BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Pada Tabel 2.1 merupakan perbandingan pada penelitian terdahulu yang berisikan ringkasan dari isi pada artikel jurnal mengenai *classification* pada prediksi penyakit Hepatitis atau Hepatitis C:

Tabel 2. 1 Penelitian Terdahulu

Penelitian 1	
Judul	*An Ensemble Learning Approach for Enhanced Classification of Patients With Hepatitisand Cirrhosis*, (9, 24491) [5]
Nama Jurnal	*IEEE Access*(Q1)
Tahun	2021
Penulis	Chicco, Davide Jurman, Giuseppe.
Metode	*. RandomForest -. Akurasi 97% *. DecisionTree -. Akurasi 94% *. Linear Regression -. Akurasi 92% *. Method -. two- features RF 95% -. AST/ALT ratio 95%
Hasil	Hasil Penelitian menujukan bahwa model terbaik yang dapat digunakan untuk tingkat akurasi pada hepatitis C adalah dengan menggunakanAlgoritma *Random Forest.*
Kesimpulan	Metode Algorima *Random Forest* memiliki akurasi keseluruhan yang lebih tinggi dibandingkan dengan *Decision Tree* dan *Linear Regression*
Penelitian 2	
Judul	*Hepatitis C Virus prediction based onmachine learning framework: a real-world case study in Egypt Heba* (65:2595–2617) [8]
Nama Jurnal	*Knowledge and Information Systems (Q2)*
Tahun	2023
Penulis	Heba Mamdouh Farghaly, Mahmoud Y. Shams, Tarek Abd El-Hafeez.
Metode	K-fold = 5 *. Naive Bayes -. 92,08%

	*. Random Forest -. 93,13% *. K-NN -. 89,75% *. LR -. 93,01% K-fold = 10 *. Naive Bayes -. 92,66% *. Random Forest -. 94,06% *. K-NN -. 90,80% *. LR -. 92,20%
Hasil	Berdasarkan Hasil *Comparison* dari masing-masing algoritma. *Random forest* memberikan akurasi yang baik pada K-fold 5, begitu juga dengan K-fold 10 memberikan peningkatan akurasi sebesar 0.91%.
Kesimpulan	Perbedaan dalam penggunaan *K-fold* memberikan efek dalam peningkatan khususnya dalam algoritma *Random Forest*, peningkatan ini tergantung pada pemberian *K-fold* yang diberikan. Sehingga memberikan akurasi yang maksimal dalam penggunaan algoritma *random forest.*
Penelitian 3	
Judul	*Applying data mining techniques to classify patients with suspected hepatitis C virus infection. (*193–198) [9]
Nama Jurnal	*Intelligent Medicine (Q3)*
Tahun	2022
Penulis	Zhao, Y.Tian, ShYu, L. Zhang, ZhZhang, W
Metode	Training Data = 80% *. Logistic Regression -. 95.67% *. Naïve Bayes -. 92.43% *. SVM -. 94.59% *. KNN -. 95.67% *. Decision Tree -. 96.75% *. Random Forest -. 97.25%
Hasil	Dari Hasil penelitian yang diberikan tingkat akurasi yang diberikan pada setiap algoritma semuanya diatas 90%. Akurasi tertinggi terdapat pada algorit *Random Forest*

	dengan tingkat akurasi sebesar 97.25%
Kesimpulan	Pada Penelitian ini, pembagian data cukup berpengaruh pada tingkat akurasi dari setiap algoritma. Akurasi yang diberikan untuk *Random Forest* dan *Decision tree* merupakan algoritma 2 terbaik dalam penelitian. Setiap tingkat akurasi pada algoritma murni tanpa bantuan peningkatan yang diberikan pada metode-metode yang mendukung pada algoritma tertentu.
Penelitian 4	
Judul	*Hepatitis classification using support vector machines and random forest.(10, 446-451)* [10]
Nama Jurnal	*IAES Internation al Journal of Artificial Intelligence(IJ-AI) (Q3)*
Tahun	2021
Penulis	Aurelia, Jane Eva Rustam, ZuhermanWirasati, Ilsya Hartini, Sri Saragih, Glori Stephani.
Metode	(Training Data = 80%) *. RandomForest -. Akurasi 98% *. SVM-Linear -. Akurasi98% *. SVM-Polynomial -. Akurasi 98% *. SVM -Gaussian RBF -. Akurasi 100%
Hasil	Berdasarkan dari tabel Perbandingan Penelitian, Bahwa Modelyang miliki akurasi tertinggi yaitu*SVM gaussian RBF*. Pemodelan tersebut mencapai 100%, sedangkanuntuk ketiga pemodelan lainnya berada di 98%.
Kesimpulan	Hasil percobaan menunjukkan bahwa kinerja pengklasifikasi *SVM* dan metode *Random Forest*memprediksi data dengan baik dan benar. Namun, jika berdasarkan hasil tabel yang diberikan, Training data yang diberikan jika 70 atau 80% data random forest memiliki akurasi yang cukup sebanding dengan algoritma SVM dengan berbagai kernelnya.
Penelitian 5	
Judul	Klasifikasi Hepatitis C virus menggunakan Algoritma C4.5. (13, 131 - 136) [6]
Nama Jurnal	Jurnal Disprotek (S5)
Tahun	2022
Penulis	Susanto,Nuri
Metode	*. C4.5. -. Akurasi 94,43% *. C4.5 – AdaBoost. -. Akurasi 95,60%
Hasil	BerdasarkanHasil penelitan, Bahwa Algoritma C4.5

	dengan metode AdaBoost memberikannilai akurasiyang lebih tinggi danlebih baik.
Kesimpulan	Dengan Menggunakan Algoritma C4.5 dapat memudahkan untuk memahami sebuah data yang dijabarkan dengan cabang pohon dalam bentuk klasifikasi. Algorima C4.5 Menggunakan Konsep information *Gain atau Entropy* untuk memilih pembagian dengan optimal. Dengan penggabungan metode adaboost dapat meningkatkan akurasi dari algortima C4.5
Penelitian 6	
Judul	*Hyperparameter Tuning* menggunakan *GridsearchCV* pada *Random Forest* untuk Deteksi *Malware* [11]
Nama Jurnal	*Multinetics*, Vol.9, No.1, (S4)
Tahun	2023
Penulis	Iik Muhamad Malik Matin
Metode	*. Random Forest - Default. -. Akurasi 99.04% *. Random Forest – Hyperparamter Tuning. -. Akurasi 99.23%
Hasil	Berdasarkan hasil penelitan, bahwa algoritma random forest menggunakan hyperparamter tuning memiliki akurasi yang paling tinggi sebesar 99.23%.
Kesimpulan	Dengan menggunakan hyperparameter tuning dapat meningkatkan akurasi sebesar 0.19%. Hal ini menunjukan bahwa hyperparameter tuning berpengaruh dalam meningkatkan akurasi.
Penelitian 7	
Judul	Komparasi Penerapan Metode Bagging dan Adaboost pada Algoritma C4.5 untuk Prediksi Penyakit Stroke[12]
Nama Jurnal	*SISTEMASI: Jurnal Sistem Informasi, Volume 11, Nomor 3* (S3)
Tahun	2022
Penulis	Nur Diana Saputri, Khalid Khalid, Dwi Rolliawati
Metode	*. C4.5 – Default -. Akurasi 92.87% *. C4.5 – AdaBoost -. 94.6%
Hasil	Berdasarkan hasil penelitan, bahwa algoritma C4.5 dengan menggabungkan pemodelan *AdaBoost* memberikan akurasi yang tinggi sebesar 94.6%, dari pada tanpa penggabungan dengan algoritma *AdaBoost*.
Kesimpulan	Dengan penggabungan antara algoritma C4-5 dan AdaBoost, dapat memberikan peningkatan yang cukup signifikan sebesar 1,73%.
Penelitian 8	
Judul	IMPLEMENTASI ALGORITMA ADAPTIVE BOOSTING (ADABOOST) DAN SINGLE LAYER PERCEPTRON (SLP)

	PADA KLASIFIKASI PENYAKIT HEPATITIS-C [13]
Nama Jurnal	ScientiCO : Computer Science and Informatics Journal Vol. 6, No. 2, (2023) E-ISSN: 2620-4118 (S5)
Tahun	2023
Penulis	Anita Desiani1, Sri Indra Maiyanti2, Bambang Suprihatin3, Rifki Kurniawan4, dan Adzra Afifah Nabila5
Metode	K-Fold = 10 *. Adaboost -. Akurasi 85% -. Presisi 64% *. SLP -. Akurasi 95% -. Presisi 40%
Hasil	Berdasarkan hasil penelitan, menujukan bahwa pemodelan *AdaBoost* masih sedikit lebih rendah dibandingkan *Singel Layer Perceptron* (SLP), dengan selih 10%.
Kesimpulan	Walaupun pada akurasi pemodelan AdaBoost masih dibawah dengan SLP, berdasarkan tingkat presisi, pemodelan *AdaBoost* masih di atas pemodelan SLP, ini menunjukan bahwa pembagian K-Fold akan berpengaruh pada tingkat presentase akurasi dan presisi.
Penelitian 9	
Judul	*A Hybrid Classification Algorithm for Abdomen Disease Prediction* [7]
Nama Jurnal	*ASEAN Journal of Science and Engineering*, 3(3) (2023) 207-218 (Q2)
Tahun	2023
Penulis	S. Vijayarani1 , C. Sivamathi2,* , P. Tamilarasi
Metode	*. SVM -. Akurasi 75% *. RIPPER -. 85% *. RF -. 89% *. WRFSVM -. 91%
Hasil	Berdasarkan hasil penelitan, menujukan bahwa pemodelan pada penggabungan algoritma yaitu WRFSVM memiliki akurasi yang tertinggi sebesar 91%.
Kesimpulan	Berdasarkan tingkat akurasi menujukan, pemodelan dengan penggabungan dua algoritma meningkatkan performa akurasi yang cukup tinggi, dibandingkan dengan algoritma tunggal saja.
Penelitian 10	
Judul	*A Comparative Analysis of Machine Learning Algorithms to Predict Liver Disease*[14]

Nama Jurnal	*Intelligent Automation & Soft Computing* (Q3)
Tahun	2021
Penulis	Mounita Ghosh1 , Md. Mohsin Sarker Raihan1 , M. Raihan2 , Laboni Akter1 , Anupam Kumar Bairagi3 , Sultan S. Alshamrani4 and Mehedi Masud
Metode	*. J48 -. Akurasi 68.77% *. SVM -. Akurasi 71.35% *. RF -. Akurasi 83.76%
Hasil	Berdasarkan pada hasil penelitian ini, menujukan pemodelan algoritma Random Forest memiliki akurasi yang tertinggi sebesar 83.76% dibandingkan dengan algoritma lainnya.
Kesimpulan	Berdasarkan tingkat akurasi algoritma Random Forest memiliki akurasi yang tertinggi, dengan didukung pada tingkat train size sebesar 80%, dan tingkat presentasi ROC sebesar 81%.

Berdasarkan tabel 2.1 Penelitian Terdahulu, terdapat artikel jurnal yang dijadikan referensi bagi penulis. Seperti, Artikel jurnal "*An Ensemble Learning Approach for Enhanced Classification of Patients With Hepatitis and Cirrhosis* yang ditulis oleh *Davide Chicco* dan *Guiseppe Jurman"* pada tahun 2021 akan dijadikan referensi untuk algoritma *Random Forest*[5]. Artike ini menjadi acuan untuk penggunaan algoritma *Random Forest,* dikarenakan tingkat performa dari akurasi algoritma *Random Forest* yang paling tinggi.

Berdasarkan tabel 2.1 Penelitian Terdahulu diatas, terdapat artikel jurnal yang dijadikan referensi bagi penulis. Seperti, artikel jurnal "*Klasifikasi Hepatitis C virus menggunakan Algoritma C4.5"* yang ditulis oleh Nuri Susanto pada tahun 2022 akan menjadi refrensi untuk algoritma *C4.5 AdaBoost*[6]. Artiken ini akan menjadi acuan dalam penggunaan algoritma *C4.5* dan Hibrida *C4.5-AdaBoost*, dikarenakan memberikan pemahaman dalam pengabungan dua algoritma untuk meningkatkan akurasi. Serta pada penelitian terdahulu, peneliti juga menjadikan artikel jurnal "*Hyperparameter Tuning* menggunakan *GridsearchCV* pada *Random Forest* untuk Deteksi *Malware*" sebagai refrensi dalam penggunakan Hyperparameter Tuning dengan tujuan untuk

meningkatkan peforma akurasi[11].

Penelitian ini berbeda dikarenakan membandingkan empat pemodelan yaitu *Random Forest*, *C.45*, *AdaBoost* dan hibrida *C4.5-AdaBoost*, dimana penggunaan algoritma *Random Forest* memiliki tingkat presentase yang tinggi, maka dari itu peneliti akan menggunakan algoritma *Random Forest* sebagai salah satu algoritma yang akan dibandingkan. Penggunaan algoritma *C4.5* memiliki akurasi yang cukup tinggi pada artikel jurnal terkait, namun jika algoritma *C4.5* dikombinasikan dengan *AdaBoost* akan meningkatkan akurasi yang cukup berbeda dibandingkan hanya menggunakan algoritma *C4.5* saja, maka dari itu penelitian akan menggunakan algoritma *C4.5- AdaBoost* sebagai perbandingan.

2.2 Tinjauan Teori

2.2.1 Hepatitis

Hepatitis merupakan salah satu masalah kesehatan didunia, yang mengakibatkan kematian pada kalangan usia. Virus Hepatitis disebabkan oleh kelompok virus yang beragam dengan struktur biologis yang beragam, transmisi, pola endemik, dan kronisitas yang memiliki kecenderungan yang sama untuk menginfeksi dan bereplikasi di hepatosit manusia[15]. Di seluruh dunia, penyakit ini disebut penyakit hati, penyakit hati atau hepatitis yang disebabkan oleh berbagai penyebab[16].

Hepatitis sering kali menyebabkan peradangan hati yang disebabkan oleh infeksi seperti virus, jamur, bakteri, parasit, atau penggunaan alkohol, obat-obatan, penyakit autoimun, atau infeksi metabolisme hati. Tingkat peradangan hati bisa menjadi kronis yang setidaknya selama 6 bulan[16]. Jenis yang paling umum disebabkan oleh infeksi virus adalah hepatitis a, b, c, d dan e. Setiap jenis berbeda tingkat keparahannya, namun jenis hepatitis manapun apabilan pasien sudah terpapar hepatitis dan tidak dilakukan penanganan lebih lanjut maka dikhawatirkan akan terjadi komplikasi khususnya pada hati. Ketika sudah terpapar, pihak rumah sakit akan

melakukan medical *chek up* pada darah dan juga hati, untuk mengetahui seberapa parah pasien tersebut[1].

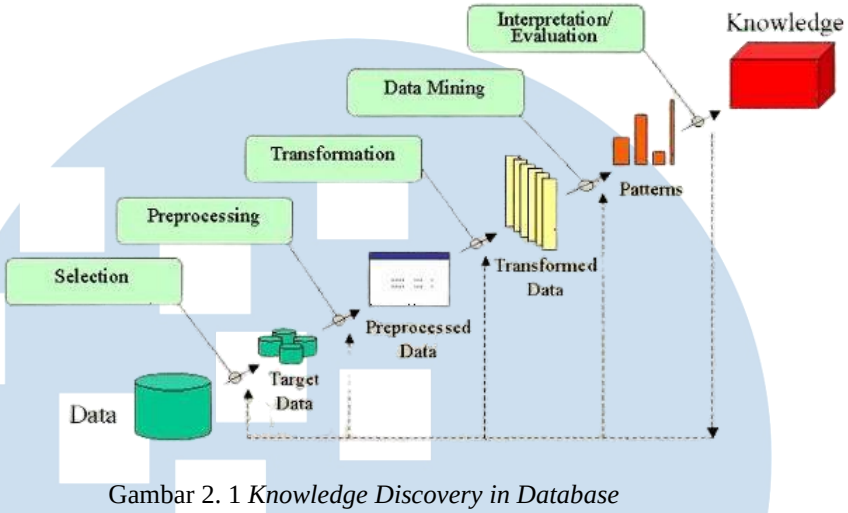
2.2.2 Rumah Sakit Umum Daerah Kota Tangerang (RSUD)

Rumah Sakit Umum Daerah atau yang disingkat RSUD merupakan satu-satu rumah sakit negeri pemerintah di pemerintahan kota tangerang. RSUD terletak di Jl. Pulau Putri Raya, Kelapa indah, Kec. Tangerang. Rumah Sakit Umum Daerah sudah berdiri sejak 2013 silam, dan hingga pada tahun 2019 covid-19 mulai meramba ke indonesia. Dalam menghadapi pandemi COVID-19, RSUD Kota Tangerang telah ditetapkan sebagai pusat penanganan dan rujukan untuk wilayah tersebut. RSUD ini telah meningkatkan kapasitas pelayanan dengan menambah ruang perawatan dan memberikan perlindungan kepada pegawai dengan menyediakan APD, akomodasi untuk petugas medis, skrining PCR berkala, serta suplemen makanan dan vitamin[17].

2.3 Framework dan Algoritma

2.3.1 Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database (KDD) adalah sebuah metode untukmendapat informasi atau *knowledge* dari database yang bertujuan untuk mengambil keputusan. Proses mengungkap informasi tersembunyi dalam database yang cukup besar disebut sebagai knowledge discovery in databases (KDD) atau data mining[18].



Gambar 2. 1 *Knowledge Discovery in Database*

Pada Gambar 2.1 *Knowledge Discovery in Database* hadir dengan beberapa langkah umum, langkah-langkah tersebut seperti: selection, preprocessing, tranformation, data mining, evaluation.

a. Data Selection

Langkah pertama yaitu *data selection*, langkah pertama dalam *Knowledge Discovery in Database* yang bertujuan sebagai pemelihan data dari sekumpulan data pada dataset yang digunakanuntuk penelitian.

b. Data Pre-Processing / Cleaning

Langkah ini merupakan tahapan yang melibatkan persiapan data atau proses manipulasi dataset yang akan digunakan dalam model sehingga data menjadi kompatibel dan sesuai. Pembersihandata dilakukan dengan cara membuang data duplikat, memeriksa data yang tidak konsisten, dan memperbaiki kesalahan data, sepertidata kosong.

c. Transformation

Langkah selanjutnya atau langkah ketiga yaitu *transformation*, langkah ini merupakan proses transformasi data dan penggabungan data yang terpilih agar data tersebut sesuai dengan proses data mining untuk menjadi jenis atau pola informasiyang akan dicari dalam basis data.

d. Data Mining

Langkah keempat atau langkah yang paling penting adalah *data mining*

yang merupakan proses untuk mencari pola, trend, atau informasi menarik dalam suatu data dengan menggunakan teknik matematika, statistika, pembelajaran mesin atau kecerdasanbuatan.

e. Interpretation / Evaluation

Langkah terakhir adalah interpretasi dan evaluasi. Langkah ini didasarkan pada hasil data mining, dan hasilnya disajikan dalam format yang mudah dipahami. Langkah ini juga mencakup penilaian cepat apakah proses data mining menghasilkan hasil yang konsisten atau tidak dengan model.

2.3.2 Algoritma *Random Forest*

Random Forest adalah sebuah metode klasifikasi dan regresi pohon keputusan. Dalam model hutan acak, setiap pohon adalah pohon klasifikasi dan regresi (CART) dan menggunakan pengurangan pengotor *Gini* dalam memilih prediktor pemisah dari subset yang dipilih secara acak dari semua variabel prediktor yang tersedia. Selain itu, setiap pohon hanya mengembalikan data sampel *bootstrap* daripada keseluruhan data asli. Penentuan kelas diambil berdasarkan mayoritas hasil vote dari semua pohon yang terbentuk[19].

Random Forest menampil lebih baik dari pfada model individual lainnya karena *Random Forest* menggunakan pohon keputusan yang tidak memiliki korelasi. Kesalahan prediksi pada satu pohon keputusan dapat ditutupi oleh kebenaran yang diperoleh dari pohon distribusi lainnyaselama arah pohon keputusan tersebut benar[20].

2.3.3 Algoritma C4.5

Algoritma *C4.5* adalah algoritma yang digunakan untuk membentuk pohon keputusan (*Decision Tree*) yang memilik bidang penambangan data yang berkembang, tetapi telah lama digunakan di berbagai bidang. Algoritma *C4.5* ditujukan untuk membuat sebuah pohon keputusan yang mudah untuk dipahami, dan menarik karena divisualisasikan kedalam bentuk gambar[21].

Pada awalnya, algoritma *C4.5* digunakan untuk mengubah data menjadi pohon keputusan dan aturan keputusan yang cocok untuk masalah klasifikasi prediktif dan penambangan data. Peta algoritma *C4.5* menetapkan nilai ke kelas yang dapat diterapkan pada klasifikasi baru[21]. Algoritma *C4.5* mengambil variabel yang terpilih dengan cara menghitung nilai *Gain* pada masing-masing variabel, nilai atribut tertinggi akan menjadi akar pertama pada pohon keputusan. Sebelum dilakukannya perhitungan nilai *Gain*, harus dilakukan nilai *Entropy*nya terlebih dahulu, rumus *entropy* terdapat pada *rumus 2.1*, setelah dilakukannya menghitung nilai *entropy* maka barulah menghitung nilai gain. Rumus *Gain* seperti pada *Rumus 2.2 Rumus Gain* [22].

$$\text{Entropy}(S)=\sum_{i}=1\text{-}p_i*\log_2(p_i)$$

Rumus 2. 1 Rumus Entropy

Di mana:

- S adalah himpunan data

- *c* adalah jumlah kelas dalam himpunan data

- *pi* adalah proporsi dari himpunan data yang termasuk dalam kelas ke-(*i*)

$$\text{Gain}(A)=\text{Entropy}(S)-\sum_{v\in \text{Values}(A)}|S||Sv|\times\text{Entropy}(Sv)$$

Rumus 2. 2 Rumus Gain

Di mana:

- Gain(A) adalah gain informasi dari atribut A

- Entropy (*S*) adalah entropi dari himpunan data S, yang mengGambarkan tingkat didalam *S.*

- Values(*A*) adalah himpunan nilai yang mungkin untuk atribut *A.*

- *Sv* adalah subset dari *S,* dimana atribut *A* memiliki nilai v.

- |S| adalah jumlah total instance dalam himpunan data *S.*

- |Sv| adalah jumlah instance dalam subset *Sv*.

- *c* adalah jumlah kelas dalam himpunan data

- *pi* adalah proporsi dari himpunan data yang termasuk dalam kelas ke-(*i*)

2.3.4 AdaBoost

AdaBoost adalah pendekatan pembelajaran mesin untuk meningkatkan akurasi aturan prediksi dengan mengkombinasikan banyak aturan yang relatif lemah dan tidak tepat. *Adaptive Boosting (AdaBoost)* adalah salah satu dari beberapa variasi algoritma untuk meningkatkan akurasi[20]. Algoritma *AdaBoost* merupakan salah satu algortima yang paling sering digunakan dan dipelajari dibanyak bidang. *AdaBoost* dapat dikombinasikan dengan algoritma klasifikasi lainnya meningkatkan kinerja klasifikasi. Tentu saja, menggabungkan beberapa model secara intuitif jika model tersebut berbeda satu sama lain[20].

Salah satu keunggulan *AdaBoost* adalah kemampuannya dalam menangani overfitting. *AdaBoost* mampu mengatasi *overfitting* dengan memperbaiki *weak learner* melalui pemberian bobot yang berbeda pada setiap data, sehingga model lebih fokus pada data yang sulit diprediksi dan tidak terlalu tergantung pada data yang mudah diprediksi. Namun, *AdaBoost* juga memiliki kelemahan, seperti sensitivitas terhadap noise dan ketidakseimbangan data. *AdaBoost* cenderung sensitif terhadap data yang

tidak seimbang, di mana kelas minoritas memiliki jumlah data yang sedikit. Hal ini dapat mengakibatkan model *AdaBoost* menghasilkan prediksi yang kurang akurat untuk kelas minoritas[23].

2.3.5 *KNNImputer*

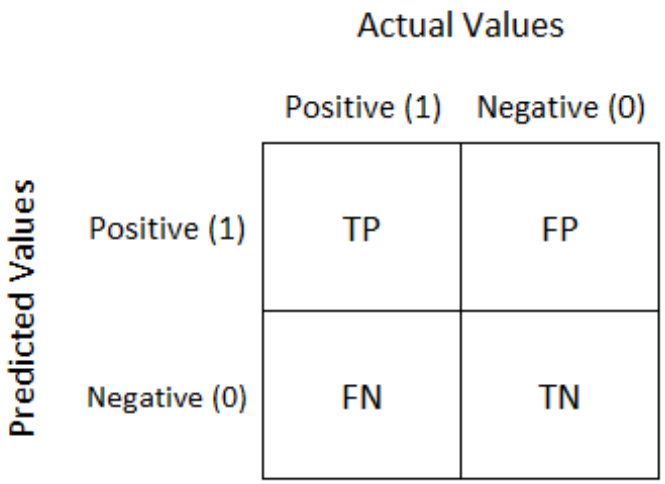
KNNImputer adalah salah satu teknik imputasi yang digunakan dalam pengolahan data untuk menangani nilai-nilai yang hilang (*missing values*) dalam dataset. Teknik ini adalah bagian dari machine learning dan khususnya sering digunakan dalam preprocessing data. *KNNImputer* termasuk dalam pustaka scikit-learn, yang merupakan pustaka machine learning populer di Python. Tujuan utama *KNNImputer* adalah untuk memperbaiki dataset yang memiliki nilai-nilai yang hilang dengan cara yang cerdas dan berdasarkan informasi yang ada dalam data[24]. Penanganan data hilang dengan *KNNImputer* dimulai dengan menentukan sejumlah tetangga terdekat atau observasi terdekat yang disimbolkan dengan k, kemudian menghitung jarak terkecil dari setiap observasi yang tidak mengandung data hilang[25].

2.3.6 *Hyperparameter*

Hyperparameter adalah parameter yang digunakan untuk mengontrol proses pembelajaran pada model machine learning. Algoritme pembelajaran mesin secara otomatis mempelajari dan menyesuaikan parameter internal mereka berdasarkan data yang diberikan. Parameter yang disesuaikan ini dikenal sebagai "parameter model", sementara hyperparameter mempengaruhi struktur atau perilaku model itu sendiri. Performa model machine learning bisa sangat bervariasi tergantung pada pemilihan dan nilai hyperparameternya. Sebagai contoh, dalam algoritma decision tree, kita memiliki hyperparameter "tree_depth"; menetapkan nilai yang moderat untuk hyperparameter ini dapat menghasilkan performa yang baik, sementara nilai yang terlalu tinggi dapat mengakibatkan overfitting atau penurunan kinerja model[26].

2.3.7 *Confusion Matrix*

Confusion Matrix merupakan sebuah metode untuk mengevaluasi yang digunakan untuk melakukan perhitungan akuras. Struktur kuadrat dari matriks konfusi diwakili oleh baris dan kolom. *Confusion Matrix* memiliki 4 kombinasi berbeda dari nilai prediksi dan nilai aktual[27].



Gambar 2. 2 Confusion Matrix

Pada Gambar 2.2 *Confusion Matrix* Empat kombonasi ini merupakan representasi hasil dari klasifikasi pada *Confusion Matrix* yaitu *True Positif, False Positif, True Negatif,* dan *False Negatif*. Metode untuk meringkas hasil dari confusion matrix meliputi: akurasi, presisi, f1-score dan recall[28].

2.3.7.1 Akurasi

Akurasi adalah salah satu metrik evaluasi yang paling umum digunakan dalam *machine learning*. Ini memberikan Gambaran tentang seberapa baik model dapat melakukan prediksi secara keseluruhan. Secara sederhana, akurasi mengukur seberapa sering model benar dalam memprediksi kelas dari data uji. Metrik ini dihitung dengan membagi jumlah prediksi yang benar oleh jumlah total prediksi yang dibuat[29]. Pada *Rumus 2.3 Rumus Akurasi*, dapat menyimpul kanhasil dari tingkat akurasi yang diberikan.

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

Rumus 2. 3 Rumus Akurasi

2.3.7.2 Presisi

Presisi adalah metrik yang sangat penting, terutama dalam kasus di mana kesalahan prediksi positif bisa memiliki konsekuensi yang mahal atau merugikan. Presisi memberikan informasi tentang tingkat ketelitian model dalam mengidentifikasi data positif. Semakin tinggi nilai presisi, semakin sedikit false positive yang dihasilkan oleh model, yang berarti model lebih jarang salah mengklasifikasikan data negatif sebagai positif[29]. Pada *Rumus 2.4 Rumus Presisi*, dapat menyimpulkan hasil dari tingkat presisi yang diberikan.

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

Rumus 2. 4 Rumus Presisi

2.3.7.3 Recall

Recall adalah metrik evaluasi dalam machine learning yang mengukur kemampuan model untuk menemukan kembali atau mengidentifikasi secara benar sebanyak mungkin *instance* positif dari kelas tertentu di antara semua *instance positif* yang sebenarnya. Dalam konteks klasifikasi, *recall* dihitung sebagai rasio antara *jumlah instance positif* yang diprediksi dengan benar (*true positive*) dibagi dengan total jumlah instance positif yang sebenarnya (*true positive* dan *false negative).* Semakin tinggi nilai recall, semakin baik model dalam mengidentifikasi *instance positif*[29]. Hal ini akan Pada *Rumus 2.5 Rumus Recall*, dapat menyimpulkan hasil dari tingkat recall yang diberikan.

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

Rumus 2. 5 Rumus Recall

2.3.7.4 F1-Score

F1-Score adalah metrik evaluasi yang menggabungkan presisi dan recall dalam satu angka. Ini membantu memberikan Gambaran yang lebih komprehensif tentang kinerja model klasifikasi, terutama dalam situasi di mana kelas target tidak seimbang. Dengan menggunakan harmonic mean dari presisi dan recall, F1-Score memberikan informasi tentang keseimbangan antara kedua metrik tersebut. Ini berguna ketika kita perlu memperhitungkan trade-off antara presisi dan recall yang signifikan dalam evaluasi model[29]. Pada *Rumus 2.6 Rumus F1-Score*, dapat menyimpulkan hasil dari tingkat f1-score yang diberikan.

$$F1\text{-}Score = \frac{2 \text{ X presisi x recall}}{\text{presisi + recall}}$$

Rumus 2. 6 Rumus F1-Score

2.4 Tools

2.4.1 *Google Colaboratory / Google Colab*

Google Colaboratory atau *Google Colab* adalah sebuah layanan yang menyerupai *Jupyter Notebook*, namun *google colab* berbasis *cloud*. *Google Colab* dapat dijalankan melalui browser mana pun seperti *opera*, *mozilla*, dan *google chrome*. *Google Colab* memberikan kemudahan bagi pemula untuk menjalankan kode *python* tanpa perlu menginstall library tertentu. Semua keperluan yang dibutuhkan seperti setting atau *adjustement* semuanya diserahkan ke *cloud*[30]. Berdasarkan aspek kemudahan *software* ini merupakan software terbaik untuk para *programmer* yang ingin mengasah pengetahuan mengenai bahasa pemograman *python*.

2.4.2 Python

Python adalah bahasa pemrograman tingkat tinggi yang dibuat oleh *Guido van Rossum*, bahasa pemrograman Python juga menawarkan pemrograman berorientasi objek (OOP) dan juga menyediakan antarmuka lintas platform. Kemampuan *Python* diperluas dengan kumpulan paket yang berada di perpustakaan *Python*, yang berisi lebih dari 100.000 paket dengan berbagai fungsi. Paket yang disediakan oleh *Python* memberi pengembang lebih banyak fleksibilitas dan kenyamanan saat membuat model[31].