

## **Case Based – Learning**

untuk memenuhi tugas Mata Kuliah

Kecerdasan Buatan RPL



Oleh :

Mohammed Yousef Gumilar 1302220085

Muhammad Ghiyats Fatiha 1302220109

**Kelas: SE-46-02**

Dosen Pengampu : Fauzan Firdaus, S.Kom., M.T. [FFS]

**PROGRAM STUDI S-1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2025**

## Daftar Isi

Daftar Isi.....	2
1. Deskripsi Kasus.....	3
2. Analisa Proses Preprocessing.....	3
3. Model Evaluasi.....	3
4. Implementasi.....	3
5. Hasil Running Program.....	6
6. Peran Anggota Kelompok.....	6

## 1. Deskripsi Kasus

Dalam dunia industri minuman khususnya wine, kualitas produk menjadi faktor yang penting dalam nilai jual. Oleh karena itu, diperlukan sistem klasifikasi otomatis berdasarkan fitur kimia untuk membantu menentukan kualitas wine. Proyek ini bertujuan membangun model klasifikasi kualitas wine menggunakan algoritma K-Nearest Neighbour (KNN) secara manual yang memprediksi kualitas wine berdasarkan fitur-fitur kimia.

Dataset yang digunakan diperoleh dari sumber terbuka (Kaggle) dengan berbagai fitur numerik seperti fixed acidity, residual sugar, alcohol, dan density, dan label target yang menunjukkan kualitas wine (low, medium, high).

## 2. Analisa Proses Preprocessing

Proses pra-pemrosesan pada algoritma baru ini jauh lebih superior karena menerapkan Normalisasi Min-Max pada empat fitur terpilih, memastikan semua fitur memiliki bobot yang setara dalam perhitungan jarak dan mengatasi kelemahan utama dari versi sebelumnya. Selain itu, penggunaan DictReader untuk memuat data sebagai daftar dictionary membuat kode lebih terstruktur dan mudah dikelola, sementara rentang nilai asli dari proses normalisasi disimpan secara cerdas untuk digunakan kembali saat demo interaktif.

## 3. Desain Algoritma

Desain algoritma ini dimulai dengan memuat data sebagai daftar *dictionary* yang terstruktur, memilih empat fitur utama, dan menstandarkan labelnya. Langkah krusial selanjutnya adalah melakukan Normalisasi Min-Max untuk mengubah skala semua fitur ke rentang 0-1, memastikan perhitungan jarak Euclidean tidak bias. Data yang telah ternormalisasi kemudian dibagi menjadi 70% data latih dan 30% data uji. Proses prediksi ini menggunakan  $K=3$ , di mana untuk setiap sampel baru, algoritma mencari tiga tetangga terdekat dari data latih dan menentukan hasilnya melalui voting mayoritas. Kinerja model dievaluasi secara komprehensif menggunakan metrik rata-rata makro pada kedua set data, dan alur kerja ini diakhiri dengan mode demo interaktif di mana input pengguna juga dinormalisasi sebelum diprediksi.

## 4. Model Evaluasi

Evaluasi model dilakukan secara rapi dan efisien, di mana fungsi `predict_knn` secara ringkas menghitung jarak dan melakukan prediksi berdasarkan voting mayoritas. Metrik performa : akurasi, presisi, recall, dan F1-score dihitung menggunakan metode rata-rata makro (macro average), sebuah pendekatan yang solid untuk kasus multi-kelas karena memberikan perlakuan yang adil untuk setiap kelas ('low', 'medium', 'high'), sementara penggunaan  $K=3$  sebagai nilai default membantu menjaga stabilitas prediksi.

## 5. Implementasi

0. Import	<pre>import csv import math import random</pre> <p>Mengimport library yang dibutuhkan</p>
1. Label Handling	<pre>def relabel(label: str) -&gt; str:     return label.strip().lower()</pre> <p>Membersihkan dan menstandarkan label kualitas. Dataset sudah pakai kata *low / medium / high*. Kita hanya memastikan hasil akhir huruf kecil tanpa spasi ekstra sehingga konsisten di model.</p>
2. Data Loading	<pre>def load_data(filename: str):     data = []     with open(filename, newline="") as f:         reader = csv.DictReader(f)         for row in reader:             data.append({                 "fixed_acidity": float(row["fixed_acidity"]),                 "residual_sugar": float(row["residual_sugar"]),                 "alcohol": float(row["alcohol"]),                 "density": float(row["density"]),                 "label": relabel(row["quality_label"]),             })     return data</pre> <p>Membaca CSV → list of dict + konversi tipe data. Kolom numerik di-cast ke *float* agar bisa diolah matematis. Label dikirim ke `relabel()` untuk distandarkan.</p>
3. Normalization (min-max)	<pre>def normalize(data):     ranges = {}     keys = ["fixed_acidity", "residual_sugar", "alcohol", "density"]     for k in keys:         vals = [row[k] for row in data]         lo, hi = min(vals), max(vals)         ranges[k] = (lo, hi)     for row in data:         row[k] = (row[k] - lo) / (hi - lo) if hi != lo else 0.0     return data, ranges</pre> <p>Skalakan ke rentang 0-1 per fitur + simpan rentang awal. KNN sensitif terhadap skala: fitur besar bisa mendominasi jarak.</p>
4. Train-Test Split (70-30)	<pre>def split_data(data, ratio: float = 0.7):     random.seed(42)     random.shuffle(data)     cut = int(len(data) * ratio)     return data[:cut], data[cut:]</pre> <p>Mengacak lalu membagi data → (train, test).</p>
5. K-Nearest Neighbor Primitives	<pre>def euclidean(a, b):     return math.sqrt(         (a["fixed_acidity"] - b["fixed_acidity"]) ** 2 +         (a["residual_sugar"] - b["residual_sugar"]) ** 2 +         (a["alcohol"] - b["alcohol"]) ** 2 +         (a["density"] - b["density"]) ** 2     )</pre> <p>Jarak Euclidean antar dua sampel (4 dimensi).</p>

	<pre>def majority_vote(labels):     counts = {}     for lbl in labels:         counts[lbl] = counts.get(lbl, 0) + 1     return max(counts, key=counts.get)</pre> <p>Mengembalikan label terbanyak di *labels* (tanpa Counter).</p> <pre>def predict_knn(train, sample, k: int = 3):     distances = [(euclidean(sample, row), row["label"]) for row in train]     distances.sort(key=lambda tup: tup[0])     top_k_labels = [lbl for _, lbl in distances[:k]]     return majority_vote(top_k_labels)</pre> <p>Prediksi satu *sample* menggunakan KNN dengan k=3.</p> <ul style="list-style-type: none"> <li># 1) Hitung jarak ke setiap data training</li> <li># 2) Urutkan dari jarak terkecil</li> <li># 3) Ambil k label terdekat</li> </ul>
<p>6. Evaluation Metrics (macro-averaged)</p>	<pre>def evaluate(y_true, y_pred):     labels = set(y_true)     accuracy = sum(t == p for t, p in zip(y_true, y_pred)) / len(y_true)     precision_sum = recall_sum = f1_sum = 0     for lbl in labels:         tp = fp = fn = 0         for t, p in zip(y_true, y_pred):             if t == lbl and p == lbl:                 tp += 1             elif t != lbl and p == lbl:                 fp += 1             elif t == lbl and p != lbl:                 fn += 1         prec_lbl = tp / (tp + fp) if (tp + fp) else 0         recall_lbl = tp / (tp + fn) if (tp + fn) else 0         f1_lbl = 2 * prec_lbl * recall_lbl / (prec_lbl + recall_lbl) if (prec_lbl + recall_lbl) else 0         precision_sum += prec_lbl         recall_sum += recall_lbl         f1_sum += f1_lbl      n = len(labels)     return {         "accuracy": accuracy,         "precision": precision_sum / n,         "recall": recall_sum / n,         "f1_score": f1_sum / n,     }</pre> <p>Hitung akurasi, presisi, recall, F1 (macro).</p>
<p>7. Main Program</p>	<pre>def main():     data = load_data("wine_quality_classification.csv")     data, ranges = normalize(data)     train, test = split_data(data, 0.7)     k = 3      y_true_train = [row["label"] for row in train]     y_pred_train = [predict_knn(train, row, k) for row in train]      y_true_test = [row["label"] for row in test]     y_pred_test = [predict_knn(train, row, k) for row in test]</pre>

```

print("Evaluasi Training:")
for metric, val in evaluate(y_true_train, y_pred_train).items():
    print(f"{metric.capitalize():10}: {val:.4f}")

print("\nEvaluasi Testing:")
for metric, val in evaluate(y_true_test, y_pred_test).items():
    print(f"{metric.capitalize():10}: {val:.4f}")

print("\n===== Inputkan Spesifikasi Wine =====")
user = {
    "fixed_acidity": float(input("Fixed Acidity (4-16): ")),      # g/L
    "residual_sugar": float(input("Residual Sugar (0.5-15): ")),  # g/L
    "alcohol": float(input("Alcohol (%) (8-14): ")),             # % vol.
    "density": float(input("Density (0.9900-1.1): ")),           # g/cm³
}

for feat in user:
    lo, hi = ranges[feat]
    user[feat] = (user[feat] - lo) / (hi - lo) if hi != lo else 0.0
prediction = predict_knn(train, user, k)
print("\nPrediksi kualitas wine:", prediction.upper())

if __name__ == "__main__":
    main()

```

Main program aplikasi

## 5. Hasil Running Program

Berdasarkan hasil eksekusi program file learning.py ([Link Github](#)), algoritma menghasilkan output sebagai berikut :

```

PS D:\SEMESTER_6\2_KECERDASAN BUATAN\UTS\Case Based Searching\Ai-project-based-learning> python learning.py
Evaluasi Training:
Accuracy : 0.8000
Precision : 0.8009
Recall : 0.8000
F1_score : 0.8003

Evaluasi Testing:
Accuracy : 0.3567
Precision : 0.3550
Recall : 0.3568
F1_score : 0.3556

===== Inputkan Spesifikasi Wine =====
Fixed Acidity (4-16): 14
Residual Sugar (0.5-15): 6.4
Alcohol (%) (8-14): 13.2
Density (0.9900-1.1): 0.9918

Prediksi kualitas wine: LOW

```

```

PS D:\SEMESTER_6\2_KECERDASAN BUATAN\UTS\Case Based Searching\Ai-project-based-learning> python learning.py
Evaluasi Training:
Accuracy : 0.8000
Precision : 0.8009
Recall : 0.8000
F1_score : 0.8003

Evaluasi Testing:
Accuracy : 0.3567
Precision : 0.3550
Recall : 0.3568
F1_score : 0.3556

===== Inputkan Spesifikasi Wine =====
Fixed Acidity (4-16): 7
Residual Sugar (0.5-15): 4
Alcohol (%) (8-14): 8.7
Density (0.9900-1.1): 0.9911

Prediksi kualitas wine: MEDIUM

PS D:\SEMESTER_6\2_KECERDASAN BUATAN\UTS\Case Based Searching\Ai-project-based-learning> python learning.py
Evaluasi Training:
Accuracy : 0.8000
Precision : 0.8009
Recall : 0.8000
F1_score : 0.8003

Evaluasi Testing:
Accuracy : 0.3567
Precision : 0.3550
Recall : 0.3568
F1_score : 0.3556

===== Inputkan Spesifikasi Wine =====
Fixed Acidity (4-16): 5.0
Residual Sugar (0.5-15): 5.8
Alcohol (%) (8-14): 13.6
Density (0.9900-1.1): 0.9963

Prediksi kualitas wine: HIGH

```

## 6. Peran Anggota Kelompok

- ❖ Mohammed Yousef Gumilar (1302220085): Perancangan dan penulisan kode Python, Penulisan laporan, dan penyusunan dokumentasi.
- ❖ Muhammad Ghiyats Fatiha (1302220109): Pengujian dan debugging, Penulisan laporan dan PPT, dan penyusunan dokumentasi.