



CASE BASED – LEARNING

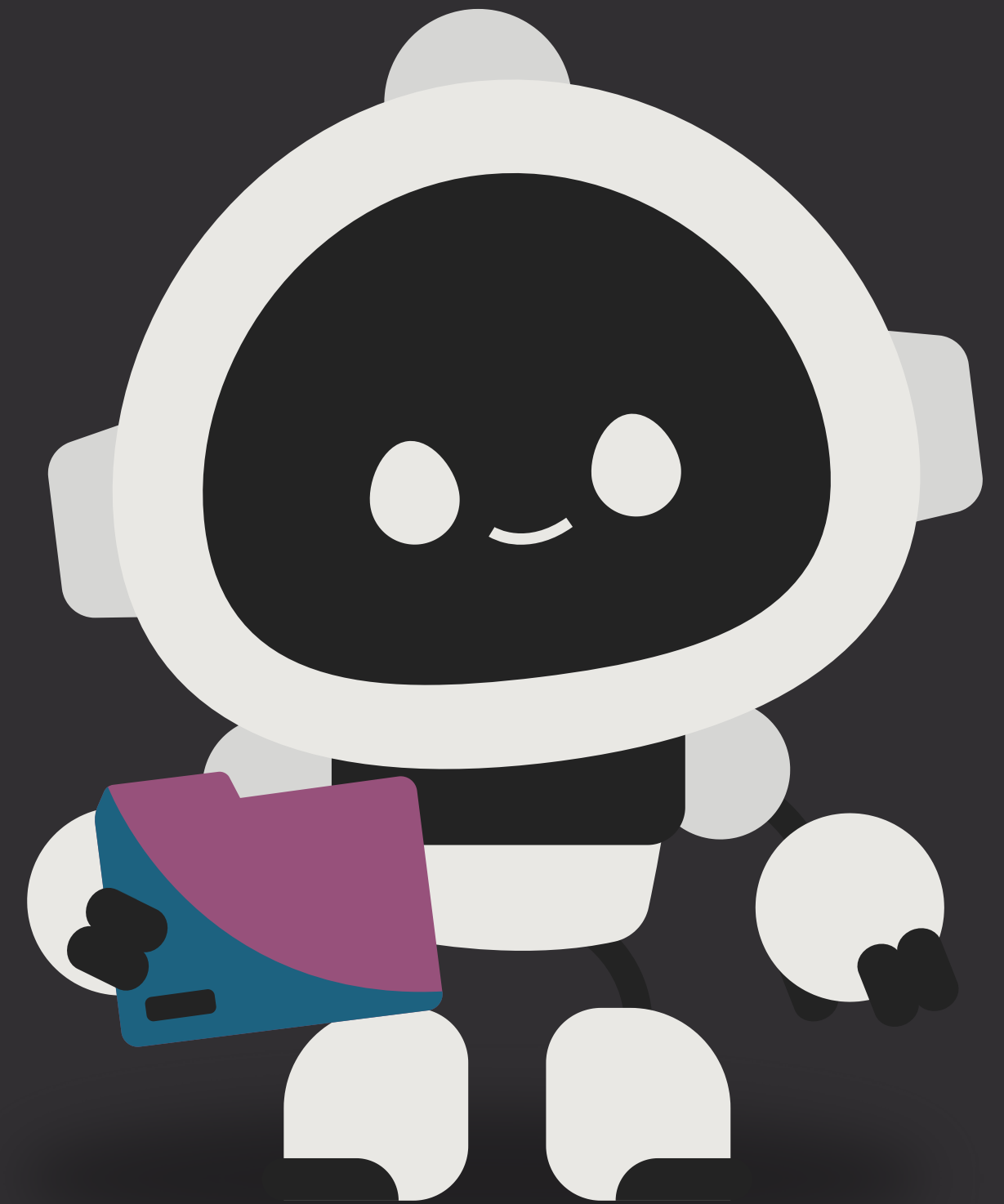


SE-46-02 : Kelompok 7

ANGGOTA

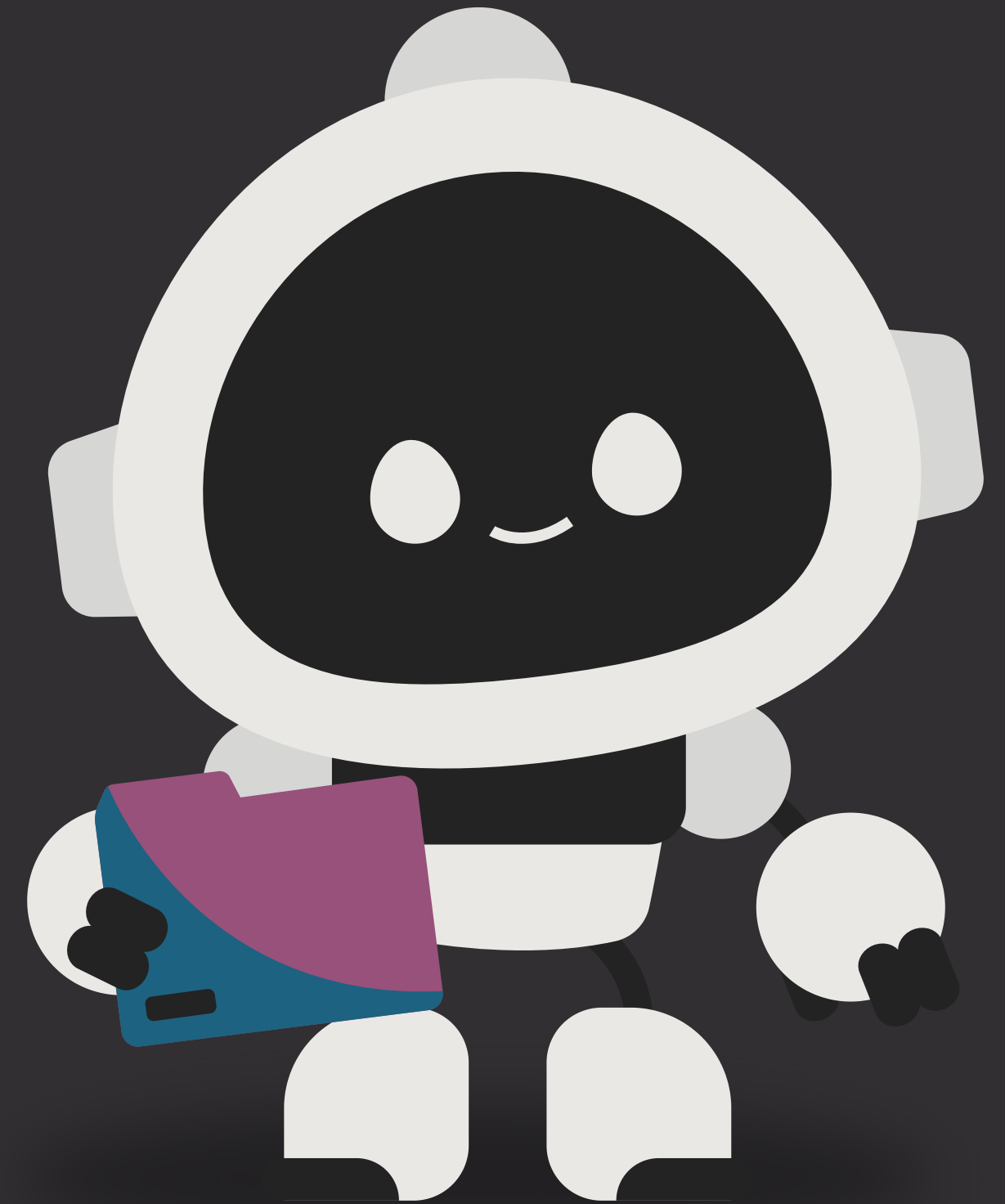
Mohammed Yousef Gumilar
(1302220085)

Muhammad Ghiyats Fatiha
(1302220109)



PEMBAHASAN

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





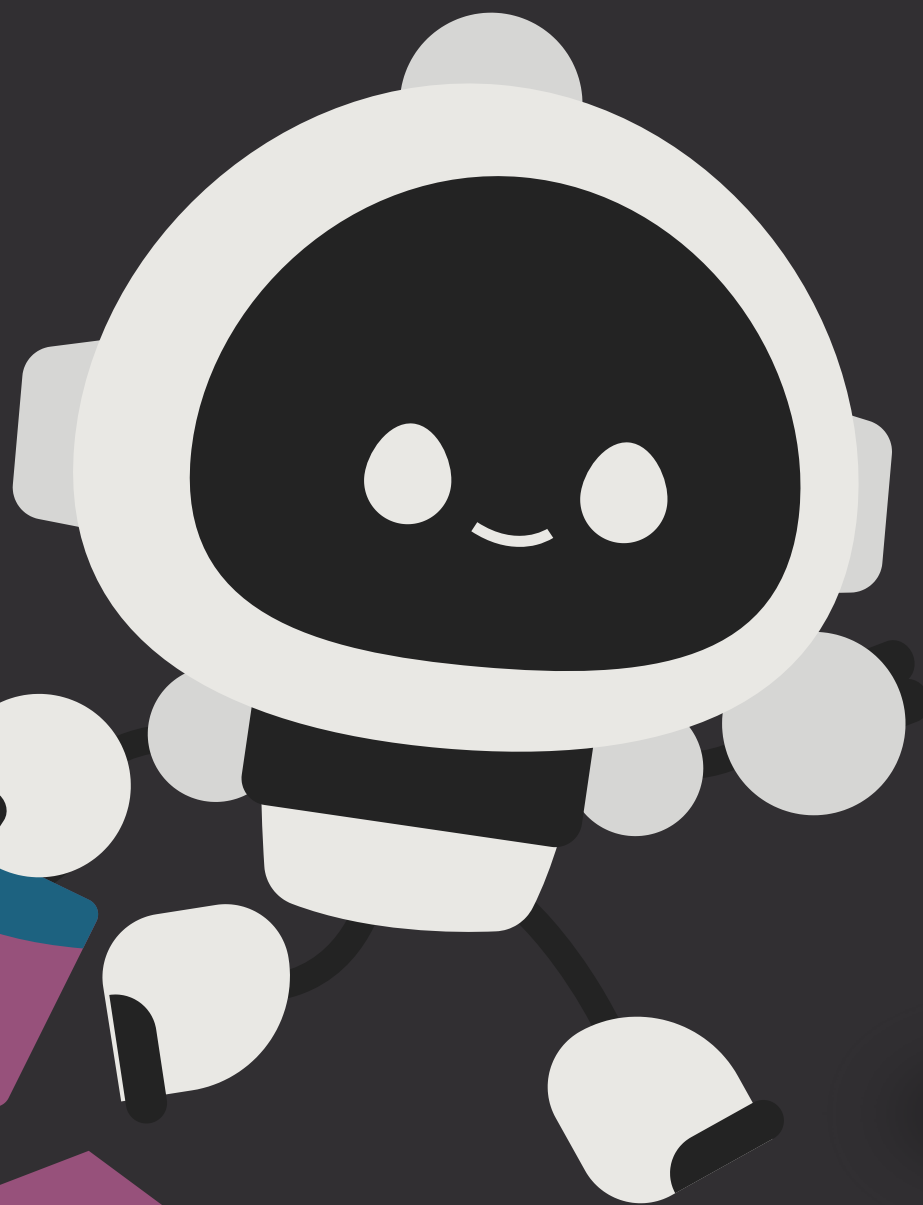
DESKRIPSI KASUS

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.

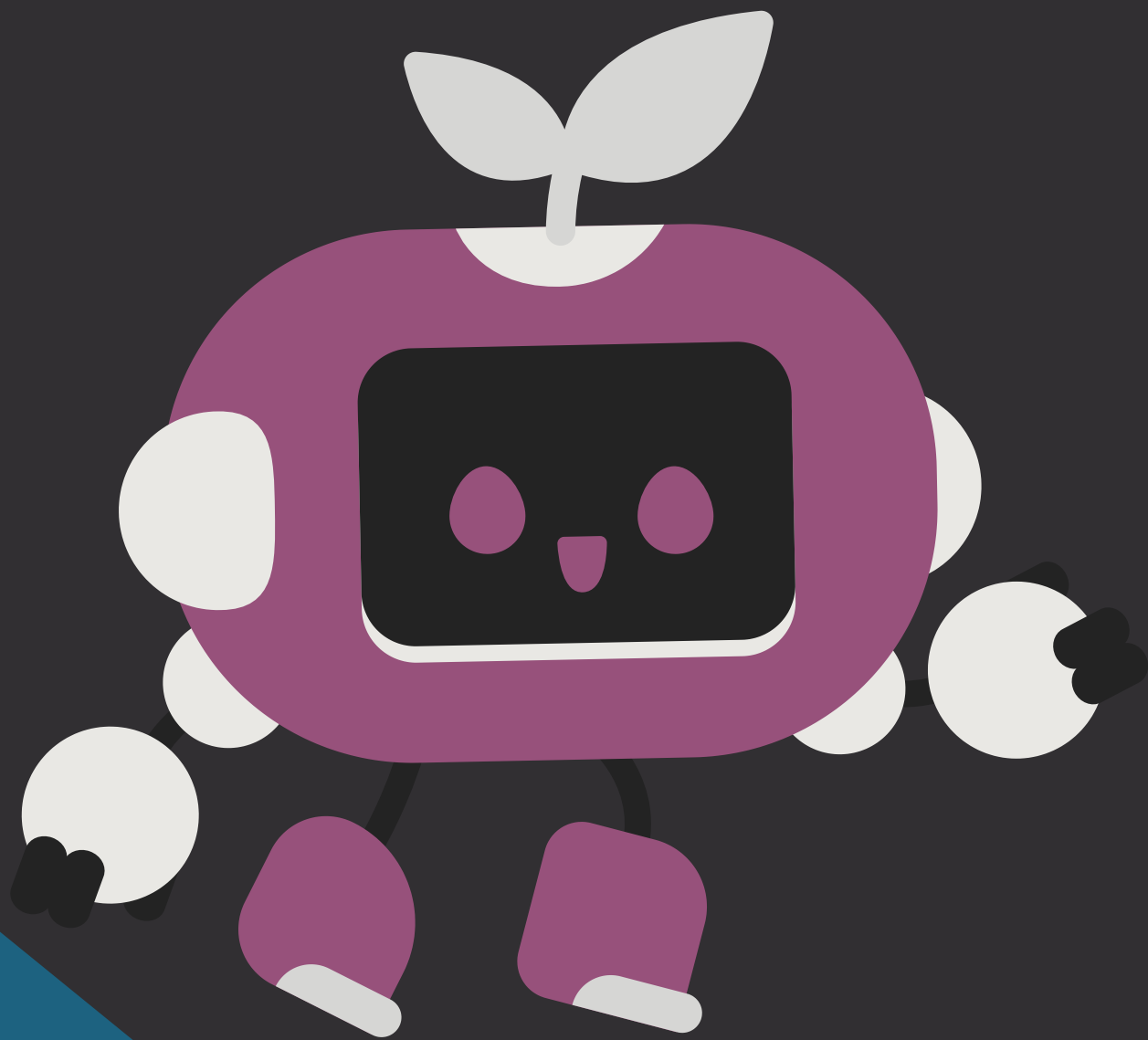


ANALISA PROSES PREPROCESSING

Pra-pemrosesan algoritma baru ditingkatkan dengan Normalisasi Min-Max pada empat fitur, memastikan bobot setara dan mengatasi kelemahan versi sebelumnya. DictReader digunakan untuk memuat data sebagai daftar dictionary, membuat kode lebih terstruktur, dan rentang nilai asli normalisasi disimpan untuk demo interaktif.

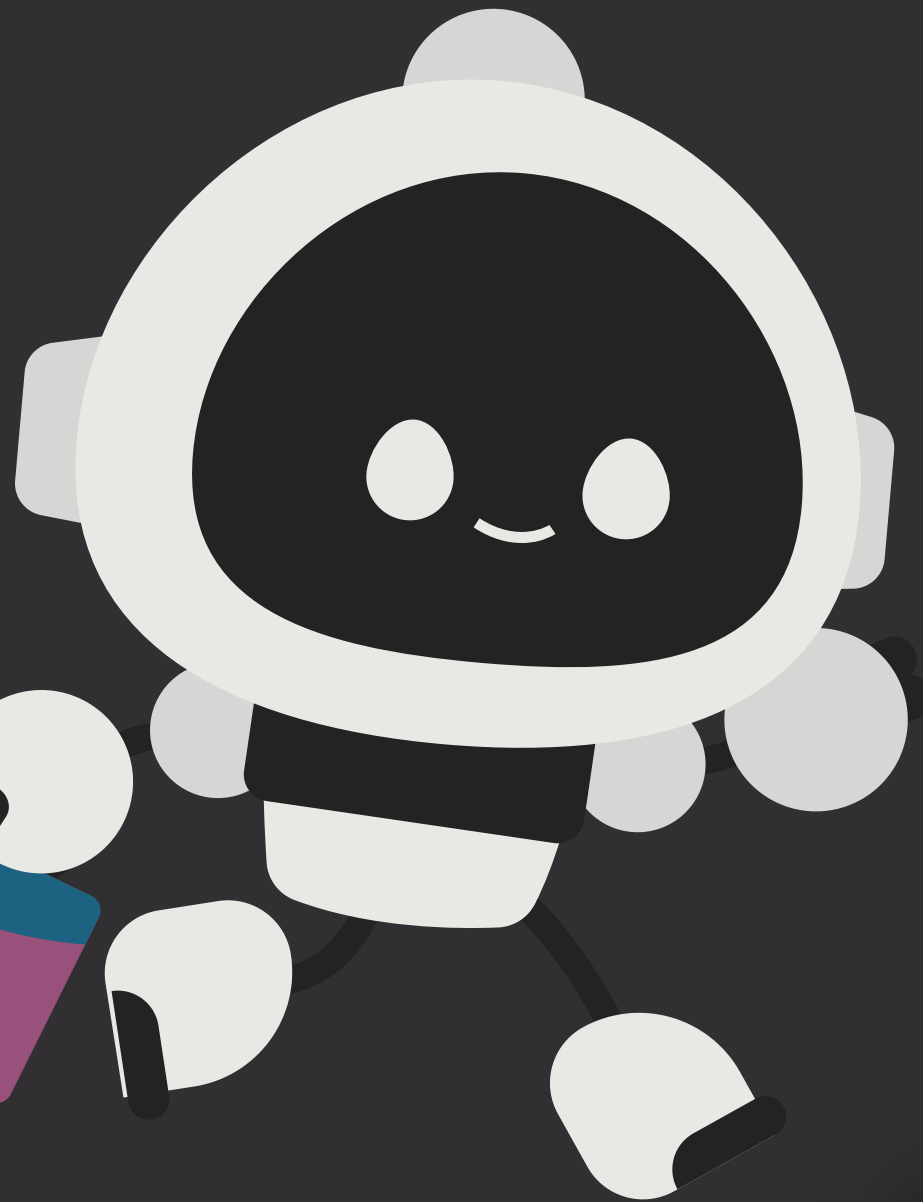


DESAIN ALGORITMA



Algoritma dimulai dengan memuat data sebagai daftar dictionary, memilih empat fitur utama, dan menstandarkan labelnya. Normalisasi Min-Max dilakukan untuk mengubah skala fitur ke 0-1. Data dibagi menjadi 70% data latih dan 30% data uji. Prediksi menggunakan $K=3$, mencari tiga tetangga terdekat dan menentukan hasil melalui voting mayoritas. Kinerja dievaluasi dengan metrik rata-rata makro pada kedua set data. Mode demo interaktif menormalisasi input pengguna sebelum prediksi.

MODEL EVALUASI



Valuasi model KNN dilakukan dengan membagi dataset menjadi 70% data latih dan 30% data uji menggunakan `random.seed(42)` untuk konsistensi. Performa model diukur dengan akurasi, presisi (macro avg), recall (macro avg), dan F1-Score (macro avg), di mana rata-rata makro memberikan bobot yang sama untuk setiap kelas.

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) -> 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> |

IMPLEMENTASI

3. Normalization (min-max)

```
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
```

Skalakan ke rentang 0-1 per fitur + simpan rentang awal.
KNN sensitif terhadap skala: fitur besar bisa mendominasi jarak.

4. Train-Test Split (70-30)

```
def split_data(data, ratio: float = 0.7):  
    random.seed(42)  
    random.shuffle(data)  
    cut = int(len(data) * ratio)  
    return data[:cut], data[cut:]
```

Mengacak lalu membagi data → (train, test).

IMPLEMENTASI

5. K-Nearest Neighbor Primitives

```
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  
    )
```

Jarak Euclidean antar dua sampel (4 dimensi).

```
def majority_vote(labels):  
    counts = {}  
    for lbl in labels:  
        counts[lbl] = counts.get(lbl, 0) + 1  
    return max(counts, key=counts.get)
```

Mengembalikan label terbanyak di *labels* (tanpa Counter).

```
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)
```

Prediksi satu *sample* menggunakan KNN dengan k=3.

- # 1) Hitung jarak ke setiap data training
- # 2) Urutkan dari jarak terkecil
- # 3) Ambil k label terdekat

IMPLEMENTASI

6. Evaluation Metrics (macro-averaged)

```
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,
    }
```

Hitung akurasi, presisi, recall, F1 (macro).

IMPLEMENTASI

7. Main Program

```
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]

    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

HASIL RUNNING PROGRAM

```
PS D:\SEMESTER_6\2_KECERDASAN BUATAN\UTS\Cas
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
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
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
```



THANK
YOU

