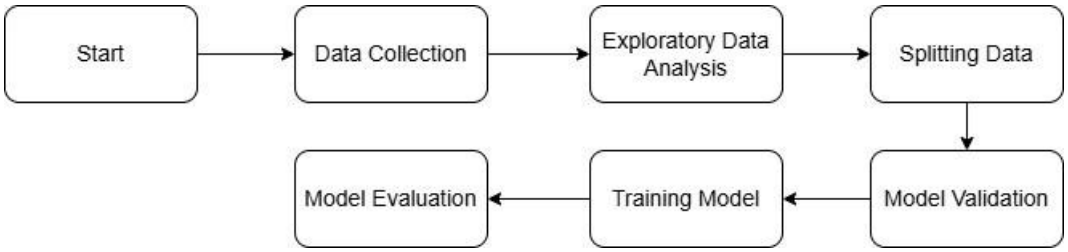
BAB III

METODOLOGI PENELITIAN

3.1 Alur Penelitian

Tahapan yang dilakukan untuk memprediksi dan mengevaluasi tingkat akurasi dari algoritma KNN (K Nearest Neighbor) dengan *machine learning* adalah sebagai berikut:



Gambar 3. 1 Alur Penelitian

3.2 Data Collection

Data collection merupakan proses pengumpulan dan pengukuran informasi tentang variabel yang ditargetkan dengan cara sistematis. Pengumpulan data yang akurat sangat penting dalam menjaga integritas penelitian.

Pengumpulan data adalah tahapan dasar dalam riset yang melibatkan proses sistematis untuk mengumpulkan dan mengukur informasi mengenai variabel yang diteliti. Pemilihan metode dan alat yang tepat, identifikasi sumber data yang relevan, serta penerapan prosedur yang terstruktur menjadi kunci dalam menghasilkan data yang akurat dan lengkap. Keakuratan data ini penting untuk menjaga validitas dan reliabilitas hasil penelitian, karena data yang tidak tepat dapat menyebabkan kesimpulan yang salah dan merusak integritas ilmiah dari keseluruhan studi.

Dalam proses KNN (K-Nearest Neighbor) tentu membutuhkan data yang cukup banyak dan sesuai dengan yang dibutuhkan, di dalam penelitian ini peneliti menggunakan data set dari *repository kaggle.* Terdapat sebanyak 1.429 dataset yang tersedia secara keseluruhan untuk dianalisis, namun data tersebut masih perlu

melalui proses pembersihan *(data cleaning)* sebelum dapat digunakan. Data tersebut memuat 23 atribut yaitu *survey_id* (responden), *ville_id* (identifier unik), *sex* (jenis kelamin), *age* (usia), *married* (status pernikahan), *number_children* (jumlah anak), *education_level* (tingkat pendidikan), *total_members* (jumlah anggota), *gained_asset* (aset yang diperoleh), *durable_asset* (aset tahan lama), *save_asset* (aset simpanan), *living_expenses* (biaya hidup), *other_expenses* (biaya lain), *incoming_salary* (gaji masuk), *incoming_own_fam* (pendapatan keluarga), *incoming_business* (pendapatan usaha), *incoming_agriculture* (pendapatan pertanian), *fam_expenses* (biaya keluarga), *labor_primary* (tenaga kerja utama), *lasting_investme* (investasi jangka panjang), *depressed* (depresi). Data ini akan digunakan untuk membangun model prediksi guna mengidentifikasi kemungkinan terjadinya depresi. Berikut data dalam format csv:

Survey_id	Ville_id	sex	Age	Married	Number_children	education_level	total_members	gained_asset	durable_asset
926	91	1	28	1	4	10	5	28912201	22861940
747	57	1	23	1	3	8	5	28912201	22861940
1190	115	1	22	1	3	9	5	28912201	22861940
1065	97	1	27	1	2	10	4	52667108	19698904
806	42	0	59	0	4	10	6	82606287	17352654
483	25	1	35	1	6	10	8	35937466	736707
849	130	0	34	0	1	9	3	41303144	21925041
1386	72	1	21	1	2	10	4	12013633	20323505
930	195	1	32	1	7	9	9	11087568	25224208
390	33	1	29	1	4	10	5	28912201	22861940
540	52	1	84	0	0	1	5	28912201	22861940
557	93	1	59	0	2	9	3	1018915	47245342
1280	232	1	38	1	4	10	6	12390944	19186414
1195	92	1	27	1	4	10	6	16521259	37155658
603	100	1	56	1	0	12	2	93596368	21140288
729	54	1	24	1	2	10	5	1108353	12219727
770	102	1	25	1	3	10	5	37172832	75432396
76	15	1	44	1	5	12	5	28912201	22861940
1374	267	1	32	1	4	9	5	28912201	22861940
379	22	1	26	1	2	7	4	82606287	20419597

Gambar 3. 2 *Sample Data Depression*

3.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) adalah pendekatan dalam analisis data yang bertujuan untuk memahami karakteristik utama dari dataset. *Exploratory data analysis* sering kali menggunakan metode visualisasi data dan teknik statistik untuk mengungkap pola, anomali, dan hubungan antar variabel. *Exploratory data analysis* data bertujuan untuk membantu dalam memahami data dan pola yang ada dalam

data. Selain itu, *exploratory data analysis* juga bertujuan untuk mengetahui anomali data sehingga dapat dilakukan tindakan selanjutnya yang kemudian dapat meningkatkan kualitas data yang ada. Dalam mengerjakan *exploratory data analysis*, biasanya data akan melalui tahap *prepocessing*. *Data preprocessing* adalah teknik untuk menyiapkan data agar lebih siap untuk dilakukan lebih lanjut dalam rangka ekstraksi pengetahuan. Selain itu dalam proses ini data akan diubah dalam bentuk yang akan lebih dipahami oleh sistem. Pada penelitian ini dilakukan *preprocessing* data berupa:

a. Data Cleaning

   Data cleaning, yaitu menghapus atau menghilangkan data yang tidak lengkap, tidak relevan dan anomali. Bisa juga dengan mengganti *record* yang hilang dengan nilai rata-rata atau yang lainnya. Tahapan pemrosesan untuk mengisi data kosong atau blank apabila memungkinkan, duplikasi data, memperbaiki data yang tidak sesuai dengan ketentuan atau salah ketik seperti kurang huruf dan kelebihan huruf, mengubah dan memodifikasi data agar data yang akan diolah adalah data yang konsisten, mengatur data yang kurang rapi dalam penulisan huruf besar dan kecil, dan mengganti format penulisan angka dan huruf sesuai dengan yang dibutuhkan. Dalam penelitian ini dataset yang dianalisis berjumlah 1.429, namun terdapat 20 *missing value* yang harus dihapus. Setelah dilakukan pembersihan data dan memastikan tidak ada duplikasi, jumlah dataset yang bersih dan siap untuk dianalisis adalah 1.409.

b. Data Transformation

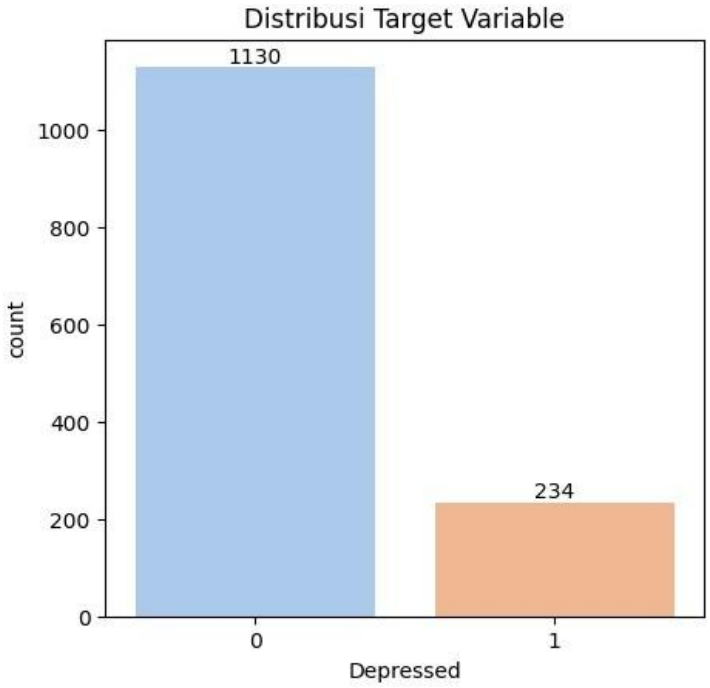
   Data transformation, yaitu mengubah data mentah menjadi bentuk yang lebih sesuai untuk dianalisis dan dimodelkan. Proses transformasi adalah tahap untuk mengubah data atribut yang selain angka ke dalam nilai angka agar data tersebut dapat diolah menggunakan algoritma KNN (K-Nearest Neighbor). Pada analisa data *depression* di Indonesia, tidak dilakukan proses transformasi data karenakan data tersebut sudah dalam bentuik *numerical* secara keseluruhan. *Numerical* seringkali juga disebut dengan *encoding* sehingga dapat dipahami oleh algoritma *machine learning.*

Data cleaning berfokus pada perbaikan kualitas data dengan menangani nilai kosong, anomali (outlier), dan ketidaksesuaian, yang dapat merusak perhitungan jarak dan penentuan tetangga terdekat, sehingga menghasilkan prediksi yang kurang akurat atau bias. Di sisi lain, data transformation melibatkan penyesuaian skala fitur (seperti normalisasi atau standardisasi) untuk menghindari dominasi fitur dengan rentang nilai besar dalam perhitungan jarak, serta transformasi lain untuk mengatasi distribusi data yang tidak normal atau meningkatkan representasi fitur yang penting. Dengan data yang telah diproses secara menyeluruh melalui data cleaning dan data transformation, KNN dapat lebih efektif mengenali pola kedekatan antar data dan memberikan hasil prediksi yang lebih baik.

3.4 Distribusi Data

Distribusi data adalah gambaran lengkap tentang bagaimana nilai-nilai dalam suatu kumpulan data tersebar atau bervariasi. Distribusi ini menunjukkan nilai-nilai apa saja yang ada dalam data, seberapa sering masing-masing nilai tersebut muncul, dan bagaimana nilai-nilai tersebut terkelompok atau terpencar.

Dalam konteks analisis statistik dan pembelajaran mesin, distribusi data merujuk pada pola penyebaran nilai-nilai dalam suatu dataset. Pemahaman mengenai distribusi data memberikan wawasan krusial tentang karakteristik bawaan dari data tersebut. Lebih dari sekadar daftar angka, distribusi data mengungkapkan frekuensi kemunculan setiap nilai atau rentang nilai, mengindikasikan nilai mana yang paling umum, nilai mana yang jarang, dan apakah nilai-nilai tersebut cenderung mengumpul di sekitar nilai tertentu atau tersebar luas. Visualisasi distribusi data, seperti melalui histogram, diagram batang, atau boxplot, memungkinkan kita untuk mengidentifikasi bentuk distribusi (misalnya, normal, skewed, bimodal), mengamati adanya nilai ekstrem (outlier), dan memahami variabilitas dalam data. Informasi ini sangat berharga dalam berbagai tahapan analisis, mulai dari pemilihan metode statistik yang tepat, penentuan asumsi model, hingga interpretasi hasil. Sebagai contoh, model pembelajaran mesin tertentu mungkin bekerja lebih baik dengan distribusi data yang mendekati normal, sehingga pemahaman distribusi membantu dalam pengambilan keputusan terkait transformasi data yang diperlukan.



Gambar 3. 3 Distribusi Data

Dari gambar ini, dapat dilihat bahwa data tidak seimbang. Ada lebih banyak data yang diprediksi *Not Depressed* (1130) dibandingkan data yang *Depressed* (234). Sumbu *x* menunjukkan kategori atau nilai dari variabel target, yaitu *"Depressed"*. Dalam hal ini, ada dua nilai: 0 (*Not Depressed)* dan 1 (*"Depressed")*. Sedangkan, sumbu Y menunjukkan jumlah atau frekuensi data untuk setiap kategori di sumbu X.

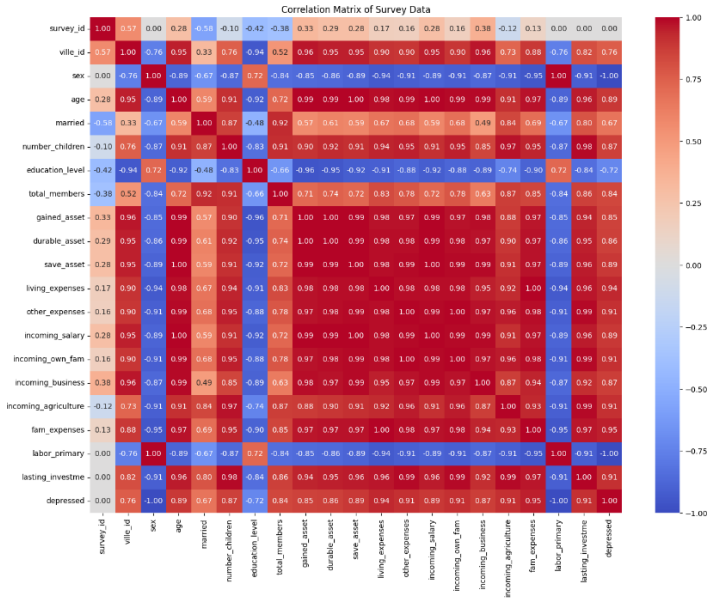
3.5 Heatmap Korelasi

Heatmap korelasi adalah alat visual untuk memahami hubungan antar berbagai faktor dalam data. Warna pada kotak-kotak heatmap menunjukkan seberapa kuat hubungan linear (hubungan yang berbentuk garis lurus) antara dua faktor.

Setiap sel dalam heatmap merepresentasikan korelasi antara dua variabel yang berbeda. Intensitas warna pada sel tersebut secara langsung menggambarkan kekuatan hubungan linear, warna yang lebih pekat menandakan korelasi yang lebih kuat, baik positif maupun negatif. Sementara itu, warna yang lebih pudar atau mendekati warna netral menunjukkan hubungan linear yang lemah atau bahkan tidak ada.

Selain intensitas warna, heatmap korelasi seringkali menyertakan nilai koefisien korelasi di dalam setiap sel, yang memberikan ukuran numerik yang tepat tentang kekuatan dan arah hubungan linear. Nilai koefisien korelasi berkisar antara -1 hingga +1. Nilai positif menunjukkan korelasi positif (ketika satu variabel meningkat, variabel lain cenderung meningkat), nilai negatif menunjukkan korelasi negatif (ketika satu variabel meningkat, variabel lain cenderung menurun), dan nilai mendekati nol menunjukkan korelasi linear yang lemah atau tidak ada.

Heatmap korelasi sangat berguna dalam analisis data eksploratif karena memungkinkan identifikasi variabel-variabel mana yang saling terkait kuat, yang mungkin mengindikasikan adanya dependensi atau potensi multikolinearitas dalam model. Informasi ini penting untuk berbagai tujuan, seperti pemilihan fitur yang relevan untuk pemodelan, pemahaman struktur data, dan identifikasi potensi hubungan sebab - akibat yang perlu diselidiki lebih lanjut. Dengan visualisasi yang intuitif, heatmap korelasi mempermudah pemahaman pola hubungan yang kompleks dalam dataset.



Gambar 3. 4 *Heatmap Korelasi*

Sumbu X dan Y menunjukkan fitur-fitur yang diukur, antara lain *sex* (jenis kelamin), *age* (usia), *married* (status pernikahan), *number_children* (jumlah anak), *education_level* (tingkat pendidikan), *total_members* (jumlah anggota keluarga),

gained_asset (aset yang diperoleh), *durable_asset* (aset tahan lama), *save_asset* (aset tabungan), *living_expenses* (biaya hidup), *other_expenses* (biaya lain-lain), *incoming_salary* (gaji masuk), *incoming_own_farm* (pendapatan dari pertanian sendiri), *incoming_business* (pendapatan dari bisnis), *incoming_no_business* (pendapatan bukan dari bisnis), *incoming_agricultural* (pendapatan dari pertanian), *farm_expenses* (biaya pertanian), *labor_primary* (tenaga kerja utama), *lasting_investment* (investasi jangka panjang), *no_lasting_investment* (tidak ada investasi jangka panjang), dan *depressed* (depresi). Warna merah tua menandakan hubungan positif yang kuat (ketika satu faktor naik, faktor lain juga cenderung naik), biru tua menunjukkan korelasi negatif yang kuat (ketika satu faktor naik, faktor lain cenderung turun), dan warna yang lebih muda menunjukkan hubungan yang lemah atau tidak ada korelasi linear yang signifikan.

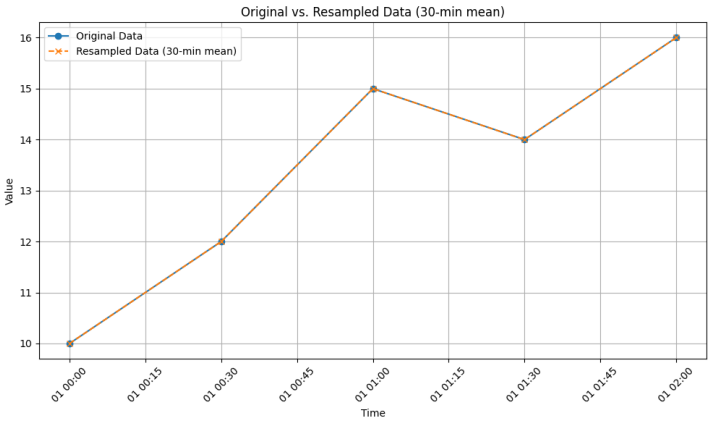
3.6 Resampled Data

Resampled adalah teknik yang digunakan untuk menangani ketidakseimbangan data, yaitu ketika jumlah sampel pada satu kelas jauh lebih banyak dibandingkan kelas lainnya. Teknik ini membantu menciptakan distribusi data yang lebih seimbang, sehingga model dapat belajar dengan lebih adil dari kedua kelas.

Penggunaan resampled data menjadi relevan terutama ketika dataset pelatihan mengalami ketidakseimbangan kelas yang signifikan, di mana satu kelas memiliki jumlah sampel yang jauh lebih sedikit dibandingkan kelas lainnya; ketidakseimbangan ini dapat menyebabkan KNN (K-Nearest Neighbor) cenderung bias terhadap kelas mayoritas, sehingga seringkali gagal mengklasifikasikan sampel dari kelas minoritas dengan benar. Teknik resampling, baik melalui oversampling (menambah sampel kelas minoritas) maupun undersampling (mengurangi sampel kelas mayoritas), bertujuan untuk menyeimbangkan distribusi kelas dalam data pelatihan yang digunakan untuk melatih model KNN (K-Nearest Neighbor). Dengan data yang telah di-resample, model KNN (K-Nearest Neighbor) dapat belajar dan mengklasifikasikan semua kelas dengan lebih adil, meskipun perlu diperhatikan potensi risiko overfitting atau hilangnya informasi yang mungkin timbul akibat proses resampling, sehingga pemilihan teknik dan

parameter resampling yang tepat serta validasi yang cermat menjadi krusial untuk memastikan peningkatan kinerja model secara keseluruhan.

Dalam kasus klasifikasi seperti prediksi depresi, jika data *"Not Depressed"* jauh lebih banyak daripada data *"Depressed,"* model cenderung belajar untuk hanya memprediksi kelas mayoritas karena dominasi jumlahnya, sehingga kelas minoritas diabaikan. Dengan *resampling,* masalah ini dapat diatasi dengan dua cara utama yaitu *oversampling* yang menambahkan data ke kelas minoritas (baik dengan menyalin data asli atau membuat data sintetis), dan *undersampling* yang mengurangi jumlah data di kelas mayoritas.



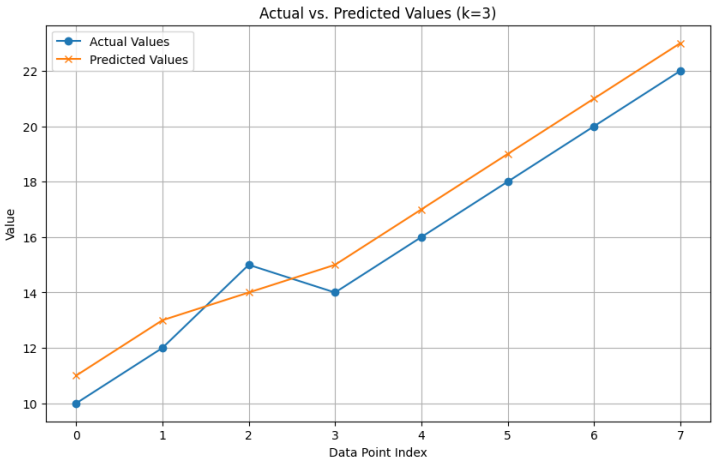
Gambar 3. 5 *Resampled Data*

Grafik ini membandingkan data asli dengan data yang di-*resampling*. Data yang diresampling, yang ditunjukkan oleh garis putus-putus berwarna oranye, merupakan representasi yang lebih halus dari data asli dengan mengambil nilai rata-rata setiap interval 30 menit. Terlihat bahwa garis resampled mengikuti tren umum data asli, namun menghilangkan fluktuasi detail yang terjadi dalam periode waktu yang lebih singkat. Setiap titik pada garis oranye mencerminkan nilai rata-rata data asli selama 30 menit sebelumnya. Sebagai contoh, nilai pada pukul 01:30 pada garis resampled adalah rata-rata dari nilai data asli antara pukul 01:00 dan 01:30. Proses resampling ini menghasilkan visualisasi tren yang lebih sederhana dan mengurangi noise dalam data.

3.7 Nilai K

Nilai K adalah jumlah tetangga terdekat yang dipertimbangkan untuk menentukan kelas atau nilai prediksi dari data uji. Pemilihan nilai K dalam

algoritma KNN (K-Nearest Neighbor) adalah langkah penting yang sangat memengaruhi hasil prediksi, karena K menentukan banyaknya tetangga terdekat yang diikutsertakan. Faktor-faktor seperti jumlah kelas pada klasifikasi, analisis tren kinerja model terhadap perubahan K, dan pemahaman konteks masalah dapat membantu dalam menentukan nilai K yang paling sesuai, dengan tujuan menyeimbangkan antara menangkap pola data yang penting dan meminimalkan pengaruh *noise* atau *outlier*. Dengan demikian, evaluasi yang seksama dalam memilih nilai K diperlukan untuk memastikan model KNN menghasilkan prediksi yang akurat dan terpercaya. Hasil penentuan nilai K secara langsung yang akan memengaruhi hasil prediksi, sehingga harus dilakukan dengan hati-hati.



Gambar 3. 6 Nilai K

Nilai K=3 dipilih karena memberikan proporsi yang baik antara terlalu umum dan terlalu spesifik. Dengan 3 tetangga terdekat, model KNN cukup fleksibel untuk menangkap pola data yang relevan, tetapi tidak terlalu terpengaruh oleh *noise* (data yang tidak diinginkan) atau outlier (nilai yang terlalu jauh).

3.8 Split Data

Split data adalah proses membagi dataset menjadi dua atau tiga bagian yang lebih kecil, di mana setiap bagian memiliki peranan pentingnya masing-masing. Rasio pembagian data antara data latih (training data) dan data uji (testing data) bisa disesuaikan tergantung pada kebutuhan analisis misalnya 70:30 atau 90:10, tergantung kompleksitas model, dan jumlah total data yang tersedia. Jika memiliki banyak data, menggunakan rasio 70:30 atau 80:20 adalah pilihan yang baik, karena

data uji cukup untuk mengukur performa. Namun, Jika data yang tersedia terbatas, rasio seperti 90:10 dapat membantu memberikan lebih banyak data ke model, meskipun evaluasi mungkin kurang detail karena jumlah data uji kecil. Rasio harus meyakinkan bahwa data uji tetap representatif terhadap keseluruhan data, sehingga evaluasi model mencerminkan performa sebenarnya. Disini dataset akan dibagi menjadi dua untuk digunakan dalam *training* dan *testing* dengan rasio 80:20.

Gambar 3. 7 *Split Data*

Gambar tersebut menunjukkan pembagian data menjadi dua bagian, yaitu data latih *(Train)* dan data uji *(Test),* dengan menggunakan rasio 80:20. Artinya, 80% dari total data digunakan untuk melatih model, sedangkan 20% sisanya digunakan untuk menguji kinerja model. Dapat dilihat bahwa jumlah data latih adalah 4 dan jumlah data uji adalah 1. Hal ini menunjukkan bahwa total data yang digunakan dalam penelitian ini adalah 5. Pembagian data ini penting untuk memastikan bahwa model yang dibangun tidak hanya baik dalam mempelajari data yang sudah ada, tetapi juga mampu memberikan prediksi yang baik pada data baru.

3.9 Training Model

Training model atau melatih model adalah kumpulan data yang digunakan untuk mengajarkan model mengenali pola dalam data sehingga model dapat belajar melakukan prediksi pada data baru, membangun, serta mengembangkan sebuah model *machine learning* yang dikenal sebagai *trained machine learning* model atau

model yang sudah dilatih. Algoritma KNN (K-Nearest Neighbor) bekerja dengan cara memasukkan data baru ke dalam kategori yang paling mirip dengan kategori yang sudah ada sebelumnya. [24]. Berbeda dengan model pembelajaran mesin pada umumnya yang membangun representasi internal melalui proses belajar, model KNN (K-Nearest Neighbor) dalam tahap pelatihannya lebih menekankan pada penyimpanan lengkap data latih. Saat melakukan prediksi untuk data baru, KNN tidak melakukan komputasi parameter. Melainkan, ia mengidentifikasi sejumlah K sampel terdekat dalam data latih berdasarkan kesamaan fitur, yang diukur dengan fungsi jarak. Prediksi kelas (untuk klasifikasi) atau nilai (untuk regresi) untuk data baru tersebut kemudian ditentukan oleh kelas atau nilai yang paling dominan di antara K tetangga terdekatnya. Dengan demikian, kualitas data latih yang tersimpan menjadi fondasi utama kinerja KNN. Pemilihan nilai K yang tepat juga krusial dan biasanya ditentukan melalui evaluasi terpisah, bukan sebagai bagian dari proses penyimpanan data latih itu sendiri.

Meskipun di era teknologi saat ini perhitungan algoritma banyak dilakukan menggunakan *machine learning*, pemahaman mendasar tentang cara kerja manual algoritma tetap penting. Secara manual, KNN bekerja dengan menentukan nilai K, menghitung jarak antara titik data baru dengan semua titik data lainnya (biasanya menggunakan jarak Euclidean), mencari K tetangga terdekat berdasarkan jarak tersebut, dan kemudian menentukan kelas atau nilai prediksi berdasarkan mayoritas kelas atau rata-rata nilai dari K tetangga terdekat tersebut. Perhitungan manual ini memberikan gambaran yang jelas tentang logika di balik algoritma KNN, meskipun dalam praktiknya, *machine learning* memungkinkan perhitungan yang jauh lebih efisien dan kompleks.

Secara garis besar, proses perhitungan manual KNN melibatkan dua langkah utama yaitu menghitung jarak antara data baru dengan semua data latih, dan kemudian menentukan kelas atau nilai prediksi berdasarkan *K* data latih terdekat. Berikut beberapa penjelasan terkait perhitungan manual menggunakan metode KNN (K-Nearest Neighbor).

- Jarak Euclidean *(Euclidean Distance)*

Euclidean distance berfungsi sebagai fondasi utama untuk mengukur kedekatan antara data baru yang hendak diprediksi dengan seluruh data latih yang ada melalui perhitungan jarak lurus antar titik dalam ruang fitur, KNN mengidentifikasi K data latih terdekat dengan data baru tersebut. Jarak Euclidean yang lebih kecil mengindikasikan kemiripan fitur yang lebih tinggi. Selanjutnya, prediksi kelas (untuk klasifikasi) atau nilai (untuk regresi) untuk data baru ditentukan berdasarkan kelas mayoritas atau nilai rata - rata dari K tetangga terdekat yang diidentifikasi menggunakan metrik jarak Euclidean ini, menjadikannya elemen krusial dalam mekanisme prediksi KNN.

- Jarak Manhattan *(Manhattan Distance)*

  Sebagai alternatif pengukuran kedekatan dalam algoritma KNN (K-Nearest Neighbor), Manhattan distance menghitung total selisih absolut nilai fitur antar dimensi, berbeda dengan Euclidean distance yang mengukur jarak langsung dalam proses prediksi KNN, *Manhattan distance* berperan dalam menentukan kedekatan data baru dengan data latih, dan sejumlah K data terdekat berdasarkan perhitungan ini dianggap sebagai tetangga. Hasil prediksi untuk data baru kemudian ditentukan oleh kelas atau nilai yang paling dominan di antara tetangga-tetangga tersebut. Penggunaan Manhattan distance dalam KNN dapat memberikan keuntungan dalam menangani *outlier* dan data dengan karakteristik dimensi yang lebih terpisah, meskipun pemilihan metrik ini seringkali didasarkan pada sifat unik dari data yang diolah.

- Jarak Minkowski *(Minkowski Distance)*

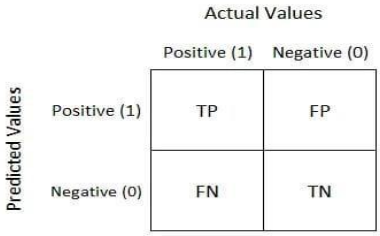
  Minkowski distance mencakup Euclidean dan Manhattan sebagai variasi spesifiknya secara matematis, ia menghitung jarak berdasarkan parameter p yang menentukan jenis perhitungan jarak antar fitur. Ketika p bernilai 2, Minkowski sama dengan Euclidean, dan ketika p bernilai 1, ia menjadi Manhattan distance. Oleh karena itu, dalam KNN, *Minkowski distance* memberikan fleksibilitas dalam memilih metrik jarak yang paling tepat untuk data yang dianalisis, dengan nilai p sebagai parameter tambahan yang

dapat disesuaikan untuk meningkatkan performa model berdasarkan karakteristik dataset.

Oleh karena itu, pemilihan metrik jarak seperti Euclidean, Manhattan, atau Minkowski dalam algoritma KNN merupakan pertimbangan penting yang didasarkan pada karakteristik spesifik dataset dan tujuan pemodelan, karena setiap metrik menangkap konsep kedekatan yang berbeda dan dapat memengaruhi pada kinerja akhir model.

3.10 Model Evaluation

Model evaluation adalah proses untuk mengukur seberapa baik sebuah model prediksi bekerja. Cara kerjanya dengan menguji model menggunakan data yang tidak digunakan selama pelatihan (data uji). Hasil prediksi model dibandingkan dengan label asli dari data uji, lalu dihitung metrik evaluasi seperti akurasi, precision, recall, atau F1-score untuk masalah klasifikasi. Metrik evaluasi digunakan untuk mengevaluasi seberapa baik kinerja model klasifikasi dalam memprediksi kelas yang benar. Tujuannya adalah untuk memahami seberapa akurat, presisi, dan lengkap prediksi model, serta mengidentifikasi area mana yang perlu ditingkatkan.



Gambar 3. 8 Metrik Evaluasi

Dalam evaluasi model klasifikasi, terdapat empat hasil utama: *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). TP menunjukkan model berhasil memprediksi positif dan hasilnya memang positif, sementara TN menunjukkan model berhasil memprediksi negatif dan hasilnya memang negatif. FP, atau kesalahan yang terjadi ketika model memprediksi positif padahal hasilnya negatif. Sebaliknya, FN, atau kesalahan yang terjadi ketika model memprediksi negatif padahal hasilnya positif. Keempat metrik ini penting untuk

menilai kinerja model klasifikasi secara komprehensif. rumus perhitungan accurary, precision, recall dan F1 score.

1. Accurary adalah metrik yang mengukur seberapa tepat model klasifikasi dalam memprediksi kelas dari keseluruhan data.

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

   Accurary, sebagai metrik evaluasi klasifikasi, mengukur proporsi prediksi yang tepat dibandingkan dengan keseluruhan prediksi yang dihasilkan model. Perhitungannya melibatkan pembagian jumlah prediksi yang benar (*true positives* dan *true negatives*) dengan total jumlah prediksi. Accurary dapat memberikan gambaran yang kurang akurat pada dataset dengan ketidakseimbangan kelas, di mana model yang cenderung memprediksi kelas mayoritas dapat menunjukkan nilai akurasi yang tinggi namun performa buruk pada kelas minoritas. Oleh karena itu, untuk evaluasi yang lebih komprehensif, terutama pada data yang distribusinya tidak merata.
2. Precision adalah metrik yang mengukur seberapa akurat prediksi positif dari sebuah model.

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

   Presisi mengukur seberapa tepat model memprediksi kelas positif. Ini adalah rasio antara prediksi positif yang benar dengan semua prediksi positif. Presisi penting jika kesalahan prediksi positif berdampak besar, misalnya dalam diagnosis penyakit untuk menghindari diagnosis palsu. Namun, presisi tinggi tidak berarti semua kasus positif terdeteksi, sehingga perlu dilihat bersama metrik lain seperti recall.
3. Recall mengukur kemampuan model untuk menemukan semua kasus positif yang sebenarnya ada dalam data.

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

Recall mengukur kemampuan model untuk menemukan semua kasus positif yang sebenarnya. Ini adalah rasio antara prediksi positif yang benar dengan total kasus positif yang sebenarnya ada. Recall penting jika melewatkan kasus positif *(false negative)* berdampak besar, misalnya dalam mendeteksi penyakit menular untuk menghindari kasus yang tidak terdiagnosis. Namun, recall tinggi tidak menjamin semua prediksi positif itu benar, sehingga perlu dilihat bersama metrik lain seperti presisi.

4. F1-Score adalah ukuran yang menggabungkan presisi dan recall dengan menghitung rata-rata harmonisnya. Metrik ini sangat berguna ketika data memiliki ketidakseimbangan kelas, karena memberikan gambaran yang lebih seimbang tentang kinerja model dibandingkan hanya menggunakan akurasi.

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

F1-score berguna untuk melihat seberapa baik model klasifikasi bekerja dengan menyeimbangkan ketepatan prediksi positif (presisi) dan kemampuan menemukan semua yang sebenarnya positif (recall). Ini dihitung menggunakan rumus yang memberikan nilai rendah jika salah satu dari presisi atau recall buruk. F1-score berguna terutama saat kita ingin model yang baik dalam memprediksi dengan benar dan tidak melewatkan kasus positif, terutama pada data yang tidak seimbang.

Disini diuraikan bahwa TP dan FN merujuk pada prediksi dan aktual kelas positif (kelas 1), dan TN dan FP merujuk pada prediksi dan aktual kelas negatif (kelas 0), maka dapat menyajikannya dalam bentuk confusion matrix berikut:

Tabel 3. 1 Metrik Evaluasi

	Prediksi Positif (1)	Prediksi Negatif (0)
Aktual Positif (1)	TP = 160	FN = 66
Aktual Negatif (0)	FP = 38	TN = 188

Tabel ini menunjukkan bagaimana kinerja model dalam memprediksi apakah sesuatu termasuk dalam kategori tertentu atau tidak. Dalam penelitian ini, model memprediksi apakah ada depresi atau tidak. Ada empat angka penting: TP (True Positive) adalah jumlah kasus depresi yang benar-benar diprediksi sebagai depresi oleh model (160 kasus). TN (True Negative) adalah jumlah kasus non-depresi yang benar diprediksi sebagai non-depresi (188 kasus). FP (False Positive) adalah jumlah kasus non-depresi yang salah diprediksi sebagai depresi (38 kasus). FN (False Negative) adalah jumlah kasus depresi yang salah diprediksi sebagai non-depresi (66 kasus). Dengan kata lain, model cukup baik dalam mengidentifikasi orang yang benar-benar depresi dan orang yang tidak depresi, tetapi ada beberapa kesalahan di mana model salah memprediksi beberapa orang sebagai depresi padahal tidak, dan sebaliknya.

Berikut tabel yang menyajikan hasil perhitungan metrik evaluasi untuk dataset *depression* dengan lebih detail.

Tabel 3. 2 Metrik Evaluasi

	Precision	*Recall*	*F1-Score*	*Support*
0	0.81	0.71	0.75	226
1	0.74	0.83	0.78	226
Accurary			0.77	452
Macro Avg	0.77	0.77	0.77	452
Weighted Avg	0.77	0.77	0.77	452

Tabel ini menyajikan metrik evaluasi untuk model klasifikasi yang telah dilatih dan diuji. Komponen tabel terdiri dari label 0 dan 1 yang menunjukkan dua kelas yang

diprediksi oleh model klasifikasi. Presisi mengukur seberapa banyak dari semua contoh yang diprediksi sebagai kelas tertentu yang benar-benar kelas tersebut. Recall mengukur seberapa banyak dari semua contoh yang sebenarnya kelas tertentu yang berhasil diprediksi oleh model. F1-score adalah rata-rata antara presisi dan recall. Support menunjukkan jumlah contoh yang sebenarnya termasuk dalam kelas tertentu. Untuk kategori 0, model memiliki ketepatan 81%, yang berarti dari semua prediksi yang diberi label 0, 81% di antaranya benar. Model mampu menemukan 71% dari semua data yang sebenarnya termasuk dalam kategori 0. Untuk kategori 1, model memiliki ketepatan 74%, dan mampu menemukan 83% dari semua data yang sebenarnya termasuk dalam kategori 1. Skor F1 untuk kategori 0 adalah 75%, dan untuk kategori 1 adalah 78%, yang menunjukkan keseimbangan antara ketepatan dan kelengkapan. Rata-rata makro dari F1-Score adalah 77%, yang merupakan rata-rata sederhana dari kedua kategori. Rata-rata tertimbang dari F1-Score juga 77%, yang mempertimbangkan jumlah sampel di setiap kategori. Setiap kategori memiliki 226 sampel, sehingga total sampel adalah 452. Model memiliki akurasi keseluruhan 77%, tetapi ada perbedaan kinerja antara kelas 0 dan 1. Model ini lebih baik dalam memprediksi kelas 1 tetapi sedikit kurang presisi dalam memprediksi kelas 1 dibandingkan kelas 0.