BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berikut merupakan tabel 2.1 Tabel penelitian terdahulu:

Tabel 2. 1 Penelitian Terdahulu

Penelitian 1	
Judul	Analisis Dan Komparasi Algoritma Klasifikasi Dalam Indeks Pencemaran Udara di DKI Jakarta [10]
Jurnal	Jurnal Informatika dan Komputer(JIKO) , Vol.4 , No.2, SINTA 4
Tahun	2021
Penulis	Syekh S A Umri, Muhammad S Firdaus, dan Aji Primajaya
Indikator & Definisi	Indeks Standar Pencemaran Udara merupakan metrik untuk menentukan level kualitas udara. Pada penelitian ini, Indeks Standar Pencemaran Udara level kualitas udara ditentukan berdasarkan lima pencemar utama: karbon monoksida (CO), sulfur dioksida (SO2), nitrogen dioksida (NO2), ozon permukaan (O3), dan partikel debu (PM10).
Metode	SVM, *Decision Tree, Naïve bayes,* KNN, *Neural Network Backpropagation*
Hasil Pembahasan	*Decision Tree* memiliki performa terbaik dengan akurasi 99.80%, kappa 0.996, RMSE 0.039, dan waktu eksekusi 0.8 detik.
Future Research	-
Penelitian 2	
Judul	Klasifikasi Tingkat Pencemaran Udara Pada Sektor Industri Dengan Metode *Random Forest* [11]
Jurnal	*Computer Science Research and Its Development* (CSIRD), Vol.13, No.3A, SINTA 3
Tahun	2021
Penulis	Suci Cahaya Hati Nasution, Fibri Rakhmawati, dan Riri Syafitri Lubis
Indikator & Definisi	Pencemaran Udara : dibagi menjadi tiga kelas yaitu rendah,sedang,dan tinggi. Hal ini memberikan gambaran tentang tingkat pencemaran udara yang disebabkan oleh kegiatan industri. Industri-industri aktif sering kali menghasilkan berbagai jenis gas emisi sebagai produk dari proses produksi mereka, seperti sulfur dioksida (SO2),

	nitrogen dioksida (NO2), hidrokarbon (HC), partikulat terlarut dalam udara (TSP), amonia (NH3), ozon (O3), dan sebagainya.
Metode	*Random Forest*
Hasil Pembahasan	*Random Forest* menunjukkan tingkat akurasi sebesar 95%, menunjukkan prediksi model yang benar, Variabel yang paling berpengaruh adalah kadar sulfur dioksida dan nitrogen dioksida, sedangkan opasitas dan partikulat memiliki pengaruh terendah.
Future Research	-. Implementasi metode klasifikasi lainnya selain *Random Forest.* -. Penggunaan data industri terbaru.
Penelitian 3	
Judul	Penerapan Metode *Extreme Gradient Boosting* (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit [12]
Jurnal	*Journal of Mathematics* (JOMTA), Vol.4, No.1
Tahun	2022
Penulis	Sri Erlina Yulianti, Oni Soesanto, dan Yuana Sukmawaty
Indikator & Definisi	-. Riwayat pembayaran : Riwayat pembayaran mencatat bagaimana nasabah membayar tagihan kartu kreditnya sepanjang waktu. Riwayat baik menunjukkan pembayaran tepat waktu, sementara yang buruk bisa termasuk keterlambatan, pembayaran minimum, atau tidak membayar sama sekali. Riwayat buruk bisa mengindikasikan risiko kredit macet. -. Jumlah tagihan bulanan : Jumlah tagihan bulanan adalah total pembayaran yang harus dilakukan oleh pemegang kartu kredit kepada penyedia setiap bulan. Ini mencakup semua transaksi yang dilakukan dengan kartu kredit selama periode tagihan yang ditetapkan. -. Jumlah kredit yang dimiliki : Jumlah kredit yang dimiliki merujuk pada total kredit yang diberikan kepada seorang individu oleh penerbit kartu kredit.
Metode	XGBoost
Hasil Pembahasan	XGBoost dengan parameter *default* pada dataset nasabah pengguna kartu kredit menghasilkan model yang cukup baik. Akurasi model mencapai 80,02%, presisi 85,32%, dan *recall* 94,86%, sehingga dapat dikategorikan sebagai klasifikasi yang baik.
Future Research	-
Penelitian 4	
Judul	Peningkatan Akurasi *K-Nearest Neighbor* Pada Data Index Standar Pencemaran Udara Kota Pekanbaru [16]
Jurnal	*IT Journal Research and Development* (ITJRD) , Vol.4, No.2, SINTA 3
Tahun	2020

Penulis	Yuliska, dan Khairul Umam Syaliman
Indikator & Definisi	-. Peningkatan Akurasi KNN: Melakukan pendekatan pembobotan atribut (*Attribute Weighting*) dan *local mean.* -. Metode KNN : Metode yang diusulkan dalam penelitian mampu meningkatkan akurasi sebesar 2.42%, dengan rata-rata tingkat akurasi sebesar 97.09%.
Metode	*K-Nearest Neighbor*, pembotoan atribut, dan *local mean*
Hasil Pembahasan	Hasil menunjukkan bahwa algoritma KNN menghasilkan akurasi sebesar 97,09% .
Future Research	-
Penelitian 5	
Judul	Analisis Klasifikasi Indeks Kualitas Udara Kota di Indonesia Menggunakan Metode *K-Nearest Neighbor* dan *Naïve Bayes* [17]
Jurnal	Mahasiswa Teknik Informatika (MANTIK), Vol.8, No.1, SINTA 5
Tahun	2024
Penulis	Ali Maulana, Ade Irma Purnamasari, dan Irfan Ali
Indikator & Definisi	-. Air Quality Index (AQI): Merupakan indikator utama yang digunakan dalam penelitian untuk mengukur kualitas udara. -. Atribut-atribut dalam data World Air Quality Index by City and Coordinates: Termasuk *country, city, AQI value, AQI category*, dan atribut lainnya yang digunakan sebagai indikator dalam analisis kualitas udara
Metode	KNN dan *Naïve Bayes* dengan *K-fold cross validation*.
Hasil Pembahasan	Hasil menunjukkan bahwa algoritma Naïve Bayes menghasilkan akurasi yang lebih tinggi yaitu sebesar 95,97% , sedangkan algoritma KNN sebesar 95,13% . Kedua metode tersebut menunjukkan kinerja yang sangat baik dengan akurasi diatas 95%.
Future Research	Penggunaan Metode Klasifikasi Lain: Selain *K-Nearest Neighbor* dan *Naïve Bayes*, penelitian menyarankan penggunaan metode klasifikasi lain seperti *Decision Trees*, *Support Vector Machine*, atau *Neural Networks* untuk membandingkan kinerja dan hasil klasifikasi.
Penelitian 6	
Judul	Klasifikasi Kualitas Udara Dengan Metode *Support Vector Machine* [18]
Jurnal	Jurnal Informatika & Rekayasa Elektronika (JIRE), Vol.4, No.1, SINTA 4
Tahun	2022
Penulis	Ade Silvia Handayani, Sopian Soim, Theresia Enim agusdi, Rumiasih, dan Ali Nurdin
Indikator & Definisi	-. Parameter kualitas udara : Parameter kualitas udara adalah ukuran atau indikator yang digunakan untuk mengevaluasi

	tingkat polusi atau kebersihan udara di suatu lokasi atau wilayah. Penelitian menggunakan lima parameter zat pencemar udara, yaitu CO (ppm), CO2 (ppm), HC (ppm), PM10 (µg/m3), suhu (°C), dan kelembaban (%). Indikator ini digunakan sebagai data input untuk proses klasifikasi menggunakan algoritma *Support Vector Machine*.
Metode	*Support Vector Machine*
Hasil Pembahasan	Hasil pengujian menunjukkan bahwa akurasi klasifikasi terbaik diperoleh pada sensor kedua, yaitu sebesar 99,33%. Ini menunjukkan bahwa metode *Support Vector Machine* efektif dalam menangani permasalahan klasifikasi dalam kasus ini.
Future Research	Penulis merekomendasikan untuk melakukan klasifikasi menggunakan metode *Support Vector Machine* (SVM) dengan variasi fungsi kernel seperti *Polynomial*, RBF, dan lainnya.
Penelitian 7	
Judul	*Implementation of Support Vector Machine Algorithm for Identifying Facial Skin Types* [19]
Jurnal	*TEST Engineering & Management*, Vol.83, Scopus (Q3)
Tahun	2020
Penulis	Marisa Tri Utami, Julio Christian Young, dan Arya Wicaksana
Indikator & Definisi	Tipe jenis kulit : tipe jenis kulit dikategorikan menjadi normal, kering, dan berminyak.
Metode	*Sport Vector Machine*
Hasil Pembahasan	Hasil menunjukkan bahwa algoritma *Support Vector Machine* berhasil mengidentifikasi jenis kulit wajah. Jumlah parameter *feature extraction* dan *hyperparameter* seperti C dan gamma memengaruhi hasil F-*score*. Nilai C adalah 120.000 dan gamma adalah 1. Hasil penelitian menunjukkan presisi, *recall*, dan akurasi masing-masing sebesar 0,85 dan F-*score* sebesar 0,85.
Future Research	-
Penelitian 8	
Judul	*Hyperparameter Tuning* menggunakan *GridsearchCV* pada *Random Forest* untuk Deteksi *Malware* [20]
Jurnal	*Multinetics*, Vol.9, No.1, SINTA 3
Tahun	2023
Penulis	Iik Muhamad Malik Matin
Indikator & Definisi	Deteksi Malware : proses identifikasi dan pengenalan perangkat lunak berbahaya atau *malicious software* (*malware*) yang dapat merusak atau mengganggu sistem komputer, perangkat mobile, jaringan, atau data pengguna.
Metode	Random Forest dan optimasi *Hyperparameter Tuning* menggunakan *GridsearchCV*

Hasil Pembahasan	Dengan menggunakan algoritma *Random Forest* dan optimasi melalui *hyperparameter tuning*, kinerja model meningkat secara signifikan. Akurasi mencapai 99,23%, presisi mencapai 99,7%, TPR mencapai 99,44% , dan F1-*Score* mencapai 99,26%. *Recall* juga mengalami peningkatan terbesar, sebesar 0,37%, sementara akurasi dan F1-*score* juga naik sebesar 0,19%.
Future Research	Penelitian selanjutnya disarankan untuk mencakup eksplorasi nilai-nilai *hyperparameter* yang lebih beragam, variasi algoritma yang lebih luas, dan teknik seleksi fitur yang lebih canggih untuk meningkatkan kinerja model secara signifikan.
Penelitian 9	
Judul	Penerapan Algoritma *K-Nearest Neighbor* dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek [21]
Jurnal	KOMPUTA Jurnal Ilmiah Komputer dan Informatika
Tahun	2023
Penulis	Robi Nurhidayat,, dan Kania Evita Dewi
Temuan Indikator & Definisi	-. Aspek kemasan : mencakup semua elemen yang terkait dengan penampilan fisik dan desain suatu produk. -. Aspek Harga : merujuk pada berbagai faktor yang memengaruhi penetapan harga suatu produk atau layanan. -. Aspek Aroma : merujuk pada karakteristik bau atau aroma suatu produk, yang dapat memengaruhi persepsi konsumen tentang kualitas, kesegaran, dan daya tarik produk tersebut. -. Aspek Efektivitas : merujuk pada kemampuan suatu produk, layanan, atau proses dalam mencapai tujuan atau hasil yang diinginkan dengan efisien dan memuaskan.
Metode	K-*Nearest Neighbor* ,ekstraksi fitur N-Gram, dan *Random Over Sampling*
Hasil Pembahasan	Hasil pengujian menunjukkan bahwa akurasi tertinggi, mencapai 98,6%, diperoleh pada analisis aspek kemasan dalam skenario data 80:20.
Future Research	Penelitian ini dapat ditingkatkan lagi dengan mengembangkan teknik penyeimbangan data yang lebih canggih, yang berpotensi meningkatkan hasil secara signifikan. Dalam penelitian ini, terlihat bahwa keseimbangan data dapat meningkatkan akurasi, dan hal ini bisa dijadikan fokus pengembangan di masa depan untuk mencapai hasil yang lebih baik.
Penelitian 10	
Judul	*Feature Selection Using New Version of V-Shaped Transfer Function for Salp Swarm Algorithm in Sentiment Analysis* [22]
Jurnal	*Computation*, Scopus (Q2)
Tahun	2023
Penulis	Dinar Ajeng Kristiyanti ,Imas Sukaesih Sitanggang , Annisa dan Sri Nurdiati

Temuan Indikator & Definisi	Fitur seleksi menggunakan *New V-TF* berbasis SSA , mengacu pada proses seleksi fitur dalam analisis sentimen yang menggunakan kombinasi dari jenis transfer *function* yang baru dan Algoritma Salp Swarm untuk memilih subset fitur terbaik yang akan digunakan dalam memodelkan sentimen.
Metode	*K-Nearest Neighbor* (KNN), *Support Vector Machine*, *Naïve Bayes*,dan *Algoritma Salp Swarm (SSA)* sebagai *Feature Selection.*
Hasil Pembahasan	Dalam hasil penelitian, terlihat adanya peningkatan sebesar 31,55%, membawa akurasi terbaik menjadi 80,95% untuk model KNN dengan penerapan New V-TF yang berbasis SSA
Future Research	-. Penelitian mendatang disarankan untuk berfokus pada evaluasi kinerja SSA-new V3-TF pada dataset berbahasa Indonesia untuk analisis sentimen serta beragam tugas pembelajaran mesin. -. Penelitian mendatang disarankan dapat mengevaluasi kemungkinan penerapan versi terbaru dari V-TF pada algoritma optimasi lain yang mengalami kendala dalam menangani dimensi (jumlah fitur) dan ukuran data, seperti PSO, GA, ACO, dan ALO, khususnya untuk optimasi fitur dalam analisis sentimen.
Penelitian 11	
Judul	*Heart Disease Prediction System using hybrid model of Multi-layer perception and XGBoost algorithms* [23]
Jurnal	*BIO Web of Conferences*
Tahun	2024
Penulis	Israa Nadheer
Temuan Indikator & Definisi	Faktor risiko infeksi kardiovaskular : kondisi atau kebiasaan yang meningkatkan kemungkinan seseorang untuk mengalami infeksi yang dapat mempengaruhi sistem kardiovaskular. Beberapa faktornya meliputi kebiasaan merokok, usia, riwayat keluarga, pola makan yang buruk, kadar lipid, kurang aktivitas fisik, hipertensi, penambahan berat badan, dan konsumsi alkohol.
Metode	*Multi-layer perception , Neural Network,* dan XGBoost
Hasil Pembahasan	Hasil menunjukkan bahwa menggabungkan model MLP-NN ke dalam XGBoost setelah mengelompokkan fitur adalah strategi yang menjanjikan untuk meningkatkan kemampuan prediktif dalam klasifikasi penyakit jantung. Pendekatan ini berhasil mencapai tingkat akurasi sebesar 96,67%, sensitivitas 95,92%, dan presisi 97,92%, dengan skor F1 mencapai 96,91%.
Future Research	Penelitian selanjutnya disarankan untuk mencakup studi yang lebih mendalam tentang metode teknis dan pemilihan algoritma untuk optimasi kinerja. Selain itu, penelitian selanjutnya disarankan untuk mengeksplorasi potensi pendekatan baru, seperti analisis sistem dan *deep neural*

	network, model klasifikasi berdasarkan sistem inferensi *neuro-fuzzy* adaptif, dan seleksi fitur berbasis optimasi partikel, Hal ini dapat mendeteksi penyakit jantung yang lebih baik.

Berdasarkan referensi pada tabel 2.1 yang berisi mengenai daftar penelitian terdahulu, penelitian ini menghasilkan temuan sebagai berikut :

1. Berdasarkan artikel jurnal yang berjudul "Analisis Dan Komparasi Algoritma Klasifikasi Dalam Indeks Pencemaran Udara di DKI Jakarta" yang ditulis oleh Syekh S A Umri, Muhammad S Firdaus, dan Aji Primajaya pada tahun 2021. Permasalahan pada penelitian tersebut adalah kualitas udara yang buruk di DKI Jakarta, yang dipengaruhi oleh aktivitas manusia, pabrik industri, dan operasi pembangkit listrik berbahan bakar fosil. Solusinya adalah dengan menggunakan teknik data mining untuk mengklasifikasikan tingkat Indeks Standar Pencemar Udara (ISPU) berdasarkan lima jenis pencemar utama: karbon monoksida (CO), sulfur dioksida (SO_2), nitrogen dioksida (NO_2), ozon permukaan (O_3), dan partikel debu (PM10). Penelitian tersebut membandingkan beberapa algoritma klasifikasi, yaitu *Neural Network*, *Support Vector Machine*, *K-Nearest Neighbors*, *Naive Bayes*, dan *Decission Tree*, dengan menggunakan *T-Test* sebagai metode uji parametrik. Berdasarkan artikel jurnal tersebut metode yang akan diadopsi adalah penggunaan 5 algoritma yaitu *Support Vector Machine*, *K-Nearest Neighbors*, *Naive Bayes*, dan *Decission Tree*. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan dataset ISPU dengan tahun yang berbeda, yaitu dari tahun 2013-2023, dan menambahkan 2 algoritma yaitu *Random Forest* dan XGBoost dengan menggunakan *hyperparameter gridsearchcv*.

2. Berdasarkan artikel jurnal yang berjudul " Klasifikasi Tingkat Pencemaran Udara Pada Sektor Industri Dengan Metode *Random Forest*" yang ditulis oleh Suci Cahaya Hati Nasution, Fibri Rakhmawati, dan Riri Syafitri Lubis pada tahun 2021, salah satu permasalahan utama dalam penelitian ini adalah bagaimana mengklasifikasikan tingkat pencemaran udara dengan akurat dan memahami pengaruh variabel-variabel seperti sulfur dioksida (SO_2), nitrogen dioksida (NO_2), opasitas, dan partikulat pada tingkat pencemaran tersebut.

Solusinya adalah penggunaan metode *Random Forest* untuk klasifikasi tingkat pencemaran udara. *Random Forest* merupakan metode *ensemble* yang dapat meningkatkan akurasi klasifikasi. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penggunaan algoritma *Random Forest*. Berdasarkan analisa gap sebelumnya, kontribusi penelitian ini adalah dengan menggunakan dataset ISPU kualitas udara di Jakarta dan penambahan lima algoritma klasifikasi yaitu KNN, *Naïve Bayes*, SVM, *Decision Tree*, dan XGBoost, dan menggunakan optimasi *hyperparameter* dan *feature selection*

3. Berdasarkan artikel jurnal yang berjudul " Penerapan Metode *Extreme Gradient Boosting* (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit "yang ditulis oleh Sri Erlina Yulianti, Oni Soesanto, dan Yuana Sukmawaty , pada tahun 2022, Masalah pada penelitian ini adalah kartu kredit macet, yang merupakan ketidakmampuan pengguna kartu kredit untuk membayar tagihan. Masalah ini dapat menyebabkan kerugian baik bagi pengguna kartu kredit maupun penyedia layanan kartu kredit. Solusinya adalah penggunaan teknik *machine learning*, khususnya metode klasifikasi XGBoost (*Extreme Gradient Boosting*). Metode ini digunakan untuk mengklasifikasi nasabah kartu kredit yang berpotensi macet. Penelitian ini juga menekankan pentingnya tuning *hyperparameter* XGBoost untuk meningkatkan kinerja model klasifikasi. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penggunaan XGBoost beserta tuning *hyperparameter*. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan topik, dan data yang berbeda yaitu mengenai tingkat kualitas udara di Jakarta.

4. Berdasarkan artikel jurnal yang berjudul "Peningkatan Akurasi *K-Nearest Neighbor* Pada Data Index Standar Pencemaran Udara Kota Pekanbaru" yang ditulis oleh Yuliska, dan Khairul Umam Syaliman pada tahun 2020. Permasalahan utama pada penelitian tersebut adalah kelemahan metode *K-Nearest Neighbor* (KNN) konvensional. Kelemahan secara spesifik yaitu metode KNN memberikan bobot yang sama pada setiap atribut. Hal ini menyebabkan atribut yang tidak relevan mendapatkan pengaruh yang sama

dengan atribut yang relevan, yang dapat menurunkan akurasi klasifikasi. Solusinya adalah melibatkan penggabungan metode pembobotan atribut (*attribute weighting*) dengan *local mean*, serta menggunakan *Gain Ratio* untuk menghitung bobot atribut dengan menggunakan data Index Standar Polusi Udara di Kota Pekanbaru untuk menguji kinerja metode yang diusulkan. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penggunaan algoritma *K-Nearest Neighbor*. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan data dan tempat penelitian yang berbeda dan penambahan algoritma serta penambahan optimasi.

5. Berdasarkan artikel jurnal yang berjudul "Analisis Klasifikasi Indeks Kualitas Udara Kota di Indonesia Menggunakan Metode *K-nearest Neighbor* dan *Naïve Bayes* " yang ditulis oleh Ali maulana, Ade Irma Purnamasari, dan Irfan Ali pada tahun 2024, Penelitian ini mengidentifikasi masalah polusi udara di kota-kota besar di Indonesia, yang berdampak negatif pada kesehatan manusia dan ekosistem. Tingginya tingkat polusi udara di Indonesia, yang termasuk dalam 26 negara dengan polusi tertinggi menurut data *AirVisual* oleh AQI, menunjukkan pentingnya pemantauan kualitas udara yang efektif. Solusinya adalah penggunaan dan evaluasi dua metode klasifikasi yang berbeda, yaitu *K-Nearest Neighbo*r (KNN) dan *Naïve Bayes*. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penerapan *Framework* KDD, dan penggunaan algoritma KNN dan *Naïve Bayes*. Berdasakan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan data dan tempat penelitian yang berbeda dan penambahan algoritma serta penambahan optimasi.

6. Berdasarkan artikel jurnal yang berjudul "Klasifikasi Kualitas Udara Dengan Metode *Support Vector Machine*" yang ditulis oleh Ade Silvia Handayani, Sopian Soim, Theresa Enim Agusdi, Rumiasih, dan Ali Nurdin pada tahun 2022, masalah dari penelitian ini adalah bahwa pencemaran udara di Indonesia dapat menjadi masalah serius yang membahayakan baik lingkungan maupun kesehatan. Metode yang ada untuk mengklasifikasikan kualitas udara belum

mencapai akurasi yang optimal. Solusinya adalah menggunakan metode Support Vector Machine (SVM) untuk mengklasifikasikan kualitas udara berdasarkan data yang diperoleh dari sensor yang mengukur parameter seperti CO, CO2, HC, PM10, suhu, dan kelembaban. Berdasarkan artikel jurnal tersebut akan. Dijadikan referensi untuk penggunaan algoritma SVM. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah menambahkan penggunan *hyperparamter* dan *feature selection SelectKBest*, sehingga akurasi menjadi optimal.

7. Berdasarkan artikel jurnal yang berjudul *"Implementation of Support Vector Machine Algorithm for Identifying Facial Skin Types"* yang ditulis oleh Marisa Tri Utami, Julio Christian Young, dan Arya Wicaksana pada tahun 2020, Masalah utama yang dihadapi dalam penelitian ini adalah identifikasi jenis kulit wajah yang sering kali sulit dilakukan oleh individu tanpa alat bantu yang tepat. Solusinya adalah pengembangan aplikasi yang menggunakan algoritma *Support Vector Machine* (SVM) untuk mengklasifikasikan jenis kulit wajah. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penggunaan algoritma SVM. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan topik dan data yang berbeda dan menambahkan optimasi pada algoritma SVM agar hasil akurasinya lebih optimal.

8. Berdasarkan artikel jurnal yang berjudul " *Hyperparameter Tuning* menggunakan *GridsearchCV* pada *Random Forest* untuk Deteksi *Malware*" yang ditulis oleh Iik Muhamad Malik Matin ,pada tahun 2023, masalah penelitian tersebut adalah kinerja model *Random Forest* dalam mendeteksi malware tidak optimal karena hanya menggunakan satu nilai untuk setiap hyperparameter. Hal ini mengakibatkan pengukuran performa yang tidak maksimal. Solusinya adalah melakukan h*yperparameter* tuning menggunakan metode *GridsearchCV*. *GridsearchCV* memungkinkan pemindaian sejumlah *hyperparameter* yang dipilih untuk menemukan kombinasi terbaik yang meningkatkan performa model *Random Forest*. Berdasarkan artikel jurnal

tersebut akan dijadikan referensi untuk penggunaan *Random Forest* dan *Hyperparameter*. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan topik dan data yang berbeda dan menambahkan beberapa nilai parameter agar hasilnya optimal. Nilai parameter yang ditambahkan pada algoritma *Random Forest* adalah `*criterion*` dengan `gini`, `*min_samples_leaf*` dengan nilai 1, `*min_samples_split*` dengan nilai 2, dan `n_estimators` dengan nilai 100.

9. Berdasarkan artikel jurnal yang berjudul " Penerapan Algoritma *K-Nearest Neighbor* dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek" yang ditulis oleh Robi Nurhidayat, dan Kania Evita Dewi ,pada tahun 2023, Permasalahan pada penelitian ini adalah pengklasifikasian menjadi *multi-label*, dan data yang tidak seimbang. Solusinya adalah menggunakan *binary relevance,* untuk mengatasi masalah klasifikasi. Sedangkan untuk mengatasi masalah data yang tidak seimbang menggunakan metode *Random Over Sampling*, dan untuk menemukan nilai parameter model menggunakan teknik *gridsearch*. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penggunaan split data menggunakan presentasi 80% dan 20% , dan sebagai acuan penggunaan *hyperparameter gridsearchcv.* Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan topik dan data yang berbeda dan menambahkan beberapa nilai parameter pada KNN yaitu `*metric*` dengan *manhattan* , `*neighbors*` dengan nilai 9, `*weights*` dengan *distance.* Tujuannya adalah agar akurasi menjadi lebih optimal.

10. Berdasarkan artikel jurnal yang berjudul " *Feature Selection Using New Version of V-Shaped Transfer Function for Salp Swarm Algorithm in Sentiment Analysis*" yang ditulis oleh Dinar Ajeng Kristiyanti ,Imas Sukaesih Sitanggang , Annisa dan Sri Nurdiati, pada tahun 2023, Masalah pada penelitian ini adalah tantangan dalam *feature selection* pada analisis sentimen yang kaya fitur. Tantangan ini meliputi pemilihan set fitur yang

paling relevan, memberikan informasi tentang hubungan antar fitur yang informatif, serta menghilangkan *noise* dari dataset berdimensi tinggi untuk meningkatkan kinerja *classifier*. Permasalahan ini mengarah pada kebutuhan untuk menemukan metode optimasi yang efektif untuk memilih subset fitur yang signifikan. Solusinya adalah penggunaan versi biner dari algoritma optimasi metaheuristik yang berbasis *Swarm Intelligence*, yaitu *Salp Swarm Algorithm* (SSA), sebagai metode pemilihan fitur dalam analisis sentiment. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penggunaan *Feature Selection*. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan topik dan data yang berbeda dan menggunakan *feature selection* SelectKBest.

11. Berdasarkan artikel jurnal yang berjudul " *Heart Disease Prediction System using hybrid model of Multi-layer perception and XGBoost algorithms*" yang ditulis oleh Israa Nadheer, pada tahun 2024. Permasalahan pada penelitian inni adalah kompleksitas data medis yang besar dan beragam, yang sering kali bersifat tidak lengkap. Kompleksitas ini mempengaruhi kinerja dan akurasi prediksi penyakit jantung. Solusinya adalah dengan menggunakan model hibrid yang menggabungkan algoritma *Multi-layer Perceptron* (MLP) dan XGBoost. Berdasarkan artikel jurnal tersebut akan dijadikan referensi untuk penggunaan algoritma Hibrida XGBoost. Berdasarkan Analisa gap penelitian sebelumnya , Kontribusi penelitian ini adalah dengan menggunakan topik dan data yang berbeda dan menggunakan model hibrida XGBoost dengan menambahkan optimasi pada model hibrida XGBoost. Tujuan penggabungan algoritma hibrida XGBoost dengan algoritma tunggal lainnya dan penggunaan optimasi agar performa kelima algoritma tersebut menghasilkan akurasi yang optimal.

2.2 Tinjauan Pustaka

2.2.1 Kualitas Udara

Kualitas udara merujuk pada kondisi keseluruhan udara di suatu wilayah yang memperhitungkan berbagai faktor fisik, kimia, dan biologis yang mempengaruhinya. Ini mencakup kandungan polutan seperti gas beracun, partikel-partikel halus, dan zat kimia lainnya yang dapat membahayakan kesehatan manusia dan lingkungan. Kualitas udara yang buruk dapat menyebabkan berbagai masalah kesehatan, termasuk gangguan pernapasan, penyakit jantung, dan bahkan kematian [24].

2.2.2 Polusi Udara

Polusi udara merupakan suatu kondisi di mana kualitas udara di lingkungan alami menurun hingga mencapai tingkat tertentu. Penurunan ini disebabkan oleh masuknya komponen tambahan seperti gas atau energi ke dalam udara, yang dipicu oleh tindakan atau aktivitas manusia [25] . Polusi udara berasal dari limbah yang dihasilkan oleh aktivitas manusia untuk memenuhi kebutuhan mereka, baik dalam sektor produksi maupun transportasi. Dengan peningkatan jumlah penduduk, terjadi peningkatan limbah yang mencemari udara [26]. Polutan yang menjadi fokus utama dalam kaitannya dengan kesehatan masyarakat yaitu partikulat, karbon monoksida, ozon, nitrogen dioksida, dan sulfur dioksida [27].

2.2.3 Indeks Standar Pencemar Udara (ISPU)

Indeks Standar Pencemar Udara (ISPU) merupakan sebuah parameter atau indeks yang digunakan untuk mengukur tingkat pencemaran udara di suatu daerah atau lokasi tertentu yang sebelumnya diatur oleh Keputusan Nomor 107/1997 dari Kepala Badan Pengelola Dampak Lingkungan mengenai pedoman teknis untuk menghitung dan melaporkan informasi ISPU. ISPU memperhitungkan 5 parameter yaitu partikulat matter (PM10), karbon monoksida (CO), sulfur dioksida (SO2), nitrogen dioksida (NO2), dan ozon (O3). Peraturan terbaru mengenai ISPU diatur dalam Peraturan Menteri Lingkungan Hidup dan Kehutanan Nomor 14/2020 , dengan penambahan 2 parameter lain yaitu partikulat (PM2.5) dan

hidrokarbon (HC), serta perubahan pada batas konsentrasi 5 parameter lainnya [28]. Nilai ISPU akhir ditentukan berdasarkan parameter dominan, yang memiliki nilai ekuivalen ISPU tertinggi di antara tujuh 7 parameter yang ada. 5 kategori nilai ISPU menunjukkan kualitas udara ambien: baik (0–50), sedang (51–100), tidak sehat (101–200), sangat tidak sehat (201–300), dan berbahaya (>300) [29].

2.3 *Framework* , Algoritma Klasifikasi dan Metode Evaluasi

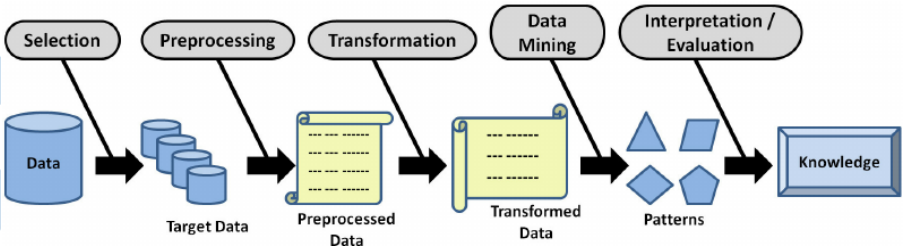
2.3.1 *Framework*

2.3.1.1 Klasifikasi

Klasifikasi merupakan teknik *data mining* untuk mengelompokkan atau menata objek, data, atau informasi ke dalam kategori atau kelas berdasarkan karakteristik atau atribut tertentu. Tujuan utama dari *classification* adalah untuk menyederhanakan kompleksitas dan memudahkan pengelompokan entitas-entitas tersebut agar dapat diidentifikasi atau dikelola lebih efisien [30].

2.3.1.2 Knowledge Discovery in Database

Knowledge Discovery in Database merupakan suatu metode yang dipakai untuk memperoleh pemahaman baru dari suatu basis data. Tujuan akhir dari proses KDD adalah untuk menyediakan informasi yang dapat digunakan sebagai dasar dalam pengambilan keputusan. Dalam tahapan KDD, terdapat serangkaian proses , Pada gambar 2.1 merupakan tahapan KDD [31].



Gambar 2. 1 Tahapan KDD

Sumber: [31]

Berdasarkan Gambar 2.1 , dapat diuraikan langkah-langkah tahapan dalam metode KDD sebagai berikut [32] :

1. *Data Selection*

   Data Selection merupakan tahapan KDD untuk proses penyaringan data yang akan digunakan dalam kegiatan data mining. Setelah itu, data yang telah terpilih akan dipisahkan dari *database* operasional

2. *Data Preprocessing*

   Data Preprocessing merupakan tahapan KDD yang mencakup langkah-langkah seperti membersihkan data untuk memperbaiki kesalahan, menghapus duplikasi, dan mengecek konsistensi data yang tidak sesuai. Pada tahap ini, juga dilakukan pengayaan data dengan informasi atau data eksternal yang diperlukan untuk meningkatkan kualitas data yang sudah ada.

3. *Transformation Data*

   Transformation Data merupakan tahapan KDD dengan mengubah bentuk data yang memiliki entitas yang belum jelas menjadi bentuk data yang siap dan valid untuk diolah pada langkah selanjutnya, yaitu *data mining*.

4. *Data Mining*

   Data Mining merupakan tahapan KDD untuk proses pengelolaan data menjadi informasi dengan melibatkan berbagai metode dan teknik. *Data mining* merupakan suatu proses pengelolaan data besar dengan tujuan menghasilkan informasi akurat, memudahkan pemecahan masalah, dan mendukung pengambilan keputusan.

5. *Knowledge Interpretation atau Evaluation*

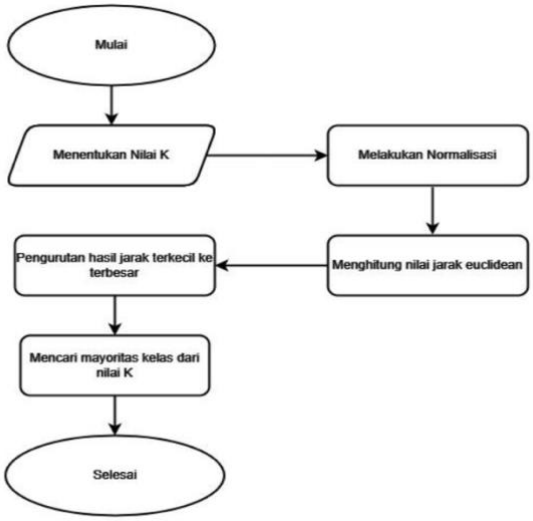
   Knowledge Interpretation atau Evaluation merupakan tahapan KDD untuk menghasilkan informasi yang dihasilkan dari proses *data mining* untuk mengeliminasi kemungkinan adanya informasi yang kontradiktif dengan fakta yang sudah ada sebelumnya. Informasi tersebut kemudian disajikan dalam bentuk yang mudah dipahami.

2.3.2 Algoritma Klasifikasi

2.3.2.1 *K-Nearest Neighbor (KNN)*

K-Nearest Neighbor adalah pendekatan klasifikasi yang mencari data latih yang memiliki kemiripan relatif dengan data uji. *K-Nearest Neighbor* dikategorikan sebagai teknik klasifikasi *lazy learning* karena tidak membangun model klasifikasi sebelumnya [33]. Dalam menentukan hasil klasifikasi, algoritma *K-Nearest Neighbor* memperhatikan jarak terdekat dari objek dengan setiap kelompok .

Terdapat lima metode untuk mencari jarak terdekat, termasuk penggunaan Jarak *Euclidean*, Jarak *Manhattan*, Jarak *Cosine*, Jarak *Correlation*, dan Jarak *Hamming*. K-Nearest Neighbor merupakan salah satu dari beberapa algoritma *supervised learning*. *K-Nearest Neighbor* tergolong ke dalam algoritma *supervised learning* karena metodenya memanfaatkan label atau kelas pada data latih untuk melakukan klasifikasi pada data *testing* [34]. Dengan menggunakan informasi yang ada pada data *training*, algoritma ini dapat memprediksi kelas atau label yang sesuai untuk data baru yang belum diketahui kelasnya. Pada Gambar 2.2 merupakan prosedur dari proses perhitungan *K-Nearest Neighbor*.



Gambar 2. 2 Prosedur K-Nearest Neighbor

Sumber : [35]

Berikut adalah langkah-langkah perhitungan algoritma *K-Nearest Neighbor* (KNN) untuk klasifikasi [35]:

1. Menentukan nilai k
2. Melakukan normalisasi min-max untuk mengubah data sehingga nilainya berada dalam rentang antara 0 dan 1.
   Perhitungan normalisasi menggunakan metode *min-max*

$$normalized = \frac{\text{Data}_{x} - \text{Data}_{min}}{\text{Data}_{max} - \text{Data}_{min}} \quad (2.1)$$

Rumus 2. 1 Perhitungan normalisasi min-max

   Dimana Data_{x} adalah data yang akan dihitung normalisasinya berdasarkan kolom datanya. Data_{min} merujuk pada nilai terkecil dalam kolom yang sama, sedangkan Data_{max} adalah nilai terbesar dalam kolom yang sama dengan data yang akan dinormalisasi. Proses normalisasi ini bertujuan untuk mengubah Data_{min} sehingga berada dalam skala tertentu berdasarkan rentang nilai antara Data_{min} dan Data_{max}
3. Menghitung jarak *Euclidean* antara data *training* dan data *testing*.
4. Mengurutkan hasil jarak data dari yang terkecil hingga yang terbesar.
5. Menemukan kelas mayoritas dari K *neighbor* dan menggunakannya sebagai hasil prediksi.
   Untuk perhitungan *Euclidean* , menggunakan rumus pada persamaan 2.2

$$d_{euclidean}(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \quad (2.2)$$

Rumus 2. 2 Rumus Euclidean

   Dimana $d_{euclidean}$ merupakan salah satu metode pengukuran jarak yang umum digunakan dalam analisis data. Metode ini digunakan untuk mengukur jarak antara dua titik dalam ruang Euclidean. Dalam konteks penggunaannya, i merupakan

jumlah dari set data yang akan diukur jaraknya, sedangkan x dan y mewakili jumlah set data uji dan latih, secara berturut-turut.

2.3.2.2 *Decision Tree*

Decision tree adalah model prediktif di mana hasil akhirnya diklasifikasikan berdasarkan struktur pohon hierarkis. Model ini mampu menganalisis variabel data, baik yang bersifat nominal maupun numerik, secara bersamaan. *Decision tree* bekerja dengan mencari solusi permasalahan dan membentuk struktur pohon di mana kriteria-kriteria berfungsi sebagai simpul yang saling terhubung. Setiap pohon memiliki cabang, dan setiap cabang mewakili suatu atribut yang harus dipenuhi untuk menuju cabang berikutnya, dan proses ini berlanjut hingga mencapai simpul daun (tidak ada cabang lagi) [36]. *Decision Tree* akan mengubah data ke dalam bentuk visual yang berupa diagram pohon keputusan serta aturan-aturan keputusan yang terkandung di dalamnya [37] . Proses eksplorasi data menggunakan algoritma *Decision Tree* dimulai dengan menghitung *Gain* dan *Entropy* dari setiap atribut pada data *training*, yang pada akhirnya menghasilkan *Gain Ratio.* Rumus 2.3 merupakan rumus Information Gain yang digunakan untuk memilih atribut yang paling informatif.

$$Gain(S, A) = Entropy(S) - \sum_{i-1}^{n} \frac{|S_i|}{|S|} * Entropy\,(S_i) \qquad (2.3)$$

Rumus 2. 3 Rumus Information Gain

S menggambarkan sebuah himpunan kasus dan atribut A dalam himpunan tersebut. $|S_i|$ mewakili jumlah kasus dalam partisi i, sedangkan $|S|$ menunjukkan total jumlah kasus dalam himpunan S.

Sementara itu, perhitungan nilai *Entropy* dilakukan menggunakan persamaan 2.4

$$Entrophy(S) = \sum_{i=1}^{n} - pi * \log_2 pi \quad (2.4)$$

Rumus 2. 4 Perhitungan Entropy

Dimana S adalah himpunan kasus yang sedang dipertimbangkan, dan n adalah jumlah partisi yang dibuat berdasarkan atribut A. Setiap partisi dinyatakan sebagai S_i, dengan i menunjukkan nomor partisi tersebut. Kemudian, pi merupakan proporsi dari kasus-kasus yang termasuk dalam partisi S_i terhadap total kasus dalam himpunan S.

Atribut yang memiliki *Gain Ratio* tertinggi dipilih untuk membentuk simpul akar. Perhitungan nilai *Gain* dan *Entropy* untuk setiap atribut dengan menghapus atribut yang sudah dipilih sebelumnya. Atribut yang memiliki *Gain Ratio* tertinggi kemudian dipilih untuk membentuk simpul internal. Langkah-langkah perhitungan ini diulangi hingga semua atribut memiliki kelas. Jika semua atribut atau pohon telah memiliki kelas, tampilkan pohon keputusan awal dan hasilkan aturan keputusan awal [38].

2.3.2.3 *Naïve bayes*

Naive Bayes adalah metode klasifikasi probabilistik yang sederhana yang menghitung serangkaian probabilitas dengan menggabungkan frekuensi dan kombinasi nilai dari data yang diberikan, menggunakan Teorema Bayes dan mengasumsikan bahwa semua atribut adalah independen satu sama lain atau tidak saling bergantung terhadap nilai pada variabel kelas [39]. Rumus 2.5 merupakan rumus dari Teorema Bayes.

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad (2.5)$$

Rumus 2. 5 Rumus Teorema Bayes

Keterangan :

Dimana *X* adalah data dengan kelas yang belum diketahui, sementara *H* merupakan suatu kelas yang spesifik atau hipotesis data. Kemudian P(H | X) menggambarkan probabilitas dari hipotesis H berdasarkan kondisi X. Probabilitas awal dari hipotesis H dinyatakan sebagai P(H). Selanjutnya, P(X | H) menunjukkan probabilitas dari X berdasarkan hipotesis H, dan P(X) adalah probabilitas dari data X itu sendiri.

2.3.2.4 *Support Vector Machine*

Support Vector Machine (SVM) adalah salah satu metode klasifikasi yang bertujuan untuk menemukan *hyperplane* dengan margin terbesar. *Hyperplane* adalah suatu garis (atau bidang) yang digunakan untuk memisahkan data antara kelas atau kategori. Margin adalah jarak antara *hyperplane* dengan data terdekat dari setiap kelas. Data yang paling dekat dengan *hyperplane* disebut sebagai *support vector* [40]. Rumus 2.6 merupakan rumus dari SVM.

$$f(xd) = \sum_{i=1}^{ns} \alpha_i y_i \overrightarrow{\chi_\iota} \overrightarrow{\chi_d} + b \quad (2.6)$$

Rumus 2. 6 Rumus SVM

Dimana *ns* adalah jumlah dari *support vector* yang digunakan dalam model. Setiap titik data memiliki nilai bobot yang terkait, dinyatakan sebagai αi. Kelas dari data tersebut diwakili oleh *y*i, sedangkan $\vec{x}i$ adalah variabel support vector. Data yang akan diklasifikasikan dilambangkan sebagai $\vec{x}d$. Selain itu, nilai eror atau bias dalam model ini dinyatakan dengan *b*.

2.3.2.5 *Random Forest*

Random Forest adalah metode *ensemble* yang terdiri dari sejumlah pohon keputusan *(decision tree*), yang bergabung untuk mengklasifikasikan data ke dalam kelas-kelas tertentu. Algoritma ini

mengandalkan *decision tree* sebagai dasarnya, di mana data masukan menjadi akar dan diolah menjadi daun-daun yang menentukan kelas-kelasnya. Melalui penggunaan sejumlah *Decision tree*, *Random Forest* dapat meningkatkan akurasi dalam mengklasifikasikan data latih yang besar [41]. Langkah pertama dalam pembentukan *Decision Tree* adalah dengan menghitung entropy dan *gain*. *Entropy* digunakan untuk menilai tingkat ketidakmurnian atribut, sementara *information gain* mengukur seberapa banyak informasi yang diperoleh dengan membagi simpul. Rumus 2.7 merupakan rumus entropy pada *Random Forest.*

$$Entropy\ (S) = \sum_{i=1}^{n} - p_{\mathrm{i}}\ \log_2 p_{\mathrm{i}} \qquad (2.7)$$

Rumus 2. 7 Rumus Enthropy Random Forest

S merepresentasikan himpunan dataset yang digunakan. Variabel n menunjukkan banyaknya jumlah kelas yang ada dalam dataset tersebut. Sedangkan pi adalah probabilitas dari kelas ke-i dalam output S. Probabilitas ini menggambarkan seberapa sering kelas tersebut muncul dalam himpunan dataset S.

Sementara itu, perhitungan nilai *Information Gain* dilakukan menggunakan persamaan 2.8

$$Gain\ (A) = Entropy\ (S) = \sum_{i-1}^{n} \frac{|\mathrm{S_i}|}{|S\ |}\ x\ Entropy\ (\mathrm{S_i}) \qquad (2.8)$$

Rumus 2. 8 Rumus Gain Random Forest

Atribut yang dilambangkan dengan A merupakan karakteristik atau fitur dalam himpunan dataset S. $|S\ i|$ menunjukkan jumlah sampel yang memiliki nilai ke-i dalam dataset tersebut, sedangkan $|S\ |$ menggambarkan total jumlah data yang ada dalam himpunan dataset S.

2.3.2.6 XGBoost

XGBoost adalah salah satu teknik *boosting* di mana serangkaian pohon keputusan dibangun secara bertahap. Setiap pohon keputusan yang dibangun berikutnya akan dipengaruhi oleh pohon keputusan sebelumnya dalam *ensemble* [42]. XGBoost dirancang untuk mengatasi *overfitting* dan meningkatkan efisiensi komputasi, dengan menyederhanakan fungsi objektif, yang memungkinkan penggabungan elemen-elemen prediksi dan regularisasi. Regularisasi digunakan untuk mengontrol kompleksitas model dan mencegah *overfitting* [43] . Berikut merupakan rumus 2.9 yaitu rumus persamaan fungsi objektif pada XGBoost.

$$\min \mathrm{L}^{(t)}\ (y, \mathrm{y_i}^{(t)}) = \min\ (\sum_{i=1}^{n} l\ \left(\mathrm{y_i}, \mathrm{y_i}^{(t)} + \ (\sum_{k=1}^{l} \Omega\ (f_k\ )\right) \quad (2.9)$$

Rumus 2. 9 Rumus Persamaan Fungsi Objektif

Dimana $L^{(t)} = \sum_{i=1}^{n} l$ adalah fungsi kerugian dengan $\mathrm{y_i}$ adalah nilai riil dan $\mathrm{y_i}^{(t)}$ adalah nilai prediksi. Sedangkan $(\sum_{k=1}^{l} \Omega\ (f_k\ )$ merupakan istilah regularisasi (penalti) dari model yang digunakan untuk menilai tingkat kompleksitas keseluruhan model.

Istilah regularisasi dapat ditentukan menggunakan rumus berikut:

$$\Omega\ (f_k\ ) = \gamma T_k\ + \frac{1}{2}\ \lambda\ \sum_{j=1}^{T_k} \omega_{kj} \quad (2.10)$$

Rumus 2. 10 Regularisasi

T_k mengacu pada simpul daun pada pohon ke-k. γ mengacu pada koefisien pengurangan jumlah simpul daun T, ω_{kj} merupakan nilai simpul daun ke-j pada pohon ke-k. Selain itu, terdapat λ, yaitu koefisien penalti yang diterapkan pada nilai simpul daun ω. Koefisien penalti ini memberikan pengaruh tertentu pada nilai ω_{kj} ,

yang dapat mempengaruhi keseluruhan struktur dan evaluasi pohon ke-k.

2.3.2.7 Hibrida

Algoritma hibrida adalah gabungan dari dua atau lebih teknik atau pendekatan yang berbeda dalam satu sistem atau metode untuk meningkatkan kinerja atau mengatasi kelemahan masing-masing teknik tersebut. Dalam konteks pembelajaran mesin dan optimisasi, algoritma hibrida menggabungkan teknik-teknik optimisasi dengan teknik-teknik pembelajaran mesin untuk menciptakan sebuah pendekatan yang lebih kuat dalam menyelesaikan masalah-masalah kompleks[44].

2.3.2.8 *Hyperparameter*

Hyperparameter adalah parameter yang nilainya ditentukan sebelum proses pembelajaran dimulai . Dalam rangka meningkatkan kualitas klasifikasi, penting untuk memilih parameter yang tepat dalam model yang diusulkan. *Hyperparameter* digunakan untuk mengelola berbagai aspek dalam pembelajaran mesin yang memiliki dampak besar terhadap kinerja dan hasil model. Terdapat beberapa jenis optimasi *hyperparameter*, seperti *GridSearchCV*, *RandomSearchCV*, optimasi bayesian, dan optimasi evolusioner [45].

Ketika membangun model menggunakan algoritma KNN, *Naïve Bayes*, SVM, *Decision Tree*, *Random Forest*, dan *XGBoost*, sebaiknya dilakukan penyesuaian parameter untuk meningkatkan kualitas model dan kinerjanya pada data uji. Penyetelan parameter ini bertujuan agar model dapat bekerja lebih efektif dan memberikan hasil yang lebih akurat [43]. Penyetelan parameter didasarkan pada penelitian [46], [47], [48], [20], [49]. Beberapa parameter yang bisa disesuaikan dalam penelitian ini untuk meningkatkan kinerja model pada data uji adalah :

a) *K-Nearest Neighbor* (KNN)

- *Metric* : Menentukan cara mengukur jarak antara titik-titik data.
- *n_neighbors* : Menentukan jumlah tetangga terdekat yang dipertimbangkan dalam klasifikasi atau regresi
- *weights* : menentukan bagaimana bobot diberikan kepada tetangga terdekat, apakah seragam, berdasarkan jarak, atau menggunakan fungsi pembobotan khusus.

b) *Naïve Bayes*

- *var_smoothing* : Mengontrol penambahan nilai kecil ke varians dari fitur data, guna menghindari pembagian dengan nol atau nilai yang sangat kecil yang dapat menyebabkan instabilitas numerik dalam perhitungan probabilitas.

c) *Support Vector Machine* (SVM)

- C : Mengontrol *trade-off* antara memaksimalkan margin pemisahan dan meminimalkan kesalahan klasifikasi
- Gamma : Menentukan seberapa jauh pengaruh dari satu titik data tunggal akan berdampak.
- Kernel : Menentukan transformasi data dan ruang fitur baru di mana model akan mencari *hyperplane* pemisahan.

d) Decision Tree

- *Criterion* : Menentukan fungsi atau metrik yang digunakan untuk mengukur kualitas pemisahan (*split*) pada setiap node dalam pohon keputusan.
- *Max Depth* : Menentukan kedalaman maksimum setiap pohon keputusan dalam model.

- *Minimum Samples Leaf* : Menentukan jumlah sampel minimum yang diperlukan dalam sebuah node daun pada pohon keputusan.
- *Minimum Samples Split* : Menentukan jumlah minimum sampel yang diperlukan untuk membagi sebuah node pada pohon keputusan.

e) *Random Forest*

- *Criterion* : Menentukan fungsi atau metrik yang digunakan untuk mengukur kualitas pemisahan (*split*) pada setiap node dalam pohon keputusan
- *Max Depth* : Menentukan kedalaman maksimum setiap pohon keputusan dalam model.
- *Minimum Samples Leaf* : Menentukan jumlah sampel minimum yang diperlukan dalam sebuah node daun pada pohon keputusan.
- *Minimum Samples Split* : Menentukan jumlah minimum sampel yang diperlukan untuk membagi sebuah node pada pohon keputusan
- *n_estimators* : Menentukan jumlah pohon keputusan yang akan digunakan dalam model *ensemble.*

f) XGBoost

- *Learning Rate*: Menentukan seberapa besar pengaruh masing-masing model terhadap model berikutnya dalam setiap iterasi.
- *Max Depth* : Menentukan kedalaman maksimum setiap pohon keputusan dalam model.
- *Minimum Child Weight* : Menentukan jumlah minimum sampel yang diperlukan di setiap cabang pohon keputusan.
- *Gamma* : Menentukan ambang batas minimal untuk pemisahan cabang pada pohon keputusan

- *n_ estimator* : Menentukan jumlah pohon keputusan yang akan digunakan dalam model *ensemble.*

2.3.2.9 *Feature Selection*

Feature Selection merupakan proses pemilihan atribut dari data yang digunakan dalam *machine learning*. Tujuannya adalah untuk memilih atribut yang secara signifikan berkontribusi pada hasil akhir dan menghilangkan yang tidak berguna. Proses ini dapat meningkatkan akurasi dan efisiensi model, serta membantu mengurangi *overfitting* [50]. Ada berbagai metode yang tersedia untuk melakukan seleksi fitur, seperti reduksi dimensi melalui teknik filter, reduksi dimensi melalui teknik pembungkus, dan reduksi dimensi melalui teknik tertanam. Kelebihan dan kekurangan setiap metode akan bervariasi tergantung pada konteks aplikasinya. Oleh karena itu, memilih metode seleksi fitur yang tepat adalah krusial untuk mencapai hasil yang optimal [51].

2.3.2.10 *SelectKBest*

SelectKBest adalah algoritme pemilihan fitur yang digunakan untuk meningkatkan akurasi prediksi atau kinerja pada dataset berdimensi tinggi. Metode ini termasuk dalam kategori *Univariate Feature Selection*, yang memilih fitur-fitur terbaik berdasarkan uji statistik univariat atau uji ANOVA. Uji statistik ini membantu dalam memilih fitur-fitur yang memiliki hubungan terkuat dengan variabel output. *SelectKBest* mengeliminasi semua fitur kecuali yang memiliki skor tertinggi. Dengan kata lain, *SelectKBest* memilih K fitur teratas yang paling relevan dengan variabel target [52].

2.3.3 Metode Evaluasi

2.3.3.1 *Confusion Matrix*

Confusion matrix merupakan metode untuk mengevaluasi performa metode klasifikasi. Dalam penggunaannya untuk mengukur kinerja, terdapat empat istilah yang merepresentasikan hasil proses klasifikasi, yaitu *True positive* (TP), *True negative* (TN), *False positive* (FP), dan *False negative* (FN). *True Negative* (TN) adalah jumlah data negatif yang terdeteksi dengan benar, sementara *False Positive* (FP) adalah data negatif yang salah terdeteksi sebagai data positif [53] . Pada Gambar 2.3 merupakan perhitungan *Confusion Matrix*.

		Keadaan Data Sebenarnya	
		TRUE	**FALSE**
Hasil Prediksi	**TRUE**	TP (*True Positive*) disebut juga *correct result*	FP (*False Positive*) disebut juga *unexpected result/ false alarm*
	FALSE	FN (*False Negative*) disebut juga *missing result*	TN (*True Negative*) disebut juga *correct rejection*

Gambar 2. 3 Komponen Confusion Matrix

Sumber: [53]

Berdasarkan Gambar 2.3 , Confusion matrix terdiri dari empat komponen sebagai berikut [54] :

1. TP (*True Positive*): Kondisi ketika data positif pada kondisi sebenarnya dan juga diprediksi sebagai positif.
2. FP (*False Positive*): Kondisi di mana data negatif pada kondisi sebenarnya, tetapi diprediksi sebagai positif.
3. TN (*True Negative*): Kondisi di mana data negatif pada kondisi sebenarnya dan juga diprediksi sebagai negatif.
4. FN (*False Negative*): Kondisi di mana data positif pada kondisi sebenarnya, namun diprediksi sebagai negatif.

Setelah data *testing* dimasukkan ke dalam *confusion matrix*, dilakukan perhitungan nilai-nilai yang telah dimasukkan tersebut untuk menghitung jumlah Presisi, *Sensitivity* (*recall*), *F1-score*, dan Akurasi .

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.11)$$

Rumus 2. 11 Rumus Akurasi

Rumus 2.11 merupakan Rumus Akurasi, Akurasi merupakan evaluasi sejauh mana hasil pengukuran mendekati nilai sebenarnya. Menentukan tingkat akurasi sangat penting untuk meningkatkan keyakinan terhadap hasil pengukuran dan menilai efektivitas suatu metode.

$$Presisi = \frac{TP}{TP+FP} \quad (2.12)$$

Rumus 2. 12 Rumus Presisi

Rumus 2.12 merupakan Rumus Presisi. Presisi adalah metrik evaluasi yang mengukur seberapa banyak dari data yang diprediksi positif oleh model yang sebenarnya benar-benar positif. Jadi, presisi memberikan gambaran tentang seberapa "presisi" atau "tepat" model dalam mengklasifikasikan data sebagai positif.

$$Recall = \frac{TP}{TP+FN} \quad (2.13)$$

Rumus 2. 13 Rumus Recall

Rumus 2.13 merupakan rumus *Recall*. *Recall* atau sensitivitas berfungsi untuk mengukur seberapa banyak *instance* yang sebenarnya positif yang berhasil diprediksi oleh model.

$$F1 - Score = \frac{2\ x\ Recall\ x\ Precision}{Recall\ x\ Precision} \quad (2.14)$$

Rumus 2. 14 Rumus Recall

Rumus 2.14 merupakan Rumus F1-*Score*. F1-*Score* merupakan keseimbangan antara presisi dan *recall*. Metrik ini berfungsi untuk mendapatkan gambaran komprehensif tentang kinerja model, terutama ketika tidak dapat mengorbankan salah satu dari presisi atau *recall*.

2.4 Tools

2.4.1 *Jupyter Notebook*

Jupyter Notebook adalah *cell-based programming envi-ronment* yang memungkinkan pengguna untuk menulis dan menjalankan kode, visualisasi output kode, menambahkan *hyperlink*, serta membuat catatan dengan menyatukan gambar di antaranya. Bahasa pemrograman *Python* dapat dieksekusi dalam lingkungan ini, dan berbagai paket *Python* seperti *pandas*, *plotly*, *numpy*, dan lainnya dapat diimpor untuk memberikan akses ke analisis data yang kuat dan alat *machine learning* [55]

2.4.2 *Python*

Python adalah bahasa pemrograman yang sangat populer dan serbaguna. Dikembangkan pada awal tahun 1990 oleh Guido van Rossum, *Python* memiliki sintaks yang sederhana dan mudah dipahami, menjadikannya pilihan yang ideal bagi pemula dalam dunia pemrograman. Kelebihan *Python* tidak hanya terletak pada kemudahan sintaksnya, tetapi juga pada kemampuannya yang mendukung berbagai paradigma pemrograman, seperti pemrograman berorientasi objek, imperatif, dan fungsional [56].

Kelebihan *Python* juga terletak pada ekosistemnya yang kaya, dengan banyaknya *library* dan *framework* yang mendukung pengembangan perangkat lunak di berbagai domain, mulai dari pengembangan web hingga kecerdasan buatan. Keberlanjutan bahasa ini juga diperkuat oleh komunitas pengembang yang besar dan aktif, serta statusnya sebagai bahasa *open source*. *Python* digunakan secara luas di berbagai industri dan aplikasi, membuktikan fleksibilitasnya dalam menangani tugas-tugas pengembangan perangkat lunak yang beragam [57].