**Laporan Praktikum Kontrol Cerdas**

**Nama : Nefi Afif Sujatyana**

**NIM : 224308093**

**Kelas : TKA-7D**

**Akun Github :** [**https://github.com/nefiafif612-cmd**](https://github.com/nefiafif612-cmd)

1. Judul Percobaan: ***Object Classification with Scikit-learn***
2. Tujuan Percobaan:
   * Memahami dasar-dasar Machine Learning dalam sistem kendali.
   * Mengimplementasikan model ML sederhana untuk klasifikasi objek.
   * Menggunakan Scikit-learn untuk membuat model ML dasar.
   * Mengintegrasikan model ML dengan Computer Vision untuk deteksi objek.
   * Mengelola dataseet dan melakukan pelatihan model sederhana.
3. Landasan Teori:
4. Analisis dan Diskusi:
   * Analisis
   * Diskusi
5. *Assigment*:

Praktikum di minggu kedua ini yaitu melakukan sejumlah tugas yaitu (penugasan variabel) untuk menyimpan data dan konfigurasi penting yang diperlukan dalam deteksi warna. Pertama, **file\_path** diisi dengan lokasi dataseet dengan format warna CSV, yang selanjutnya dibaca lewat ‘**pd.read\_csv()**’ dan disimpan dalam bentuk variabel **color\_data**. Kemudian, X diisi dengan nilai RGB yang diambil dari kolom ‘R’,

‘G’, dan ‘B’, sementara y menyimpan label warna yang terdapat dalam kolom ‘**ColorName**’. Karena label berbentuk teks, **LabelEncoder** diterapkan untuk mengubah label tersebut menjadi bentuk numerik yang disimpan dalam variabel **y\_encoded**. Fitur RGB kemudian dinormalisasi menggunakan **StandardScaler**, dan hasilnya disimpan di **X\_scaled** agar model SVM beroperasi secara optimal. Selanjutnya, dataset dipisahkan menjadi data pelatihan dan pengujian dengan menggunakan ‘**train\_test\_split()**’, dan hasilnya disimpan dalam variabel **X\_train, X\_test, y\_train, serta y\_test**.

Dalam bagian model *machine learning*, **svm\_model** mendapatkan objek ‘**SVC()**’ yang diatur dengan **kernel RBF**, serta parameter ‘**C=10**’ dan ‘**gamma=’scale’**’. Model ini kemudian dilatih dengan data pelatihan, dan hasil prediksinya disimpan dalam **y\_pred** untuk tujuan evaluasi akurasi. Untuk deteksi warna secara *real-time*, diisi dengan objek ‘**cv2.VideoCapture(0)**’ guna mengakses kamera. Variabel **detected\_colors\_1**, **detected\_colors\_2**, **detected\_true\_labels\_1**, dan **detected\_true\_labels\_2** digunakan untuk menyimpan hasil prediksi serta label asli secara dinamis, yang kemudian dipakai untuk menghitung *running accuracy*.

Setiap kali satu frame diambil dari kamera, dua ***Region of Interest*** (**ROI**) dihitung dan disimpan dalam **ROI1** dan **ROI2**, sementara warna rata-rata dalam ROI tersimpan di **avg\_color1** dan **avg\_color2**. Warna rata-rata ini dibalik urutannya dari **BGR** ke **RGB** dan dinormalisasi sebelum digunakan untuk prediksi warna oleh model SVM. Hasil prediksi disimpan dalam **color\_pred1** dan **color\_pred2**, yang kemudian dibandingkan dengan warna terdekat dari dataset melalui fungsi **find\_nearest\_color()**.

Hasil prediksi serta akurasi ditampilkan di layar kamera menggunakan OpenCV, dengan informasi yang diperbarui secara dinamis berdasarkan *running accuracy* dari 50 prediksi terakhir. Penugasan variabel yang terstruktur dengan baik dalam program ini memungkinkan aliran data yang lancar dari *preprocessing, training model*, hingga prediksi *real-time*.

1. Data dan Output Hasil Pengamatan:

1. Kesimpulan:
2. Saran:
3. Daftar Pustaka: