aturan yang disesuaikan dengan kebutuhan dan konteks penelitian. Dalam penerapannya, *rule-based classifier* menggunakan logika *if-then* untuk menentukan kelas suatu data, dan akan mengelompokkan teks ke dalam kategori seperti positif atau negatif sesuai dengan aturan yang telah dirancang sebelumnya [30].

2.2.3 *Text Mining*

Text Mining digunakan dalam menggali dan memperoleh informasi yang diperlukan dengan menerapkan teknik-teknik seperti *machine learning*, *data mining*, *natural language processing*, manajemen pengetahuan, dan pencarian informasi. Metode ini meliputi tahapan seperti

klasifikasi teks, pengambilan informasi, dan ekstraksi kata-kata dari dokumen [31]. *Text Mining* juga merupakan suatu proses untuk mengidentifikasi pengetahuan baru dengan mengekstrak informasi secara otomatis dari sejumlah besar teks tidak terstruktur. Teknik ini memungkinkan pemrosesan dan analisis data dalam jumlah yang besar untuk memperoleh informasi relevan dari sekumpulan teks. Beberapa penerapan *text mining* meliputi analisis sentimen pada media sosial, ekstraksi informasi untuk pengkategorian teks, serta peringkasan teks dengan menyederhanakan teks panjang menjadi satu atau dua paragraf yang lebih ringkas [32]. Tahap awal pada *text mining* memerlukan proses *preprocessing* data, yang melibatkan berbagai macam metode untuk membersihkan serta menstandarkan teks sebelum diolah lebih lanjut. Berbagai metode ini meliputi [33]:

1. *Cleansing*

Tahap *cleansing* dilakukan untuk membersihkan data teks dari elemen-elemen yang tidak memiliki makna penting dalam analisis. Proses ini mencakup penghapusan ulang cuitan (*retweet*), nama pengguna akun yang diawali dengan simbol "@", karakter HTML, emotikon, *hashtag*, tautan URL, serta alamat situs web. Pembersihan ini bertujuan untuk meningkatkan kualitas data agar proses analisis dapat berjalan lebih akurat dan fokus pada isi utama dari komentar atau teks yang dianalisis.

2. *Case Folding*

Tahap *case folding* bertujuan untuk melakukan standarisasi teks dengan mengubah seluruh karakter huruf menjadi bentuk huruf kecil. Langkah ini penting agar perbedaan kapitalisasi tidak memengaruhi hasil analisis, karena kata yang sama dengan huruf besar dan kecil akan dianggap berbeda jika tidak distandarkan. Selain itu, proses ini juga menyaring elemen non-huruf seperti angka dan tanda baca, yang umumnya tidak memberikan kontribusi bermakna dalam pemrosesan teks berbasis linguistik.

3. *Tokenizing*

Proses *tokenizing* dilakukan untuk memecah kalimat atau teks panjang menjadi unit-unit terkecil yang disebut token, biasanya dalam bentuk kata atau frasa. Tahapan ini penting karena memungkinkan sistem untuk memahami struktur kalimat secara lebih sistematis dalam analisis teks. Dengan memisahkan kata-kata menjadi token, data menjadi lebih mudah diolah untuk tahap selanjutnya seperti analisis frekuensi, klasifikasi, atau pemetaan makna.

4. *Filtering*

Tahap *filtering* bertujuan untuk menyaring kata-kata yang dianggap tidak memiliki kontribusi signifikan terhadap makna atau konteks analisis, seperti kata hubung, kata depan, atau kata umum lainnya yang disebut *stopword*. Kata-kata ini biasanya bersifat sangat umum dan sering muncul, tetapi tidak memberikan nilai informatif dalam analisis sentimen. Oleh karena itu, proses ini dilakukan dengan mengacu pada daftar *stopword* tertentu agar teks menjadi lebih bersih dan fokus pada kata-kata yang bermakna secara analitis.

5. *Stemming*

Stemming merupakan proses penting dalam pra-pemrosesan teks yang bertujuan untuk mengubah kata berimbuhan menjadi bentuk dasarnya. Teknik ini dilakukan agar variasi morfologis dari suatu kata, seperti prefiks atau sufiks, tidak dianggap sebagai entitas yang berbeda dalam analisis. Dengan mengembalikan kata ke bentuk dasarnya, sistem dapat menyederhanakan representasi data dan memastikan bahwa makna utama tetap terjaga untuk proses klasifikasi maupun analisis semantik selanjutnya.

2.2.4 Transformasi Data

Transformasi data adalah tahap penting dalam proses *data mining*, pada tahap ini data mentah diubah atau disesuaikan ke dalam format yang sesuai dengan kebutuhan algoritma analisis yang digunakan. Hal ini dilakukan karena beberapa teknik pada *data mining* memerlukan struktur atau bentuk tertentu agar dapat diproses secara efektif dan menghasilkan output yang akurat [34]. Data yang telah melalui proses transformasi selanjutnya dimanfaatkan dalam tahap analisis untuk memperoleh pengetahuan baru. Proses ini dapat dilakukan dengan berbagai macam pendekatan, seperti analisis statistik, algoritma *machine learning*, serta teknik visualisasi informasi yang dapat membantu memahami pola dan hubungan dalam data [35].

2.2.5 Aplikasi TikTok

TikTok merupakan *platform* media sosial dan video musik asal Tiongkok yang diluncurkan pada September 2016. Aplikasi ini menyediakan layanan video pendek yang dapat diakses oleh siapa saja melalui ponsel. Popularitas TikTok meningkat pesat, menjadikannya salah satu aplikasi dengan jumlah unduhan terbanyak. TikTok memiliki misi untuk mendorong kreativitas dan menghadirkan kesenangan melalui video pendek yang tetap menarik, serta tidak membosankan. *Platform* ini menyediakan berbagai kategori konten, seperti humor, edukasi, *fashion*, kecantikan, kuliner, dan *vlog*, sehingga dapat menarik perhatian banyak pengguna [36].

2.2.6 Kesehatan Mental

WHO menjelaskan bahwa kesehatan mental adalah keadaan kesejahteraan di mana seseorang mampu menangani tekanan hidup sehari-hari, menyadari potensi dirinya, bekerja secara efektif, dan memberikan kontribusi positif bagi masyarakat [37]. Menurut data WHO, satu dari empat orang di seluruh dunia mungkin mengalami penyakit mental atau neurologis di beberapa titik dalam hidup mereka. Penyakit mental

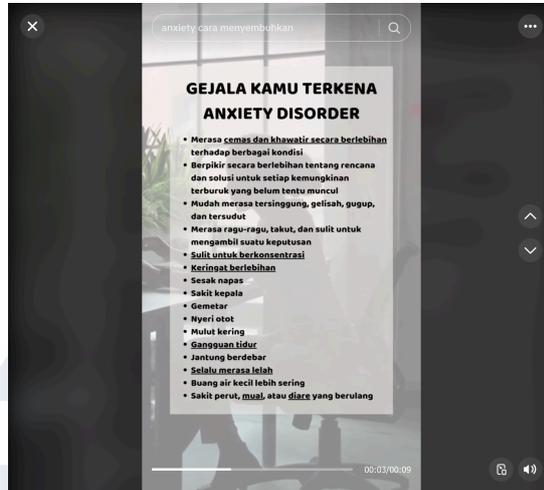
merupakan salah satu penyebab utama masalah kesehatan dan kecacatan di seluruh dunia, yang saat ini memengaruhi sekitar 450 juta orang [38]. Meningkatnya masalah kesehatan mental diperburuk oleh stigma masyarakat, kurangnya kesadaran tentang pentingnya kesehatan mental, dan terbatasnya akses terhadap perawatan dan sumber daya kesehatan mental, terutama di negara-negara berkembang. Dalam hal ini, media sosial merupakan alat yang berguna untuk mengurangi stigma terkait dengan masalah kesehatan mental, meningkatkan kesadaran publik, dan menyebarkan informasi tentang topik tersebut [39].

2.2.6.1 Konten Kesehatan Mental

Menurut Kamus Besar Bahasa Indonesia (KBBI), konten diartikan sebagai informasi yang dibuat dan ditampilkan pada suatu halaman web. Istilah "konten" dalam bahasa Inggris mengacu pada informasi yang disajikan melalui perangkat elektronik atau media lainnya. Istilah "konten media" mengacu pada berbagai sumber daya yang dibuat secara daring oleh pengguna *platform* atau layanan digital, termasuk blog, wiki, forum, gambar digital, video, berkas audio, dan iklan. Oleh karena itu, istilah "konten" merujuk pada berbagai jenis dan format informasi yang memberikan nilai tambah dalam suatu media [32]. Konten dirancang semenarik mungkin dengan menggunakan teks sebagai *caption*, disertai elemen visual seperti gambar atau video [40].

Untuk itu, konten kesehatan mental adalah suatu konten yang memiliki topik mengenai kesehatan mental. Konten kesehatan mental biasanya disajikan dengan cara menarik, singkat, dan mudah dimengerti, sehingga dapat menarik minat banyak orang, khususnya pada kalangan muda. Berbagai *platform* seperti TikTok, Instagram, dan YouTube digunakan untuk menyebarkan informasi mengenai kesehatan mental. Para pembuat konten, baik memiliki latar belakang profesional maupun non-profesional, berbagi materi meliputi

pemahaman mengenai gejala gangguan mental, pengalaman pribadi, serta tips untuk merawat diri (*self-care*) [34].



Gambar 2. 1 Contoh Konten Kesehatan Mental

Sumber: @berbagirasa9 [41]

Gambar 2.1 merupakan salah satu konten kesehatan mental yang diunggah oleh pengguna TikTok @berbagirasa9. Konten tersebut membahas berbagai gejala yang mungkin dialami seseorang dengan gangguan kecemasan (*anxiety disorder*), seperti rasa khawatir yang berlebihan, kesulitan berkonsentrasi, serta keluhan fisik seperti sesak napas, sakit kepala, atau nyeri otot. Melalui penyajian informasi semacam ini, pengguna berupaya meningkatkan kesadaran terhadap gangguan kecemasan dengan menyederhanakan tanda-tanda yang bisa dikenali oleh khalayak umum [41].

2.2.7 *Self-Diagnose*

Istilah "*self*" yang berarti diri sendiri dan "*diagnose*" yang berarti kemampuan mengidentifikasi penyakit, digabungkan untuk membentuk istilah "*self-diagnose*". Istilah ini menggambarkan proses di mana seorang individu menggunakan pengetahuan pribadi yang diperoleh dari sumber tidak resmi, seperti teman, keluarga, internet, atau pengalaman sebelumnya, untuk menyimpulkan bahwa mereka memiliki penyakit atau masalah mental. Sebelum terdapat internet, seseorang sering melakukan

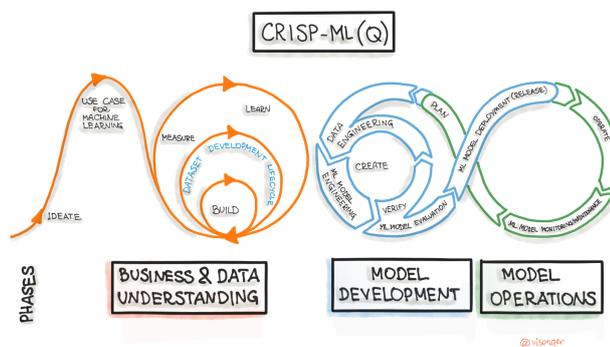
diagnosa secara mandiri dengan mencocokkan gejala yang dialaminya dengan pengalaman orang lain [42].

Kecenderungan melakukan *self-diagnose* dapat memicu perasaan cemas, takut, dan stres berlebih [37]. Melakukan *self-diagnose* dapat membuat seseorang melakukan kesalahan dalam mengaitkan gejala dialami terkait penyakit tertentu, karena penyakit yang dialami mungkin berbeda dengan hasil setelah didiagnosis. Selain itu, memungkinkan seseorang untuk enggan berkonsultasi dengan profesional medis karena perasaan cemas dan takut timbul sebelumnya. Akibatnya, hal ini dapat berdampak buruk pada kesehatan mental karena menumbuhkan rasa khawatir atas berbagai hal yang belum tentu terjadi [43].

2.3 Teori tentang *Framework/Algoritma* yang digunakan

2.3.1 *Cross Industry Standard Process for Machine Learning (CRISP-ML)*

CRISP-ML merupakan suatu kerangka metodologi yang dirancang untuk membangun solusi machine learning dengan memperhatikan tingkat interpretabilitas yang sesuai dengan kebutuhan bisnis [44]. Pendekatan ini tidak hanya fokus pada aspek teknis dalam pengembangan model, tetapi juga mempertimbangkan bagaimana hasil dari model tersebut dapat dipahami dan dimanfaatkan oleh pemangku kepentingan non-teknis. Dengan demikian, CRISP-ML menjadi panduan penting dalam memastikan bahwa implementasi machine learning berjalan efektif sekaligus dapat dipertanggungjawabkan secara praktis dalam konteks dunia nyata [44].



Gambar 2. 2 Tahapan Pengembangan dalam Metode CRISP-ML [45]

Gambar 2.2 merupakan gambaran dari berbagai tahapan yang terdapat dalam metode CRISP-ML. Secara keseluruhan, model proses CRISP-ML(Q) mencakup enam tahapan utama, yaitu: pemahaman bisnis dan data, rekayasa data (*data preparation*), pengembangan model *machine learning*, jaminan kualitas untuk aplikasi ML, proses *deployment*, serta pemantauan dan pemeliharaan. Pada setiap tahapan dalam proses ini, pendekatan kualitas dalam CRISP-ML(Q) menekankan pentingnya penetapan persyaratan dan batasan, seperti standar performa, kualitas data, dan ketahanan model. Selain itu, setiap fase juga mencakup penugasan tugas-tugas spesifik seperti pemilihan algoritma, pelatihan model, serta identifikasi risiko yang berpotensi mengganggu efektivitas dan keberhasilan sistem ML, seperti bias, *overfitting*, atau masalah replikasi. Untuk meminimalkan risiko-risiko tersebut, digunakan metode jaminan kualitas seperti validasi silang (*cross-validation*), dokumentasi proses, dan pencatatan hasil secara sistematis. Berikut ini merupakan berbagai tahapan dalam CRISP-ML, yaitu [46]:

1) *Business and Data Understanding*

Tahapan ini bertujuan untuk memahami secara menyeluruh permasalahan bisnis yang ingin diselesaikan serta mengevaluasi kelayakan data yang tersedia. Tujuan bisnis yang telah diidentifikasi akan diterjemahkan ke dalam bentuk tujuan analitik yang relevan, termasuk menentukan jenis output yang diharapkan dari proses *machine learning*. Di samping itu, dilakukan peninjauan terhadap aspek teknis, hukum, dan etika untuk memastikan proyek dapat dijalankan secara bertanggung jawab.

2) *Data Preparation*

Data yang telah dikumpulkan selanjutnya dipersiapkan agar layak digunakan dalam proses pelatihan model. Tahapan ini mencakup pembersihan data dari elemen tidak relevan atau bermasalah, penanganan nilai kosong (*missing values*), serta proses normalisasi dan seleksi fitur. Selain itu, dapat dilakukan rekayasa fitur baru

maupun augmentasi data agar model dapat mengenali pola secara lebih optimal.

3) *Modeling*

Pada tahap ini, dilakukan pemilihan algoritma *machine learning* yang sesuai dengan karakteristik data dan tujuan bisnis. Proses pelatihan model (*training*) akan menyesuaikan parameter dan struktur model berdasarkan data yang telah disiapkan sebelumnya. Model yang dibangun kemudian diuji pada data validasi untuk mengetahui kemampuan awalnya dalam melakukan prediksi.

4) *Evaluation*

Evaluasi dilakukan untuk mengukur sejauh mana performa model memenuhi kriteria yang telah ditentukan sebelumnya, seperti akurasi, *precision*, *recall*, dan *f1-score*. Jika hasilnya belum sesuai, model dapat dioptimasi atau dilakukan pemilihan ulang metode yang lebih tepat. Proses evaluasi ini menjadi penentu apakah model layak untuk diterapkan ke tahap implementasi atau perlu dilakukan perbaikan lanjutan..

5) *Deployment*

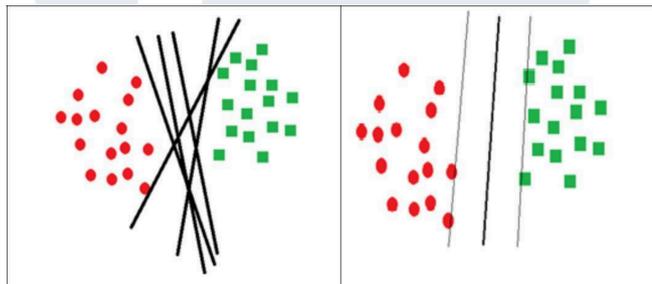
Model yang telah lolos evaluasi akan diterapkan ke dalam sistem produksi agar dapat digunakan oleh pengguna akhir. Tahap ini melibatkan integrasi dengan perangkat lunak atau perangkat keras tertentu serta pengujian fungsionalitas untuk memastikan hasil prediksi berjalan sesuai ekspektasi. Proses *deployment* juga mempertimbangkan aspek skalabilitas dan keamanan data dalam lingkungan operasional.

6) *Monitoring and Maintenance*

Setelah model diimplementasikan, performanya harus dipantau secara berkala untuk mengidentifikasi potensi penurunan akibat perubahan data atau konteks bisnis. Jika ditemukan penurunan performa, maka perlu dilakukan penyesuaian atau pelatihan ulang model. Tahap ini penting untuk memastikan bahwa solusi *machine learning* tetap relevan dan akurat dalam jangka panjang.

2.3.2 Algoritma *Support Vector Machine* (SVM)

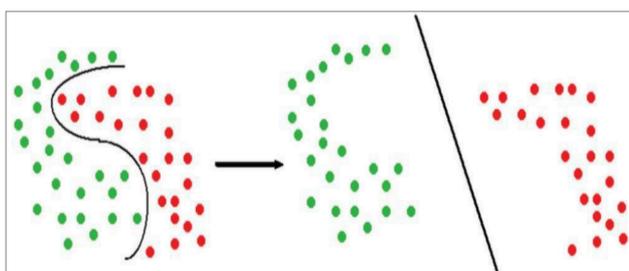
Support Vector Machine (SVM) merupakan metode klasifikasi dalam *machine learning* yang dikembangkan oleh Vladimir Vapnik dengan berbasis *supervised learning*. Teknik ini bekerja dengan memanfaatkan pola dari data pelatihan untuk memprediksi kelas. Proses klasifikasi menggunakan garis pemisah (*hyperlane*) untuk membedakan penilaian positif dan negatif. Garis yang terjauh dari titik data pelatihan terdekat untuk setiap kelas merupakan garis pemisah yang baik karena margin yang lebih besar akan mengurangi kesalahan generalisasi pengklasifikasi [47]. Sebagai gambaran terkait dengan proses pencarian *hyperlane* atau garis pemisah yang terkadang belum mencapai posisi optimal. Gambar 2.1 dan 2.2 mengilustrasikan area di mana penempatan garis pemisah tidak ideal [48].



Gambar 2. 3 Proses Menemukan Hyperplane yang Optimal

Sumber: [48]

Gambar 2.3 merupakan ilustrasi gambaran dua keadaan dalam proses kategorisasi menggunakan *hyperplane*. Di sebelah kiri, banyak *hyperplane* yang dicoba untuk memisahkan kedua kelompok data (titik merah dan kotak hijau), namun tidak ada yang efektif, sehingga menimbulkan masalah kategorisasi. Di sebelah kanan, *hyperplane* optimal ditampilkan sebagai garis lurus dengan memisahkan dua grup dengan margin terbesar, menggaris bawahi gagasan *Support Vector Machines* (SVM) yang berfokus pada pemisahan paling efisien agar model dapat memprediksi dengan baik pada data baru [48].



Gambar 2. 4 Perubahan Kasus Non-linear Menjadi Linear

Sumber: [48]

Gambar 2.4 menggambarkan penggunaan pendekatan pemetaan fitur oleh algoritma *Support Vector Machine* (SVM) untuk mengubah data *non-linear* menjadi data *linier*. Batas *non-linear* kompleks memisahkan dua kelas data yang ditunjukkan dengan titik hijau dan merah di sisi kiri. Prosedur pemetaan fitur pada SVM digunakan untuk memodifikasi ruang data, sehingga kelas dapat dibagi dengan *hyperplane linier*, seperti terlihat di sebelah kanan. Setelah transformasi, kelas hijau dan merah pada gambar dipisahkan dengan jelas oleh garis lurus, sehingga memungkinkan penerapan teknik pembelajaran mesin lebih efisien, serta akurat [48].

Dalam konteks klasifikasi, suatu dataset dapat direpresentasikan sebagai sekumpulan pasangan data, di mana setiap data terdiri dari fitur x_1 dan label kelas y_1 . Fitur tersebut berupa vektor berdimensi n , sedangkan label y_1 mengindikasikan kelas dari data, yang pada klasifikasi biner umumnya bernilai 1 (positif) atau -1 (negatif). Tujuan utama dari metode klasifikasi biner adalah memetakan setiap vektor fitur ke dalam salah satu kelas tersebut [49].

Secara fungsional, prediksi klasifikasi dilakukan melalui sebuah fungsi keputusan, yang dituliskan dalam bentuk $h(x) = \text{sign}(p(x))$, dengan $p(x)$ berfungsi sebagai pemisah antara dua kelas. SVM bekerja dengan mencari sebuah *hyperplane* terbaik yang mampu memisahkan data dari dua kelas tersebut dengan margin atau jarak pemisah terbesar. Jika data tidak dapat dipisahkan secara linear, maka SVM akan memetakan data ke ruang fitur berdimensi lebih tinggi menggunakan transformasi khusus.

Dalam ruang tersebut, pemisahan antar kelas dilakukan dengan mencari bidang pemisah dalam bentuk [49]:

$$w^T \phi(x_1) + b = 0$$

Rumus 2. 1 Persamaan Hyperplane pada Algoritma SVM [49]

w adalah vektor bobot yang tegak lurus terhadap bidang pemisah, dan b merupakan bias. Namun, kenyataannya data sering kali tidak benar-benar dapat dipisahkan secara sempurna meskipun sudah dipetakan ke dimensi yang lebih tinggi, maka digunakan variabel toleransi kesalahan (disebut juga *slack variable*) yang diberi simbol ξ_1 . Nilai $\xi_1 \geq 0$ menunjukkan bahwa terdapat pelanggaran terhadap margin, atau bahwa data tersebut diklasifikasikan secara keliru oleh model. SVM jenis ini disebut sebagai *maximum margin classifier*, karena algoritma berupaya memaksimalkan jarak pemisah sambil meminimalkan kesalahan klasifikasi. Proses optimasi bidang pemisah dilakukan melalui pemrograman kuadrat dengan model sebagai berikut [49]:

$$\begin{aligned} & \text{minimalkan } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \\ & \text{dengan syarat } y_1(w^T \phi(x_i) + b) \geq 1 - \xi_1 \\ & \text{dan } \xi_1 \geq 0 \end{aligned}$$

Rumus 2. 2 Fungsi Optimasi pada Algoritma SVM dengan Soft Margin [49]

Parameter C berperan sebagai pengatur keseimbangan antara dua tujuan, yakni mempertahankan margin yang besar dan mengurangi jumlah total kesalahan. Nilai ini disebut juga sebagai parameter regulasi. Untuk mengatasi data yang tidak dapat dipisahkan secara linear, SVM mengandalkan pendekatan yang disebut kernel *trick*. Pendekatan ini memungkinkan data diproyeksikan ke ruang fitur berdimensi lebih tinggi tanpa perlu melakukan transformasi eksplisit. Kernel sendiri merupakan fungsi yang menghitung tingkat kemiripan antara dua titik data berdasarkan produk dalam (*inner product*) dari vektor-vektor fitur tersebut. Secara umum, fungsi kernel dinotasikan sebagai $K(x_i, x_j) =$

$\langle \phi(x_i), \phi(x_j) \rangle$, dan dapat diimplementasikan dalam beberapa bentuk klasik, antara lain [49]:

a. Kernel Polinomial:

$$K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d$$

Rumus 2. 3 Kernel Polinomial [49]

b. Kernel radial basis function (RBF):

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

Rumus 2. 4 Kernel Polinomial [49]

c. Kernel sigmoid:

$$K(x_i, x_j) = \tanh(-\gamma \langle x_i, x_j \rangle + C)$$

Rumus 2. 5 Kernel Sigmoid [49]

Kernel linear sendiri merupakan bentuk khusus dari kernel polinomial dengan derajat $d = 1$, dan digunakan ketika model cukup bekerja dalam ruang fitur asli. Sementara itu, kernel lainnya digunakan untuk menghitung kemiripan dalam ruang fitur yang telah ditransformasikan. Pemilihan fungsi kernel yang tepat sangat bergantung pada karakteristik data dan jenis permasalahan klasifikasi yang dihadapi. Dengan memilih kernel yang sesuai, performa model SVM dapat ditingkatkan secara signifikan [49].

Berikut ini merupakan pseudocode dari algoritma *Support Vector Machine* (SVM) yang digunakan untuk membangun model klasifikasi berdasarkan data yang tersedia [50]. Pseudocode ini menggambarkan langkah-langkah utama dalam proses pelatihan dan evaluasi model, termasuk pemilihan konfigurasi hyperparameter yang optimal. Penyajiannya bertujuan untuk memberikan pemahaman teknis yang lebih jelas terkait alur kerja algoritma SVM secara sistematis.

Tabel 2. 2 Pseudocode SVM [50]

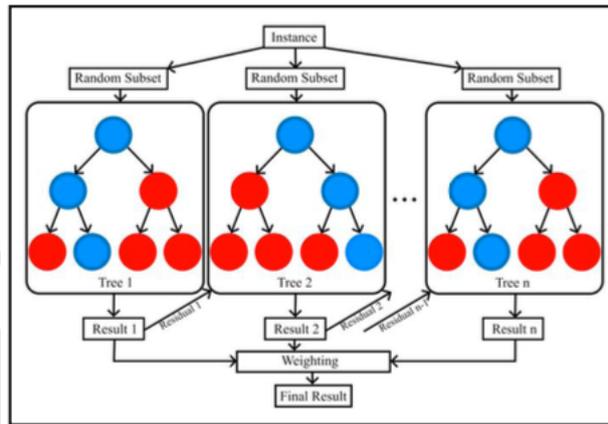
<p>Input: Dataset D, ruang hyperparameter Θ, fungsi target skor $T(\theta)$, batas maksimum n° evaluasi n_{max}.</p> <p>Split randomly the D into N folds; satu untuk data latih dan sisanya untuk data uji.</p> <p>Build model m pada dataset train menggunakan pendekatan SVM.</p> <p>Choose konfigurasi awal $\theta_0 \in \Theta$.</p> <p>Assess skor aslinya $y_0 = T(\theta_0)$.</p> <p>Initialize $S_0 = \{\theta_0, y_0\}$</p> <p>While $t <$ jumlah iterasi maksimum yang dilakukan</p> <p>For $m = 1, \dots, m_{max}$ do</p> <p>Choose pengaturan hiperparameter baru $\theta_0 \in \Theta$ dengan meningkatkan fungsi U_m</p> <p>$\theta_m = \arg\theta_0 \in \Theta \max U_m(\theta, S_t)$,</p> <p>Analyze H in θ_m untuk mendapatkan skor numerik baru $y_m = T(\theta_m)$.</p> <p>Strengthen data $S_m = S_{m-1} \cup \{\theta_m, y_m\}$.</p> <p>Update model pengganti.</p> <p>$m = m + 1$</p> <p>End for</p> <p>End while</p> <p>Extract hiperparameter yang dioptimalkan.</p> <p>Build Model SVM menggunakan hiperparameter yang disetel dari kumpulan data uji.</p> <p>Solve masalah optimasi, mengevaluasi akurasi dan menyimpannya dalam array.</p> <p>Output: Keakuratan rata-rata klasifikasi.</p>

Tabel 2.2 menggambarkan proses umum pelatihan dan optimasi model SVM menggunakan pendekatan berbasis pencarian *hyperparameter*. Proses dimulai dengan membagi dataset secara acak ke dalam beberapa bagian (*fold*), lalu membangun model awal berdasarkan konfigurasi hyperparameter tertentu. Model kemudian dievaluasi untuk mendapatkan skor awal, dan dilakukan iterasi guna mencari pengaturan *hyperparameter* terbaik yang mampu meningkatkan performa model. Dalam setiap iterasi, dilakukan analisis dan perhitungan ulang terhadap skor model, lalu memperbarui data dan model pengganti. Setelah proses iteratif selesai, *hyperparameter* yang telah dioptimalkan digunakan untuk membangun model SVM akhir. Hasil akhir berupa akurasi klasifikasi dari model tersebut disimpan sebagai *output*

2.3.3 Algoritma *Extreme Gradient Boosting* (XGBoost)

Algoritma *Extreme Gradient Boosting* (XGBoost) merupakan pengembangan dari algoritma *gradient tree boosting* dengan menggunakan metode *ensemble* untuk menangani masalah *machine learning* berskala besar dan efisien. XGBoost mampu memecahkan

berbagai masalah klasifikasi, regresi, dan pemeringkatan. XGBoost adalah penghitungan pengumpulan pohon berdasarkan banyak pohon sebelumnya (CART). Komponen keberhasilan XGBoost berasal dari kapasitasnya untuk beradaptasi dengan pengaturan yang berbeda dan meningkatkan perhitungan sebelumnya [51].



Gambar 2. 5 Ilustrasi XGBoost [52]

Gambar 2.5 mengilustrasikan hubungan berurutan antara pohon dalam skema *boosting* dan pengoperasian algoritma XGBoost. XGBoost membangun pohon-pohonnya secara bertahap, berbeda dengan *Random Forest* (RF) yang membentuk pohon secara independen dan paralel. Setiap pohon yang ditambahkan dirancang untuk mempelajari sisa kesalahan (residual) dari prediksi sebelumnya guna meningkatkan akurasi model. Proses ini diulang hingga jumlah pohon yang telah ditentukan tercapai. Bukan dengan menerapkan sistem *voting* sebagaimana pada RF, XGBoost menggabungkan hasil dari seluruh pohon dengan memberikan bobot berdasarkan seberapa besar kontribusi masing-masing pohon terhadap hasil prediksi akhir [52]. Prinsip regularisasi diterapkan dalam XGBoost untuk menurunkan kemungkinan terjadinya *overfitting* selama pelatihan model. Untuk merepresentasikan proses ini secara matematis, digunakan suatu fungsi objektif yang dirumuskan dalam bentuk persamaan berikut ini [53].

$$O = \sum_{i=1}^n L(y_i F(x_i)) + \sum_{k=1}^t R(f_k) + C$$

Rumus 2. 6 Rumus XGBoost [53]

Rumus 2.4 merupakan persamaan rumus XGBoost, yang memiliki penjelasan sebagai berikut [53]:

- a. $L(y_i F(x_i))$ merepresentasikan fungsi kerugian (*loss function*). Fungsi ini berguna untuk menghitung seberapa akurat prediksi model terhadap data aktual, dengan cara membandingkan hasil prediksi dengan nilai sebenarnya.
- b. $R(f_k) + C = \alpha H + \frac{1}{2} n \sum_{j=1}^h \omega_j^2$ merupakan fungsi regulasi yang ditambahkan ke dalam fungsi objektif. Tujuannya adalah untuk menjaga model agar tetap sederhana dan mencegah terjadinya *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri pada data latih.
- c. α merepresentasikan tingkat kompleksitas dari masing-masing daun pada pohon keputusan.
- d. H merepresentasikan total jumlah daun dalam satu pohon keputusan.
- e. n merepresentasikan faktor pengali dalam proses penalti.
- f. ω_j^2 merepresentasikan kuadrat dari nilai output yang dihasilkan pada setiap daun.
- g. C adalah konstanta yang dapat diabaikan dalam proses optimasi.

Berikut ini disajikan pseudocode dari algoritma XGBoost yang digunakan dalam penelitian [54]. Pseudocode ini menggambarkan langkah-langkah utama dalam proses klasifikasi, mulai dari pemisahan data hingga evaluasi akurasi model. Penulisan pseudocode bertujuan untuk memberikan representasi terstruktur dari logika kerja algoritma tanpa terikat pada sintaks bahasa pemrograman tertentu, sehingga memudahkan pemahaman terhadap alur proses yang dilakukan oleh model XGBoost.

Tabel 2. 3 Pseudocode XGBoost [54]

<p>fungsi objektif (params): $Dataset = [(X_{train}, Y_{train}), (X_{test}, Y_{test})]$</p> <p>Bangun XGBClassifier dengan menggunakan params. $Params = \{learningRate, MaxDept, NumberEstimator, subsample\}$ and dataset</p> <p>Hitung prediksi dengan mengevaluasi X_{test} pada model</p> <p>Hitung akurasi dari data <i>training</i>.</p> <p>Kembalikan nilai akurasi dari data <i>testing</i>.</p>
--

Tabel 2.3 merupakan pseudocode dari XGBoost. Pseudocode yang ditampilkan menggambarkan langkah-langkah dalam merancang fungsi objektif untuk optimasi *hyperparameter* pada algoritma XGBoost, khususnya dengan pendekatan *Bayesian Optimization*. Dalam proses ini, dataset dibagi menjadi dua bagian, yakni data latih dan data uji, yang masing-masing terdiri dari fitur (X) dan label (y). Selanjutnya, model XGBClassifier dibangun dengan menggunakan parameter-parameter yang ingin diuji, seperti *learning rate*, *max depth*, *number of estimators*, dan *subsample*. Setelah model dilatih dengan data latih, dilakukan evaluasi terhadap data uji guna menghasilkan prediksi. Akurasi dari hasil prediksi ini kemudian dihitung dan dijadikan sebagai nilai keluaran dari fungsi objektif. Nilai akurasi ini menjadi tolok ukur untuk menilai efektivitas kombinasi parameter yang digunakan, sehingga pendekatan ini membantu dalam memilih konfigurasi terbaik yang dapat meningkatkan performa model klasifikasi berbasis XGBoost.

2.3.4 TF-IDF

Setelah melalui tahap pra-pemrosesan, data teks perlu diubah ke dalam bentuk numerik agar dapat diproses oleh algoritma *machine learning*. Salah satu metode yang umum digunakan untuk tujuan ini adalah TF-IDF (*Term Frequency-Inverse Document Frequency*). TF-IDF memberikan bobot pada setiap kata dalam dokumen berdasarkan dua hal, yaitu seberapa sering kata tersebut muncul dalam satu dokumen (*term*

frequency) dan seberapa jarang kata tersebut muncul di seluruh kumpulan dokumen (*inverse document frequency*). Dengan metode ini, kata-kata yang dianggap lebih relevan terhadap konteks dokumen akan memiliki bobot yang lebih tinggi, sehingga membantu model dalam mengenali pola atau informasi penting dari teks [55]. Berikut ini merupakan cara perhitungan *Term Frequency Inverse Document Frequency* (TF-IDF) [56]:

1. *Term Frequency* (TF) merepresentasikan frekuensi kemunculan fitur t dalam dokumen d , yang menunjukkan tingkat kepentingan fitur tersebut dalam konteks dokumen tersebut.

$$TF_t = (t, d)$$

Rumus 2. 7 Rumus *Term Frequency* (TF)[56]

2. Nilai *Inverse Document Frequency* (IDF) dihitung dengan mengambil logaritma dari perbandingan antara jumlah seluruh dokumen n dengan jumlah dokumen df yang mengandung fitur t .

$$IDF_t = \log \frac{n}{df(t)} + 1$$

Rumus 2. 8 Rumus *Inverse Document Frequency* (IDF) [56]

3. Nilai *Term Frequency-Inverse Document Frequency* (W_t) diperoleh dengan mengalikan nilai TF (*Term Frequency*) dengan nilai IDF (*Inverse Document Frequency*) untuk fitur tertentu.

$$W_t = TF_t \times IDF_t$$

Rumus 2. 9 Rumus TF-IDF [56]

2.3.5 *Random Oversampling*

Salah satu teknik paling populer untuk mengatasi masalah ketidakseimbangan dalam data adalah resampling, yang merupakan salah satu dari berbagai strategi untuk meningkatkan hasil prediksi. Terdapat beberapa teknik resampling yaitu, *SMOTE Oversampling*, *Random Oversampling*, *Random Undersampling*, *NearMiss Undersampling*, *SMOTE dengan Tomek Links*, dan *SMOTE dengan Edited Nearest Neighbors* [57]. *Random Oversampling* merupakan metode yang

digunakan untuk mengatasi ketidakseimbangan jumlah data antar kelas dalam proses pelatihan model. Teknik ini dilakukan dengan menambahkan data dari kelas minoritas secara acak ke dalam data pelatihan hingga jumlahnya sebanding dengan kelas mayoritas. Proses ini diawali dengan menghitung selisih jumlah data antara kelas mayoritas dan minoritas. Selanjutnya, data dari kelas minoritas dipilih secara acak dan disalin ke dalam dataset pelatihan secara berulang, sebanyak jumlah selisih yang telah dihitung sebelumnya. Dengan demikian, distribusi kelas menjadi lebih seimbang, sehingga model tidak cenderung bias terhadap kelas yang lebih dominan [58].

2.3.6 Pemisahan Data

Pemisahan data (*data split*) adalah proses yang digunakan untuk membagi dataset menjadi dua bagian, yaitu data pelatihan dan data pengujian. Teknik ini merupakan salah satu faktor penting yang mempengaruhi seberapa baik model klasifikasi bekerja pada algoritma pembelajaran mesin [59]. Data *training* digunakan sebagai dasar pembelajaran bagi model untuk mengenali pola dari data yang sudah diketahui kebenarannya. Data ini membantu model membangun pemahaman terhadap hubungan antar fitur berdasarkan kondisi nyata. Setelah itu, model akan diuji menggunakan data *testing*, yaitu data yang belum pernah dipelajari sebelumnya. Tujuan dari pengujian ini adalah untuk menilai seberapa baik model mampu mengklasifikasikan data secara akurat berdasarkan pembelajaran yang telah dilakukan [60].

Dalam proses klasifikasi menggunakan algoritma *supervised learning*, tingkat akurasi model tidak hanya ditentukan oleh karakteristik data, tetapi juga oleh proporsi pembagian data tersebut. Komposisi yang tidak seimbang atau kurang tepat antara data training dan testing dapat berdampak negatif pada kemampuan model dalam mengenali pola dan menghasilkan prediksi yang akurat. Oleh karena itu, pemilihan proporsi yang sesuai menjadi aspek penting yang perlu diperhatikan [61]. Tujuannya dari pemisahan data adalah agar model mampu mengenali pola

secara umum, tidak hanya pada data yang sudah dikenal, tetapi juga dapat diterapkan pada data baru. Dalam penerapannya, pembagian ini biasanya menggunakan rasio 80% untuk pelatihan dan 20% untuk pengujian, atau dalam beberapa kasus juga digunakan rasio 70 banding 30 tergantung kebutuhan dan karakteristik dataset [62].

2.3.7 Evaluasi Kinerja Model

Evaluasi merupakan salah satu tahap krusial dalam proses *machine learning*, karena digunakan untuk menilai sejauh mana kinerja model yang dibangun. Proses ini membantu menentukan apakah model sudah cukup baik atau masih memerlukan penyesuaian lebih lanjut. Evaluasi juga memastikan bahwa model tidak hanya menghasilkan hasil yang baik pada data pelatihan, tetapi juga mampu memberikan prediksi yang akurat pada data baru. Pada bagian ini akan dijelaskan elemen-elemen utama dalam *machine learning*, yang meliputi data, algoritma, model, dan evaluasi, serta bagaimana semuanya saling berkaitan untuk menghasilkan solusi yang optimal dalam berbagai penerapan *machine learning* [63]. Evaluasi dilakukan dengan melihat beberapa metrik, di antaranya akurasi dan *F1 Score*. Akurasi menunjukkan seberapa besar proporsi prediksi yang benar, baik pada kelas positif maupun negatif, terhadap keseluruhan data uji. Sementara itu, *F1 Score* digunakan untuk mengevaluasi keseimbangan antara *precision* dan *recall*, sehingga dapat mencerminkan kemampuan model dalam menghasilkan prediksi yang tepat dan relevan. Seluruh nilai evaluasi ini, termasuk akurasi, *precision*, *recall*, dan *F1 Score*, diperoleh dengan memanfaatkan *Confusion Matrix* sebagai dasar perhitungannya [64].

1. *Confusion Matrix*

Confusion matrix merupakan sebuah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi, terutama dalam mengukur tingkat akurasi dan efektivitas model dalam memprediksi data uji. Metode ini menjadi alat bantu penting dalam menilai bagaimana algoritma machine learning mengklasifikasikan

data ke dalam kategori yang benar maupun salah. Di dalam *confusion matrix* terdapat empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Berikut ini merupakan bentuk umum dari *confusion matrix* [65].

Tabel 2. 4 Gambaran Umum *Confusion Matrix* [65]

Kelas Prediksi	Kelas Aktual	
	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	TP	FP
<i>Negative</i>	FN	TN

Tabel 2.2 merupakan gambaran umum dari *confusion matrix*. *True Positive* (TP) menggambarkan situasi ketika model berhasil membuat prediksi yang sesuai dengan kondisi sebenarnya. Sebaliknya, *False Negative* (FN) terjadi ketika model tidak mengenali data yang seharusnya diklasifikasikan sebagai benar. Sementara itu, *False Positive* (FP) merupakan kondisi di mana model memberikan prediksi yang dianggap benar, namun kenyataannya tidak sesuai dengan data aktual [65].

2. Accuracy

Akurasi merupakan rasio yang menunjukkan jumlah prediksi benar dari model, baik pada kelas positif maupun kelas negatif, jika dibandingkan dengan total keseluruhan data yang diuji. Ukuran ini digunakan untuk menilai tingkat ketepatan model klasifikasi yang digunakan. Nilai akurasi dapat dihitung menggunakan rumus berikut [64].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Rumus 2. 10 Rumus Akurasi [64]

3. Precision

Precision merupakan rasio yang menunjukkan proporsi prediksi positif yang benar dibandingkan dengan seluruh prediksi

yang diklasifikasikan sebagai positif oleh model. Metrik ini digunakan untuk menilai seberapa tepat model dalam memprediksi data yang termasuk dalam kelas positif. Nilai *precision* dihitung menggunakan rumus berikut [64].

$$Precision = \frac{TP}{TP + FP}$$

Rumus 2. 11 Rumus Presisi [64]

4. *Recall*

Recall merupakan rasio yang menunjukkan jumlah prediksi positif yang berhasil dikenali dengan benar oleh model, dibandingkan dengan seluruh data yang sebenarnya termasuk dalam kelas positif. Metrik ini digunakan untuk menggambarkan kemampuan model dalam menemukan kembali data yang relevan. Nilai *recall* dihitung menggunakan rumus berikut [64].

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2. 12 Rumus *Recall* [64]

5. *F1 Score*

F1 Score merupakan ukuran yang menggabungkan nilai *precision* dan *recall* ke dalam satu metrik tunggal. Nilai ini digunakan untuk menilai keseimbangan antara ketepatan dan keberhasilan model dalam mengklasifikasikan data. *F1 Score* dihitung menggunakan rumus berikut [64].

$$F1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$$

Rumus 2. 13 Rumus *F1-Score* [64]

2.4 Teori tentang *tools/software* yang digunakan

2.4.1 Python

Python merupakan bahasa pemrograman interpretatif yang sangat fleksibel dengan filosofi desain dengan menekankan kejelasan kode. Python adalah bahasa pemrograman yang memadukan kekuatan, fungsionalitas, dan sintaksis yang intuitif. Python juga dilengkapi dengan

pustaka standar yang besar dan beragam. Manfaat lain dari bahasa ini adalah memiliki komunitas yang cukup besar di belakangnya. Banyak paradigma pemrograman yang didukung oleh Python, dengan penekanan khusus pada pemrograman imperatif, fungsional, dan berorientasi objek. Sebagai bahasa pemrograman dinamis, Python juga menonjolkan fitur manajemen memori otomatis [66].

2.4.2 Anaconda

Anaconda merupakan perangkat lunak yang dirancang untuk mempermudah pengelolaan paket, dependensi, dan lingkungan kerja dalam pengembangan proyek data *science* berbasis Python. Platform ini menyediakan sarana terintegrasi yang memungkinkan pengguna untuk mengelola kebutuhan teknis proyek secara efisien. Melalui Anaconda, pengguna dapat menjalankan berbagai IDE seperti Jupyter Lab, Jupyter Notebook, Spider, dan opsi lainnya [67].

2.4.3 Jupyter Notebook

Jupyter Notebook adalah proyek sumber terbuka yang menggabungkan kode Python yang dapat langsung dieksekusi dengan teks *Markdown*. Melalui integrasi ini, lingkungan interaktif dikenal sebagai *notebook* tercipta, yang memungkinkan pengguna untuk berinteraksi dengan teks atau kode di satu lokasi [68]. Jupyter Notebook juga merupakan sebuah program pengembangan interaktif yang beroperasi dalam lingkungan web dan dapat diakses melalui *browser* bawaan. Dalam Jupyter Notebook, setiap blok kode dapat dieksekusi secara mandiri, menciptakan fleksibilitas yang tinggi, serta memungkinkan pengguna untuk dengan mudah mengeksplorasi dengan kode [69]. Dokumen yang ditulis menggunakan aplikasi Jupyter Notebook disebut Jupyter Notebook, dan memiliki ekstensi *file* .ipynb. Dokumen ini memiliki elemen teks kaya, termasuk tautan, foto, dan kalkulasi matematika, beserta kode komputer. Komponen-komponen ini dapat digabungkan menjadi satu wadah interaktif oleh pengguna [70].