PENERAPAN ALGORITMA K-NEAREST NEIGHBOR PADA MINAT BELI MOBIL BEKAS MENGGUNAKAN PENDEKATAN CRISP-DM

Ridho Hidayatullah, Doni Abdul Fatah, Achmad Yasid
Sistem Informasi, Universitas Trunojoyo Madura
Jl. Raya Telang, Kecamatan Kamal, Bangkalan, Jawa Timur, Indonesia
Ridhoh32@gmail.com

ABSTRAK

Saat ini, perkembangan dunia usaha semakin berfokus pada pemenuhan kebutuhan konsumen, termasuk dalam sektor bisnis jual-beli mobil bekas. Fenomena ini menunjukkan bahwa banyak konsumen cenderung memilih mobil bekas karena menawarkan kualitas yang memadai dengan harga yang lebih murah dibandingkan mobil yang baru. Beragam kebutuhan masyarakat Indonesia, seperti generasi muda yang menginginkan mobil bergaya, keluarga yang membutuhkan kendaraan luas, hingga pengusaha yang memprioritaskan kapasitas angkut dan kenyamanan, dapat terpenuhi melalui ketersediaan mobil bekas yang mudah diakses di pasar. Penelitian ini bertujuan untuk menganalisis pola pembelian mobil bekas dengan menggunakan algoritma K-Nearest Neighbor (K-NN) pada nilai K=7. Hasil penelitian menunjukkan bahwa model yang dikembangkan memiliki tingkat akurasi yang cukup tinggi sebesar 93%. Nilai *precision* tercatat sebesar 96% untuk kelas 0 dan 88% untuk kelas 1, sedangkan nilai *recall* mencapai 94% pada kelas 0 dan 91% pada kelas 1. Selain itu, nilai *f1-score* tercatat sebesar 95% pada kelas 0 dan 89% pada kelas 1. Temuan ini membuktikan bahwa algoritma K-Nearest Neighbor memiliki performa yang sangat baik dalam mengklasifikasikan dataset terkait pembelian mobil bekas.

Kata kunci : *K-Nearest Neighbor, CRISP-DM, Klasifikasi, Machine Learning*

1. PENDAHULUAN

Perkembangan dunia usaha saat ini mulai berorientasi pada konsumennya, termasuk dalam bisnis jual-beli mobil bekas. Seiring waktu, banyak konsumen lebih memilih mobil bekas karena kualitasnya yang baik dan harganya yang lebih terjangkau, sehingga menjadi pilihan yang lebih disukai dibandingkan mobil baru [1].

Transportasi adalah sarana yang digunakan untuk memindahkan barang atau orang dalam jumlah tertentu ke lokasi tertentu dalam waktu yang telah ditentukan. Saat ini, kebutuhan akan alat transportasi telah menjadi kebutuhan primer. Banyak orang cenderung memilih menggunakan kendaraan pribadi, seperti sepeda motor atau mobil, dibandingkan dengan transportasi umum. Kendaraan pribadi, khususnya mobil, menawarkan kenyamanan dan kemewahan serta dapat digunakan untuk perjalanan baik dalam kota maupun luar kota. Seiring dengan perkembangan zaman, produsen mobil terus menghadirkan berbagai pilihan mobil terbaru. Promosi dan iklan mobil baru yang semakin intensif sering kali mendorong konsumen untuk menjual mobil lama mereka dan menggantinya dengan model terbaru. Hal ini menciptakan peluang untuk memperjualbelikan mobil bekas yang masih layak pakai kepada konsumen lainnya [2].

Usia dan gaji merupakan dua faktor penting yang sering dipertimbangkan dalam pengambilan keputusan untuk membeli suatu produk, termasuk mobil. Berdasarkan hal tersebut, penelitian ini mengusulkan penggunaan algoritma machine learning untuk memprediksi minat beli konsumen terhadap mobil dengan mempertimbangkan dua variabel utama, yaitu usia dan gaji. Proses prediksi dilakukan melalui klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN). Adapun rumusan masalah yang diajukan dalam penelitian ini adalah: "Bagaimana tingkat akurasi algoritma K-Nearest Neighbor dalam memprediksi minat beli konsumen terhadap mobil berdasarkan variabel usia dan gaji?" Selain itu, tren pasar juga menunjukkan bahwa banyaknya penjual mobil bekas yang menawarkan kendaraan dengan desain modern dan fitur tambahan menjadi daya tarik tersendiri bagi konsumen. Dalam persaingan bisnis yang semakin ketat, penjual mobil bekas perlu menerapkan strategi yang efektif, seperti memanfaatkan data penjualan harian. Dengan menganalisis data tersebut, penjual dapat memahami perilaku dan preferensi konsumen, mengikuti tren pasar, serta mengevaluasi kinerja produk perusahaan untuk meningkatkan penjualan secara lebih optimal. [3].

Berdasarkan penelitian terdahulu penggunaan algoritma K-NN dipakai untuk Penerapan Algoritma Klasifikasi K-Nearest Neighbor pada Penyakit Diabetes seseorang. Dimana pada hasil penelitian tersebut didapatkan akurasi sebesar 76,56% pada percobaan nilai K=11. Dimana penelitian ini berhasil diterapkan sesuai tujuan yang diharapkan sehingga nilai 11 merupakan nilai K yang dapat digunakan dalam penerapan algoritma K-NN pada penyakit diabetes [4].

Penelitian selanjutnya penggunaan algoritma K-NN untuk Mengklasifikasi Citra Belimbing Berdasarkan Fitur Warna dimana pada hasil penelitian tersebut didapatkan tingkat akurasi sebesar 93.33% pada percobaan nilai K=7, dimana penelitian ini

berhasil diterapkan sesuai tujuan yang diharapkan guna mengidentifikasi tingkat kematangan buah belimbing berdasarkan citra dengan algoritma K-NN [5].

Penelitian selanjutnya yaitu penggunaan algoritma K-NN pada Implementasi Algoritma K-NN Untuk Klasifikasi Seleksi Penerima Beasiswa dimana pada hasil penelitian ini ada 89 data yang digunakan dan terseleksi sekitar 30 orang. Hasil pengujian akurasi menggunakan *confusion matrix* didapatkan nilai sebesar 90.5%. Hal ini menunjukkan bahwa K-NN ini bisa digunakan untuk mengklasifikasikan seleksi penerimaan beasiswa [6].

Permasalahan yang telah diuraikan diatas dapat dianalisis melalui data penjualan sebelumnya untuk mengetahui jenis mobil yang paling banyak atau paling sedikit terjual berdasarkan estimasi penghasilan dan usia calon pembeli. Data tersebut dapat menjadi panduan bagi perusahaan dalam memahami minat pelanggan terhadap mobil bekas. Hal ini menunjukkan bahwa estimasi pendapatan dan usia merupakan faktor penting yang dipertimbangkan dalam keputusan pembelian suatu produk, seperti mobil bekas. Oleh karena itu, data ini dapat dijadikan dasar untuk mengembangkan program yang mampu mengklasifikasikan minat beli mobil bekas menggunakan algoritma K-NN berdasarkan kriteria tersebut.

2. TINJAUAN PUSTAKA

2.1. Machine Learning

Teknologi *machine learning* (ML) adalah sistem yang dirancang untuk dapat belajar secara mandiri tanpa membutuhkan arahan langsung dari penggunanya. Proses pembelajaran ini didasarkan pada berbagai disiplin ilmu, seperti statistika, matematika, dan data mining, yang memungkinkan mesin untuk menganalisis data tanpa perlu dilakukan pemrograman ulang atau perintah manual. Beberapa metode yang umum digunakan dalam pengklasifikasian menggunakan *machine learning* meliputi *Decision Trees* (DT), *Naive Bayes* (NB), *Random Forest*, *Rotation Forest* (RF), *K-Nearest Neighbor* (KNN), *Artificial Neural Networks* (ANN), *Support Vector Machine* (SVM), dan metode lainnya [7].

Pembelajaran mesin (*Machine Learning*) adalah studi yang terus berkembang dalam bidang kecerdasan buatan, yang fokus pada konsep pengenalan pola dan pembelajaran komputasi pada mesin. *Machine learning* menggunakan berbagai algoritma pembelajaran yang berfungsi untuk memprediksi dan membantu pengambilan keputusan secara otomatis berdasarkan data yang tersedia. Proses *machine learning* ini dibangun melalui kerangka kerja *Data Preparation* dan *Data Modeling*, yang memungkinkan sistem untuk dijalankan secara otomatis dan mandiri. Dengan menggunakan model pembelajaran mesin seperti ini, para pemangku kepentingan dapat melakukan analisis prediksi dari dataset yang dapat sangat membantu perusahaan dalam membuat keputusan yang lebih tepat [8].

2.2. Klasifikasi

Klasifikasi merupakan salah satu proses utama dalam data mining yang bertujuan untuk menemukan pola pembelajaran atau hubungan dalam data dan menentukan fitur atau label kelas dari sampel yang dikategorikan. Proses ini melibatkan pengelompokan data ke dalam kategori yang telah ditentukan sebelumnya berdasarkan karakteristik yang terdapat pada data tersebut. Klasifikasi termasuk dalam metode *supervised learning*, di mana model dibangun dengan menggunakan data yang sudah dilabeli untuk mempelajari pola-pola yang ada. Tujuan utamanya adalah untuk membedakan kelas label dari dataset dengan cara mencari model atau atribut yang dapat memprediksi kelas dari suatu objek dengan tingkat akurasi yang tinggi. Dengan demikian, klasifikasi memungkinkan untuk membuat prediksi yang tepat dan mengelompokkan data secara otomatis ke dalam kategori yang relevan[9].

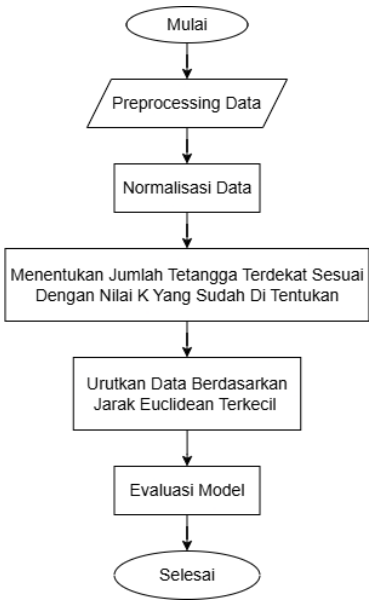
Klasifikasi adalah proses mengelompokkan atau memberi label pada data baru berdasarkan karakteristik tertentu. Studi ini menganalisis atribut dari data yang ada untuk memprediksi kelas data yang belum diketahui. Prosesnya meliputi tiga tahap utama: konstruksi model, aplikasi model, dan evaluasi. Pada tahap konstruksi, data latih yang sudah memiliki variabel dan kelas digunakan untuk menentukan kelas data baru. Selanjutnya, data yang telah diklasifikasikan dievaluasi untuk mengukur akurasi hasil dari pengembangan dan penerapan model. [10].

2.3. K-Nearest Neighbor

K-Nearest Neighbour (K-NN) adalah sebuah studi yang menggunakan algoritma *supervised learning*, di mana suatu *instance* baru diklasifikasikan dengan cara mencari sejumlah K tetangga terdekat dalam dataset yang sudah ada. Proses klasifikasi dilakukan dengan mempertimbangkan mayoritas kategori yang dimiliki oleh tetangga terdekat tersebut. Dengan kata lain, kategori dari instance baru ditentukan berdasarkan kategori yang paling sering muncul di antara K tetangga yang paling mirip atau terdekat secara jarak. Metode ini sering digunakan dalam berbagai aplikasi klasifikasi karena kesederhanaannya dan kemampuannya dalam mengatasi masalah klasifikasi dengan data yang tidak terstruktur [6].

K-nearest neighbor (KNN) merupakan algoritma yang digunakan untuk melakukan klasifikasi terhadap suatu data dengan memanfaatkan data pembelajaran (*train data set*) yang diambil dari sejumlah tetangga terdekatnya (*nearest neighbors*). Dalam algoritma ini, parameter k merepresentasikan jumlah tetangga terdekat yang diperhitungkan. Proses klasifikasi dilakukan dengan menganalisis mayoritas kedekatan jarak dari tetangga-tetangga tersebut, sehingga data baru dapat dikelompokkan ke dalam kategori yang

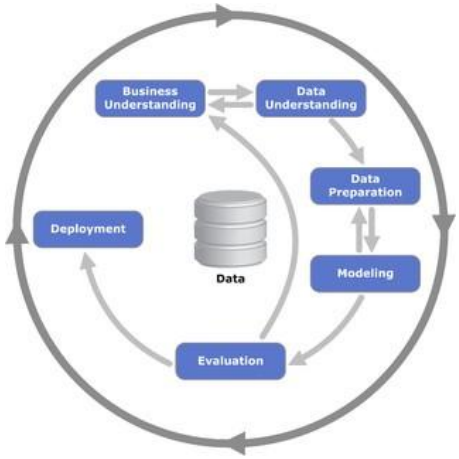
paling sesuai berdasarkan informasi dari tetangga terdekatnya. Algoritma ini sangat berguna dalam berbagai aplikasi karena kesederhanaan konsep dan kemampuannya untuk bekerja dengan baik pada dataset yang relatif kecil [11].



Gambar 1. Flowchart Metode K-NN

3. METODE PENELITIAN

3.1. CRISP-DM



Gambar 2. Metode CRISP-DM

CRISP-DM (*Cross Industry Standard Process for Data Mining*) merupakan suatu pendekatan standar dalam pemrosesan data mining yang dibangun untuk memastikan bahwa data melalui setiap tahap secara terstruktur, terdefinisi dengan jelas, dan efisien. Studi literatur ini terdiri dari enam tahap yang saling berulang, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, *Evaluation*, dan *Deployment*, seperti yang ditunjukkan pada Gambar 1. Tujuan utama dari studi CRISP-DM adalah untuk mengimplementasikan analisis yang sistematis dalam memecahkan masalah penelitian atau tantangan yang dihadapi oleh bisnis atau perusahaan [12].

3.2. Business Understanding

Pemahaman bisnis adalah tahap awal dalam proses penelitian di mana studi ini melakukan analisis mendalam terhadap topik yang diteliti, serta mengidentifikasi dan memahami masalah bisnis yang ada. Pada tahap ini, penelitian berfokus pada pencarian solusi yang tepat untuk mengatasi tantangan yang dihadapi oleh bisnis atau perusahaan. Selain itu, penelitian ini juga merancang rencana atau strategi yang terstruktur untuk mencapai tujuan penelitian, yang dapat mencakup penentuan langkah-langkah spesifik, pengumpulan data yang relevan, serta pemilihan metode yang sesuai. Tujuan utama dari tahap ini adalah untuk memastikan bahwa penelitian yang dilakukan relevan dengan kebutuhan bisnis dan dapat memberikan solusi yang efektif terhadap permasalahan yang ada [13].

3.3. Data Understanding

Tahap kedua adalah langkah pertama yang sangat krusial dalam proses penelitian ini, karena menentukan dasar bagi analisis lebih lanjut. Pada tahap ini, data dikumpulkan dengan hati-hati dari berbagai sumber yang relevan dan dipilih dengan seksama untuk memastikan kelengkapannya. Setelah data terkumpul, langkah selanjutnya adalah menganalisisnya secara mendalam guna mendapatkan pemahaman yang jelas mengenai pola, tren, dan karakteristik yang ada dalam data tersebut. Proses analisis ini penting untuk memastikan bahwa data yang diperoleh memiliki kualitas yang baik dan siap untuk digunakan. Setelah tahap ini, dataset kemudian diproses lebih lanjut pada tahapan selanjutnya untuk memastikan akurasi, keterandalan, dan relevansi informasi yang dapat diambil dari data tersebut [14].

Dataset yang digunakan adalah *Social Network Ads* yang didapatkan dari website Kaggle.com. Gambaran tentang dataset dapat dilihat pada tabel berikut :

Tabel 1. Fitur pada dataset

Atribut	Keterangan
User ID	User id customer
Gender	Jenis kelamin calon customer
Age	Usia customer
EstimatedSalary	Gaji calon customer
Purchased	Beli (1) / tidak beli (0)

3.4. Data Preparation

Pada fase ketiga, pemilihan data, pemberian label pada data atau fitur yang digunakan, selanjutnya dilakukan *preprocessing* yaitu :

3.5. Memperbaiki Missing Value

Tahapan *preprocessing* ini diantaranya memperbaiki *missing value*, membersihkan dataset dengan membuang data yang tidak digunakan seperti data kosong atau data duplikat. Pada tahap penelitian ini tidak memperbaiki *missing value* karena dataset tidak memiliki *missing value*.

3.6. Membuang Fitur Yang Tidak Digunakan

Pemilihan fitur *Age* dan *EstimatedSalary* untuk data *train* dilakukan karena kedua fitur ini dianggap paling relevan dengan target klasifikasi, seperti keputusan untuk membeli atau tidak membeli produk. Selain itu, menggunakan dua fitur utama mempermudah proses visualisasi hasil klasifikasi dalam grafik 2D, sehingga lebih mudah untuk dianalisis. Langkah ini juga membantu menyederhanakan model dan mengurangi risiko *overfitting*, karena terlalu banyak fitur dapat menyebabkan model menjadi terlalu kompleks dan kurang mampu melakukan generalisasi pada data uji. Dengan fokus pada fitur yang signifikan, proses klasifikasi menjadi lebih efisien dan akurat, tanpa gangguan dari fitur yang kurang berpengaruh atau redundan.

3.7. Split Fitur Menjadi Fungsi X (Age, EstimatedSalary) dan y (Purchased)

Tahap selanjutnya adalah merubah bentuk dataset menjadi fungsi X (*Age*, E*stimatedSalary*) dan y (*Purchased*) sebagai label. Proses pemisahan ini berguna untuk proses pelatihan *data train* dan *data test*.

3.8. Split Dataset Menjadi 4 Bagian

Tahap selanjutnya adalah memisah fungsi X dan y menjadi 4 bagian, yaitu X_train, X_test, y_train, dan y_test. Dimana X_train dan X_test a menjadi data *train set*, sedangkan y_train dan y_test menjadi data *test set*. Rumus untuk melakukan *split dataset* adalah sebagai berikut :

$$N_{test} = p_{test}.N \quad (1)$$
$$N_{train} = N - N_{test}$$

Keterangan :

a. N adalah jumlah total data
b. Ntrain adalah jumlah data train
c. Ntest adalah jumlah data test
d. Ptest adalah proporsi data yang dialokasikan untuk test

3.9. Normalisasi

Tahap keempat adalah melakukan proses pemodelan terhadap dataset yang telah diproses menjadi dataset final. Pada tahap ini, dataset dibagi menjadi dua bagian utama, yaitu data latih (training data) dan data uji (testing data). Data latih adalah data yang digunakan untuk melatih model algoritma klasifikasi, sehingga model dapat memahami pola dan karakteristik dari data yang diberikan. Proses ini bertujuan untuk membangun model yang mampu melakukan prediksi dengan tingkat akurasi yang optimal. Sementara itu, data uji digunakan untuk menguji performa model yang telah dilatih. Data uji ini terdiri dari data yang tidak pernah digunakan selama proses pelatihan, sehingga hasil evaluasi dari data uji memberikan gambaran yang objektif mengenai kemampuan model dalam melakukan prediksi terhadap data baru. Pembagian dataset ini sangat penting untuk memastikan bahwa model tidak hanya bekerja baik pada data yang telah dikenal (overfitting), tetapi juga memiliki generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya.. Pada tahap ini dilakukan *scaling feature* pada dataset yang telah di *preprocessing* dengan *fit transform* yang merubah bentuk dataset menjadi ternormalisasi menggunakan *MinMaxScaler* yang mana untuk menstandarisasikan fitur dengan merubah skala nilai dari dataset kedalam skala diantara 0–1 atau desimal. Rumus untuk melakukan normalisasi *MinMaxScaler* adalah sebagai berikut :

$$X = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

Keterangan :

a. X adalah data yang di pilih,
b. Xmin adalah data terkecil,
c. Xmax data terbesar

3.10. Menentukan Nilai K dan Metric Optimal

Tahap selanjutnya adalah melakukan proses klasifikasi menggunakan algoritma K-Nearest Neighbors (K-NN) dengan *hyperparameter tuning* menggunakan *GridSearchCV* untuk mencari parameter terbaik. Pertama, algoritma *KNeighborsClassifier* diinisialisasi dengan parameter default. Lalu, *GridSearchCV* digunakan untuk mencoba berbagai kombinasi parameter, seperti jumlah tetangga terdekat (*n_neighbors*), jenis metrik jarak (*metric*), dan nilai parameter pangkat (*p*) untuk metrik *Minkowski*. Parameter-parameter ini didefinisikan dalam *dictionary param_grid*. *GridSearchCV* mengevaluasi performa setiap kombinasi parameter menggunakan teknik *cross-validation* (cv=5) dengan metrik evaluasi berupa akurasi (*scoring='accuracy'*). Akhirnya, model dilatih pada data pelatihan (X_train, y_train) untuk menentukan kombinasi parameter terbaik berdasarkan hasil *tuning* tersebut. Berikut adalah rumus yang digunakan pada tahap ini :

a. Rumus Hyperparameter Tuning

$$\theta^* = argmax_{\theta \in param\_grid}\, Accuracy_{avg}(\theta) \quad (3)$$

Keterangan :

θ∗: Kombinasi parameter terbaik.
θ: Kombinasi parameter tertentu dalam ruang pencarian (param_grid).
Accuracyavg(θ): Akurasi rata-rata untuk kombinasi parameter θ

b. Rumus K-NN dengan Matrix Euclidean

$$d_i = \sqrt{\sum_{i=1}^{p} (x_1 - x_2)^2} \quad (4)$$

Keterangan :

d = jarak
I = variable data
P = dimensi data
x_1 = sampel data
x_2 = data uji

3.11. Train Model

Tahap berikutnya dalam proses ini adalah melatih model menggunakan algoritma K-NN (K-Nearest Neighbors). Pada fase ini, algoritma K-NN diterapkan dengan memanfaatkan matriks *Euclidean* untuk menghitung jarak antar data. Selain itu, nilai K yang mengindikasikan jumlah tetangga terdekat yang dipertimbangkan dalam proses klasifikasi, telah ditentukan sebelumnya. Proses pelatihan ini bertujuan untuk memungkinkan model mengenali pola-pola dalam data yang ada dan mempersiapkannya untuk melakukan prediksi dengan akurasi yang optimal pada data baru.

3.12. Evaluasi Performa Confusion Matrix

Fungsi confusion_matrix dari pustaka sklearn.metrics digunakan untuk menghitung *confusion matrix*, yang menggambarkan perbandingan antara nilai prediksi (y_pred) dan nilai yang sesungguhnya (y_test). *Confusion matrix* terdiri dari empat elemen utama yaitu *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), *False Negatives* (FN). Fungsi ini mengembalikan sebuah matriks yang menyatakan jumlah dari setiap kategori di atas, yang membantu dalam mengevaluasi kinerja model klasifikasi. Berikut adalah rumus *Confusion Matrix* yang digunakan :

$$confusion\ matrix = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (5)$$

3.13. Evaluasi Performa Akurasi Skor

Tahap selanjutnya adalah mengevaluasi kinerja model klasifikasi dengan fungsi accuracy_score(y_test, y_pred) menghitung akurasi model, yang merupakan persentase prediksi yang benar dari total data uji. Akurasi ini dicetak dalam bentuk persentase, memberikan gambaran umum seberapa baik model dalam melakukan prediksi secara keseluruhan. Berikut adalah rumus matematika yang digunakan :

$$Akurasi = \frac{jumlah\ prediksi\ benar}{jumlah\ total\ data}\ x\ 100 \quad (6)$$

Keterangan :

a. jumlah prediksi yang benar adalah jumlah prediksi yang sesuai dengan nilai sebenarnya.
b. Jumlah total data adalah jumlah keseluruhan data uji.

3.14. Evaluasi Performa Skor Presisi

Pengukuran seberapa akurat prediksi oleh model untuk kelas tertentu.

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

a. Evaluasi Performa Skor Recall
Kemampuan model mendeteksi data positif.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

b. Evaluasi Performa F1-Score
rata-rata harmonik antara precision dan recall, yang memberikan gambaran keseluruhan kinerja model, terutama ketika ada ketidakseimbangan kelas.

$$F1 - Score = 2\ x \frac{precision\ x\ recall}{recall + precision}\ x\ 100\% \quad (9)$$

4. HASIL DAN PEMBAHASAN

4.1. Business Understanding

Usia dan Gaji merupakan salah satu kriteria yang diperhitungkan dalam membeli suatu produk, oleh sebab itu pada penelitian ini membuat program *machine learning* untuk memprediksi minat beli *customer* dalam membeli mobil berdasarkan usia dan gaji dengan melakukan klasifikasi menggunakan algoritma K-Nearest Neighbor.

4.2. Data Understanding

Tahap kedua yaitu *Data Understanding*, dimana pada penelitian ini diharuskan memahami seluk beluk dataset secara mendalam. Misalnya mengeksplorasi apa saja yang ada di dalam dataset seperti berapa banyak baris entries pada dataset, berapa banyak fitur yang ada pada dataset dan apa saja fitur didalamnya, memeriksa apakah dataset memiliki *missing value* atau tidak, tipe data apa saja yang terdapat pada dataset.

Tabel 2. Informasi Ringkas Dataset

Kolom	Data Tidak Kosong	Tipe Data
User ID	400	Integer
Gender	400	String
Age	400	Integer
EstimatedSalary	400	Integer
Purchased	400	integer

Tabel 2 menjelaskan Dataset yang digunakan memiliki 400 baris data dan 5 fitur di dalamnya, yaitu User ID, *Gender*, *Age*, *EstimatedSalary*, dan *Purchased*. Terdapat tipe data string dan integer di dalamnya.

4.3. Data Preparation

Pada tahap ketiga ini dataset diolah terlebih dahulu atau *preprocessing* sebelum dilakukan pemodelan. Pada tahap ini dataset yang digunakan tidak memiliki *missing value*, maka pada proses ini tidak perlu memperbaiki *missing value*. Tahap *preprocessing* selanjutnya adalah memisahkan kolom menjadi fungsi X (Usia dan Gaji) dan y (beli/tidak beli).

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.25, random_state = 0)
```

Tahap selanjutnya menjelaskan tahap split dataset menjadi 4 bagian, yaitu X_train, X_test, y_train, dan y_test. Data yang dipisah ini dialokasikan sebanyak 75% untuk data train dan 25% untuk data test. Tujuan nya agar program dapat mempelajari data yang dilatih dan data menjadi lebih sempurna serta meningkatkan nilai akurasi pada tahap modelling.

4.4. Modelling

```
mms = MinMaxScaler()
X_train = mms.fit_transform(X_train)
X_test = mms.transform(X_test)
```

Tahap selanjutnya menjelaskan tahap *scaling feature* pada dataset yang telah di preprocessing dengan fit transform yang merubah bentuk dataset menjadi ternormalisasi menggunakan *MinMaxScaler* yang mana untuk menstandarisasikan fitur dengan merubah skala nilai dari dataset kedalam skala diantara 0–1 atau desimal.

```
param_grid = {
    'n_neighbors': [3, 5, 7, 9],
    'metric': ['euclidean', 'manhattan', 'minkowski'],
    'p': [1, 2]
}

grid_search = GridSearchCV(estimator=classifier,
param_grid=param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)

print("Best Parameters:", grid_search.best_params_)
classifier = grid_search.best_estimator_
```

Tahap selanjutnya menjelaskan penerapan algoritma K-NN dengan menentukan nilai K menggunakan metric seperti *euclidean distance*, *manhattan*, dan *minkowski*. Untuk menentukan metric apa dan berapa nilai K yang paling optimal adalah dengan menggunakan *library sklearn* yaitu *GridSearchCV*. Dengan menggunakan *library* ini dapat mengetahui berapa nilai K yang paling akurat digunakan.

Selanjutnya adalah melakukan train model algoritma K-NN dengan melatih data train menggunakan metric dan nilai K yang sudah di dapatkan menggunakan *library GridSearchCV*.

```
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1),
y_test.reshape(len(y_test),1)),1))

print(classifier.predict(mms.transform([[30,
120000]])))
```

Sebagai uji coba penelitian ini dibuatkan fungsi prediksi untuk memprediksi minat beli *customer* dan mencoba menambahkan nilai random Usia dan Gaji yang tidak terdapat pada dataset yang sudah dimiliki yaitu usia 30 dan gaji 120000 menghasilkan nilai 1 yang berarti beli.

4.5. Evaluation

Gambar 2 menjelaskan performa model klasifikasi dengan menghitung dua metrik utama, yaitu:

```
Performa Confusion Matrix :
 [[64  4]
 [ 3 29]]

Akurasi Score :
 93.0 %
```

Gambar 3. Performa confusion matrix dan akurasi score

a. Confusion Matrix: Matriks ini menunjukkan bagaimana prediksi model dibandingkan dengan data sebenarnya, terdiri dari empat elemen utama:
   - 64 (True Negative): Jumlah data kelas 0 yang diprediksi dengan benar sebagai kelas 0.
   - 4 (False Positive): Jumlah data kelas 0 yang salah diprediksi sebagai kelas 1.
   - 3 (False Negative): Jumlah data kelas 1 yang salah diprediksi sebagai kelas 0.
   - 29 (True Positive): Jumlah data kelas 1 yang diprediksi dengan benar sebagai kelas 1.

   Matriks ini menunjukkan bahwa model membuat 64 prediksi benar untuk kelas 0 dan 29 prediksi benar untuk kelas 1, sementara ada 7 kesalahan (4 salah sebagai kelas 1, dan 3 salah sebagai kelas 0).
b. Akurasi Score: Akurasi dihitung sebagai jumlah prediksi yang benar dibagi dengan total jumlah data, dimana model berhasil memprediksi dengan benar sebanyak 93% dari keseluruhan data uji.

```
Hasil Akurasi Klasifikasi :
              precision    recall  f1-score   support

           0       0.96      0.94      0.95        68
           1       0.88      0.91      0.89        32

    accuracy                           0.93       100
   macro avg       0.92      0.92      0.92       100
weighted avg       0.93      0.93      0.93       100
```

Gambar 4. Classification report

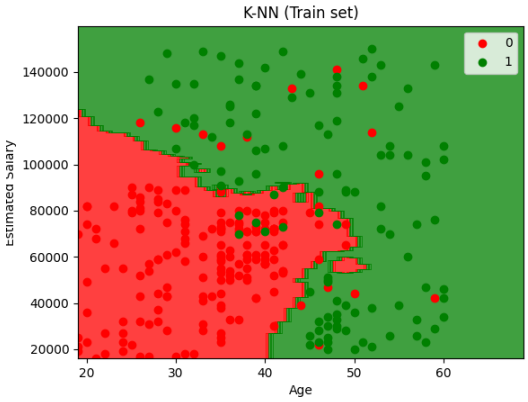
Gambar 3 menjelaskan pengukuran laporan yang berisi metrik evaluasi performa model klasifikasi secara mendetail sebagai berikut :

a. Precision: Mengukur proporsi prediksi positif yang benar. Kelas 0 memiliki *precision* sebesar 96%, artinya dari semua prediksi untuk kelas 0, 96% adalah benar. Kelas 1 memiliki *precision* sebesar 88%.
b. Recall: Mengukur kemampuan model untuk mendeteksi semua data yang benar dalam suatu kelas. Kelas 0 memiliki *recall* sebesar 94%, yang berarti model berhasil mendeteksi 94% dari total data sebenarnya untuk kelas 0. Kelas 1 memiliki *recall* sebesar 91%.
c. F1-Score: Rata-rata harmonis antara *precision* dan *recall*, yang memberikan keseimbangan antara kedua metrik tersebut. Kelas 0 memiliki *F1-score* sebesar 95%, sementara kelas 1 memiliki *F1-score* sebesar 89%.

d. Accuracy: Tingkat keseluruhan keakuratan model dalam memprediksi kelas dengan benar, yang mencapai 93% (dihitung dari total prediksi benar dibagi dengan jumlah total data).

Kode tersebut digunakan untuk menghasilkan dan menampilkan *classification report*, yaitu laporan yang berisi metrik evaluasi performa model klasifikasi secara mendetail. Fungsi classification_report dari pustaka *scikit-learn* menghitung berbagai metrik evaluasi, seperti *precision*, *recall*, dan *F1-score*. Metrik *precision* menunjukkan ketepatan model dalam memprediksi suatu kelas tertentu, recall mengukur kemampuan model dalam mendeteksi semua data yang benar dalam kelas tersebut, sementara *F1-score* merupakan rata-rata harmonis dari *precision* dan *recall* yang memberikan gambaran seimbang mengenai performa model. Dengan menampilkan laporan ini, pengguna dapat memperoleh wawasan yang lebih terperinci mengenai bagaimana model bekerja untuk setiap kelas, sehingga mempermudah analisis dan pengambilan keputusan dalam langkah-langkah perbaikan model.

Selanjutnya adalah menampilkan visualisasi model data train yang telah dilatih menggunakan library matplotlib

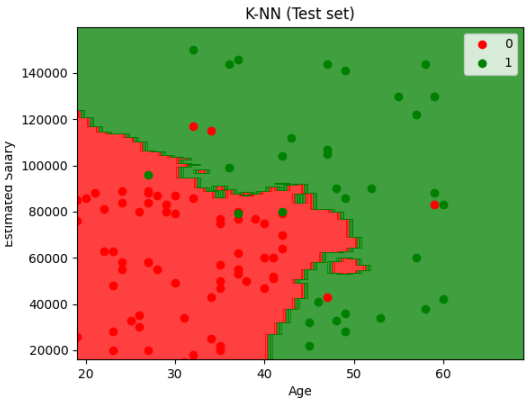


Gambar 5. Visualisasi K-NN train set

Gambar 5 menjelaskan pembagian wilayah keputusan (*decision boundary*) berdasarkan usia (*Age*) dan estimasi gaji *(Estimated Salary*). Berikut adalah penjelasannya :

a. Area berwarna merah menunjukkan wilayah di mana model memprediksi calon pelanggan tidak membeli (kelas 0).
b. Area berwarna hijau menunjukkan wilayah di mana model memprediksi calon pelanggan membeli **(**kelas 1).
c. Titik merah melambangkan data sebenarnya untuk kelas 0 (tidak membeli).
d. Titik hijau melambangkan data sebenarnya untuk kelas 1 (membeli).
e. Titik yang berada di area dengan warna yang sesuai (contoh: titik merah di area merah) menunjukkan prediksi yang benar oleh model.
f. Sebaliknya, titik yang berada di area warna yang tidak sesuai (contoh: titik hijau di area merah) menunjukkan kesalahan prediksi.

Berikutnya adalah hasil visualisasi model data test yang telah dilatih menggunakan library matplotlib.



Gambar 6. Visualisasi K-NN test set

Gambar 6 menjelaskan bagaimana model melakukan klasifikasi pada data *testing* berdasarkan fitur usia (*Age*) dan estimasi gaji (*Estimated Salary*). Berikut adalah penjelasan :

a. Pola *decision boundary* yang terlihat kompleks menunjukkan bahwa model mempertimbangkan kedekatan data dengan tetangganya untuk membuat keputusan.
b. Secara umum, sebagian besar titik data terlihat sesuai dengan wilayahnya, yang mengindikasikan performa model yang cukup baik.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan bahwa penerapan Algoritma K-Nearest Neighbor pada minat beli mobil bekas menggunakan pendekatan CRISP-DM telah berhasil dibuat dengan tingkat akurasi hampir sempurna sebesar 93%, melalui pembagian dua cluster, yaitu cluster 1 (beli) dan cluster 0 (tidak beli). Pada performa akurasi recall, model memperoleh skor sebesar 94% pada cluster 0 dan 91% pada cluster 1, sedangkan pada performa F1 score, model mendapatkan skor sebesar 95% pada cluster 0 dan 89% pada cluster 1. Selain itu, hasil visualisasi pada data latih (Train Set) menunjukkan bahwa model K-NN mampu memisahkan data dengan baik, di mana sebagian besar titik berada di area yang sesuai dengan label sebenarnya. Pada data uji (Test Set), model masih mampu memprediksi dengan cukup baik meskipun terdapat beberapa kesalahan klasifikasi seperti berberapa titik merah yang berdada di area hijau, ataupun berberapa titik hijau berada pada area merah. Pembagian wilayah prediksi tetap mengikuti pola dari data latih, tetapi dengan tingkat kerumitan yang lebih sederhana, menunjukkan kemampuan generalisasi model yang baik terhadap data baru. Secara keseluruhan, model ini cukup baik untuk tugas

klasifikasi berdasarkan fitur Age dan Estimated Salary. Untuk saran penelitian selanjutnya dapat dikembangkan dengan menggunakan data yang lebih banyak atau menggunakan metode klasifikasi yang berbeda untuk bisa mendapatkan hasil akurasi yang lebih tinggi.

DAFTAR PUSTAKA

[1] A. Nugraha, "Analisis Bauran Pemasaran dalam Penjualan Mobil Bekas di Perkasa Mobil," *Jurnal Administrasi Bisnis FISIPOL UNMUL*, vol. 10, no. 4, hlm. 273–278, 2022, [Daring]. Tersedia pada: http://e-journals.unmul.ac.id/index.php/jadbis/index

[2] R. W. Dari, "Metode Multi Attribute Utility Theory (MAUT) untuk Sistem Pendukung Keputusan Pemilihan Mobil Bekas," *Jurnal KomtekInfo*, vol. 10, no. 2, hlm. 73–79, Jun 2023, doi: 10.35134/komtekinfo.v10i2.378.

[3] I. Ali dan A. Rizki Rinaldi, "PENERAPAN METODE K-NEAREST NEIGHBOR UNTUK PREDIKSI PENJUALAN SEPEDA MOTOR TERLARIS," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1, hlm. 585–590, Feb 2023.

[4] N. Azizah, M. R. Firdaus, R. Suyaningsih, F. Indrayatna, dan U. Padjadjaran, "Penerapan Algoritma Klasifikasi K-Nearest Neighbor pada Penyakit Diabetes," *SEMINAR NASIONAL STATISTIKA AKTUARIA II*, 2023, [Daring]. Tersedia pada: http://prosiding.snsa.statistics.unpad.ac.id

[5] D. Imantata Muhammad dan N. Falih, "Penggunaan K-Nearest Neighbor (KNN) untuk Mengklasifikasi Citra Belimbing Berdasarkan Fitur Warna," *JURNAL INFORMATIK*, vol. 17, no. 1, hlm. 9, Apr 2021.

[6] J. Homepage, S. R. Cholil, T. Handayani, R. Prathivi, dan T. Ardianita, "IJCIT (Indonesian Journal on Computer and Information Technology) Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa," *IJCIT (Indonesian Journal on Computer and Information Technology)*, vol. 6, no. 2, hlm. 118–127, Nov 2021.

[7] P. R. Sihombing dan A. M. Arsani, "COMPARISON OF MACHINE LEARNING METHODS IN CLASSIFYING POVERTY IN INDONESIA IN 2018," *Jurnal Teknik Informatika (Jutif)*, vol. 2, no. 1, hlm. 51–56, Jan 2021, doi: 10.20884/1.jutif.2021.2.1.52.

[8] R. G. Wardhana, G. Wang, dan F. Sibuea, "PENERAPAN MACHINE LEARNING DALAM PREDIKSI TINGKAT KASUS PENYAKIT DI INDONESIA," *Journal of Information System Management (JOISM) e-ISSN*, vol. 5, no. 1, hlm. 40–45, 2023.

[9] N. Ajijah dan A. Kurniawan, "Klasifikasi Teks Mining Terhadap Analisa Isu Kegiatan Tenaga Lapangan Menggunakan Algoritma K-Nearest Neighbor (KNN)," *Jurnal Sains Komputer & Informatika (J-SAKTI*, vol. 7, no. 1, hlm. 254–262, Mar 2023.

[10] A. Pebdika, R. Herdiana, dan D. Solihudin, "KLASIFIKASI MENGGUNAKAN METODE NAIVE BAYES UNTUK MENENTUKAN CALON PENERIMA PIP," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, no. 1, hlm. 452–458, Feb 2023.

[11] K. Kristiawan dan A. Widjaja, "Perbandingan Algoritma Machine Learning dalam Menilai Sebuah Lokasi Toko Ritel," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr 2021, doi: 10.28932/jutisi.v7i1.3182.

[12] O. D. Kurnia *dkk.*, "Analisis Perbandingan Algoritma Naïve Bayes dengan K-Nearest Neighbor (KNN) Pada Dataset Mobile Price Classification Penulis Korespondensi: Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi) 1174," *INOTEK*, vol. 8, hlm. 1174–1183, Agu 2024.

[13] V. Bayu Anwari, "SKANIKA: Sistem Komputer dan Teknik Informatika Implementasi Algoritma K-Nearest Neighbors Pada Analisis Sentimen Masyarakat Terhadap Penerapan Pemberlakuan Pembatasan Kegiatan Masyarakat," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 1, hlm. 72–81, Jan 2022.

[14] R. Lutfansyah, "Analisis Kelayakan Pembiayaan Anggota Koperasi Dengan Metode Komparasi Algoritma K-Nearest Neighbors Dan Naive Bayes (Studi Kasus Pada KSP. XYZ)," *JURNAL ILMU KOMPUTER*, vol. 2, no. 2, hlm. 169–177, Des 2024.