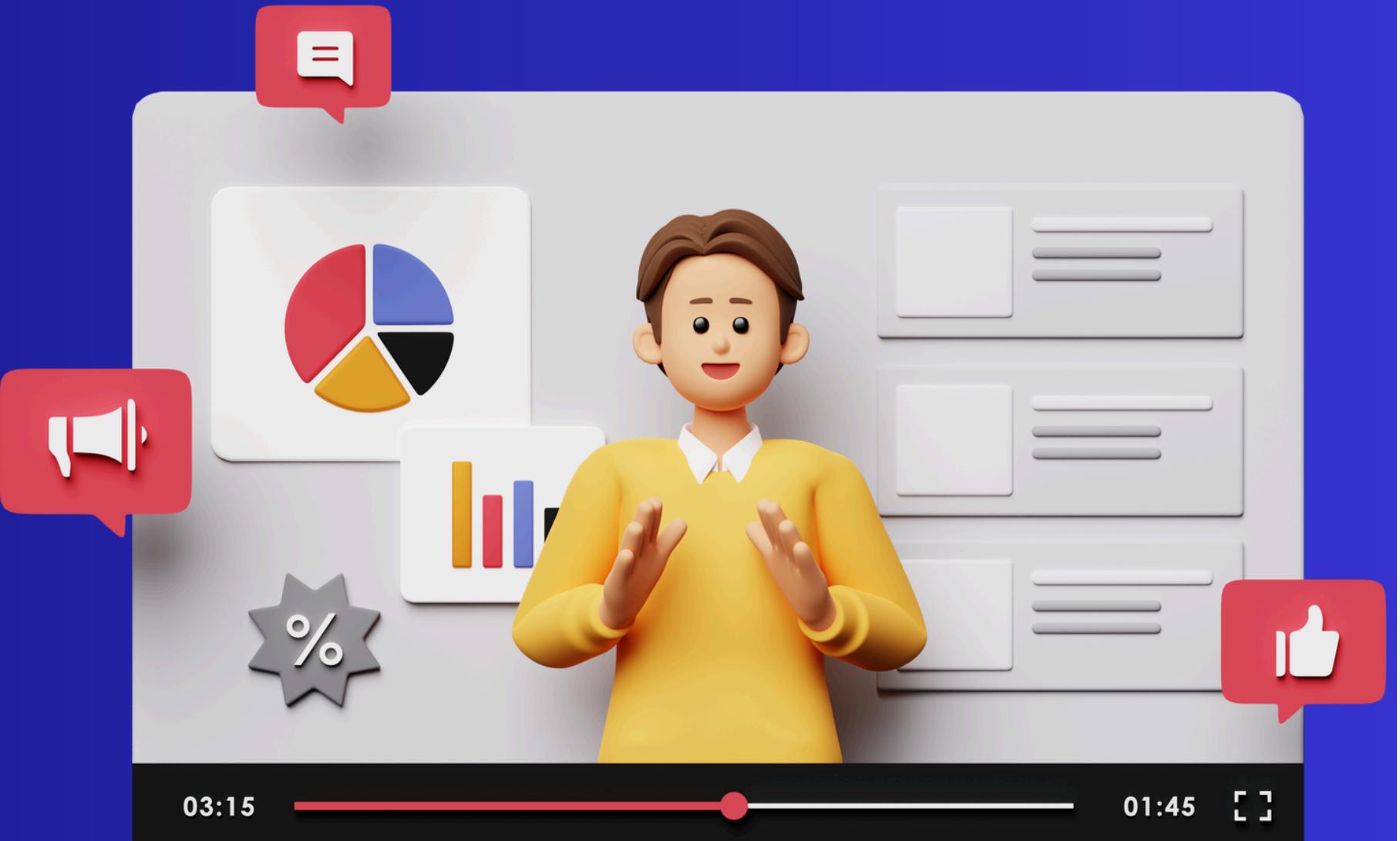


PROJECT PENGOLAHAN CITRA DIGITAL

MARIPOSA: Melakukan Deteksi
Terhadap Jenis Kupu-Kupu
Beracun



Our Member



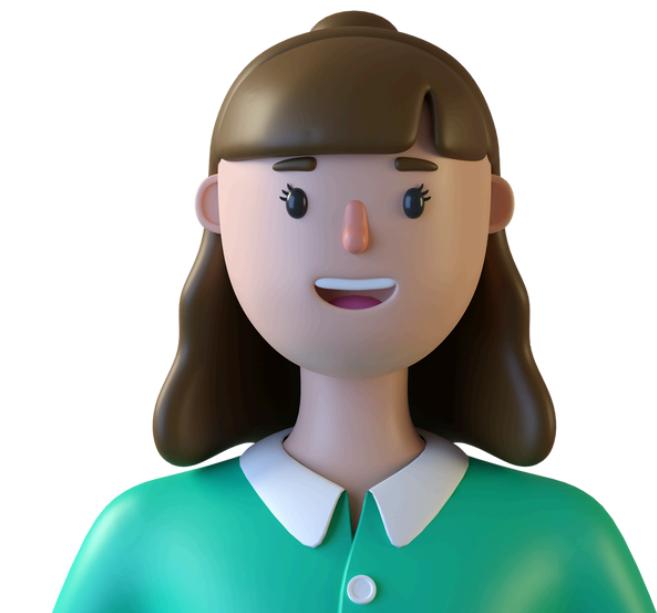
Adina Ayu Daraswita
F1D022030



Nabila Noor Azizah
F1D022081



Rizal Dwi Kurniawan
F1D022091



Ajundasrika Anugrahanti TS
F1D022108



Iman Handriawan B
F1D22320010

Latar Belakang

Pemilihan Dataset Kupu-Kupu

Kupu-kupu adalah salah satu serangga yang paling indah dan menarik di bumi ini. Tapi ternyata dibalik keindahannya itu tersimpan racun dalam tubuhnya.

Secara ilmiah, spesies kupu-kupu seperti Monarch (*Danaus plexippus*), Pipevine Swallowtail (*Battus philenor*), dan Striated Queen (*Danaus gilippus*) terkenal karena toksisitasnya yang berasal dari konsumsi tanaman inangnya selama tahap larva. Racun yang terkandung pada spesies ini memberikan perlindungan dari predator.

Melalui project ini kami akan melakukan deteksi dan identifikasi spesies kupu-kupu secara efisien dengan pre processing (meningkatkan kualitas gambar) dan melihat akurat melalui algoritma Machine learning sederhana.





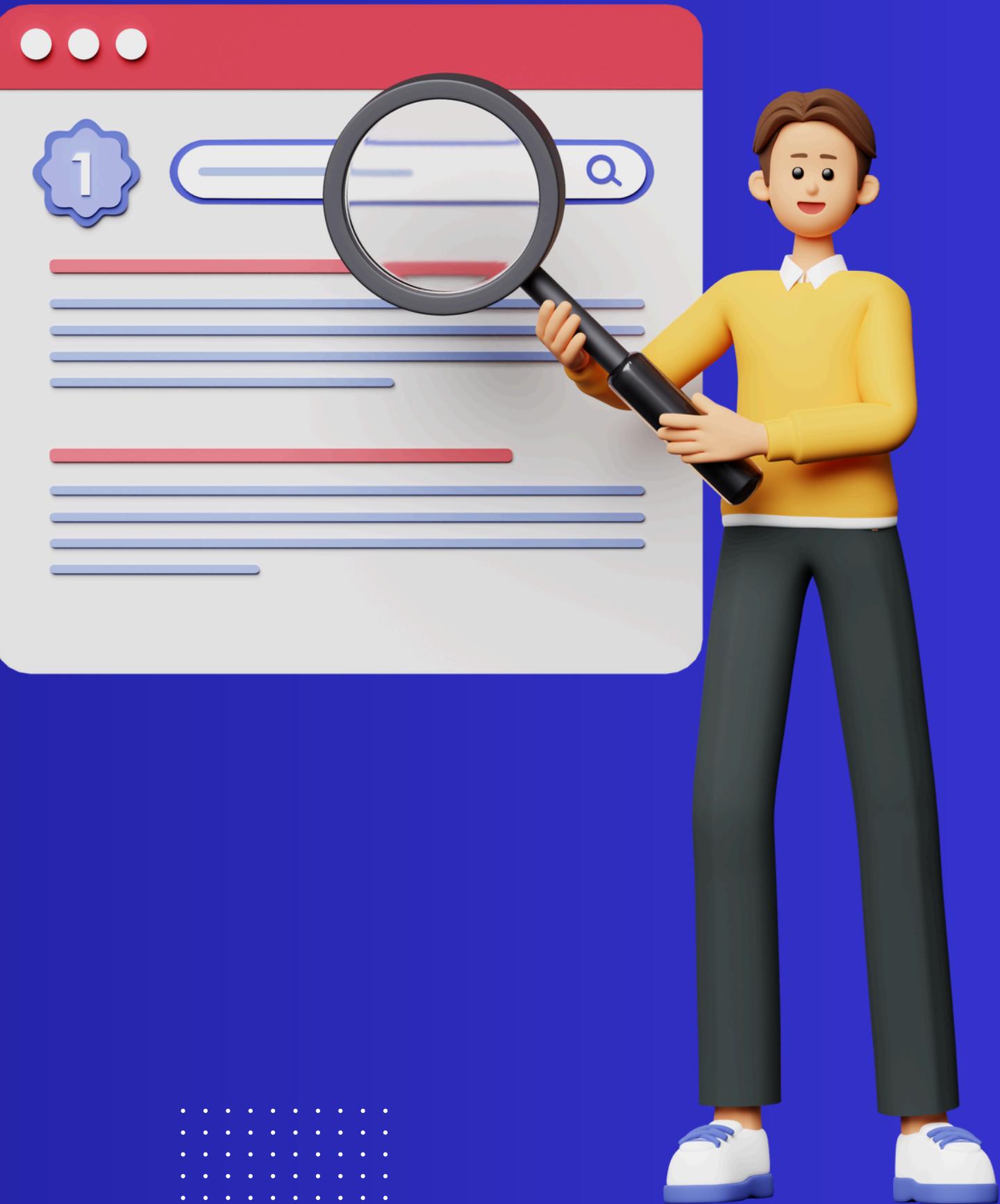
Deskripsi Program

ButterflyToxDetect adalah sistem pengolahan citra digital berbasis kecerdasan buatan yang dirancang untuk mendekripsi dan mengidentifikasi spesies kupukupu beracun secara akurat dan efisien. Program ini fokus pada tiga spesies utama: Monarch (*Danaus plexippus*), Pipevine Swallowtail (*Battus philenor*), dan Striated Queen (*Danaus gilippus*).

PENJELASAN DATASET & LABEL



- Dataset ini berisi kumpulan gambar kupu-kupu dari berbagai spesies yang berbeda bernama “Butterfly Classification Dataset”.
- Dataset terdiri dari 30 kelas/label utama, dan terdapat 3 kelas yang diklasifikasi yaitu:
 - a. Monarch
 - b. Pipevine Swallow
 - c. Strated Queen
- Total dataset terdiri dari 1.000 gambar kupu-kupu.



Tahapan Processing

GrayScale



Normalisasi



Ekualisasi Histogram



Median Filter



Modus Filter





GrayScale

```
for i in range(height):
    for j in range(width):
        gray = int(0.299 * img[i, j, 0] + 0.587 * img[i, j, 1] + 0.114 * img[i, j, 2])
        grayImg[i, j] = gray
```

GrayScale merupakan salah satu tahapan pengubahan bentuk gambar, dari yang mulanya berwarna atau 3D menjadi bentuk keabuan atau 2D. Hal ini dilakukan untuk mengurangi kompleksitas pengolahan gambar menjadi matriks 2 dimensi.





Normalisasi

Normalisasi adalah proses penskalaan dengan melakukan pengurangan resolusi citra untuk mengubah rentang intensitas piksel dalam gambar agar berada dalam rentang tertentu.

```
if min_val < 0 or max_val > 255:  
    img = np.clip(img, 0, 255)  
normalized_img = ((img - np.min(img)) / (np.max(img) - np.min(img))) * 255
```

Ekualisasi Histogram

Ekualisasi Histogram adalah suatu proses perataan histogram (mengubah derajat keabuan suatu piksel (r) dengan derajat keabuan yang baru (s) dengan suatu fungsi transformasi T, yang dalam hal ini $s = T(r)$), dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata.

$$h_i = \frac{n_i}{n}, i = 0, 1, \dots, L - 1$$

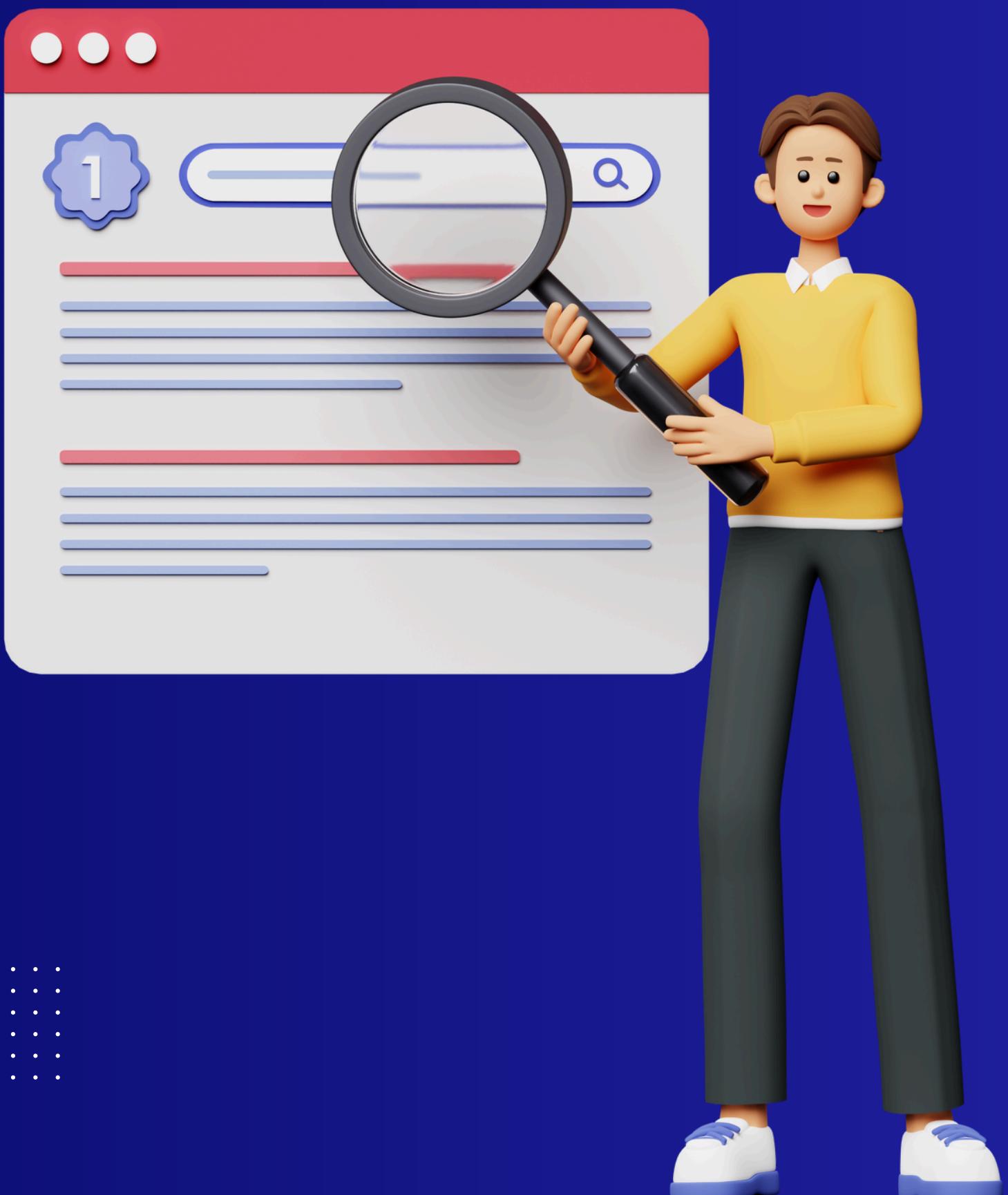


Median Filter

Median filter atau filter nilai tengah adalah bentuk penajaman melalui nilai tengah dari siatu citra. Tujuan utamanya adalah untuk memperbaiki kualitas gambar dengan meratakan intensitas piksel di sekitar setiap titik gambar. Sebagai tambahan, median filter membantu meningkatkan kualitas visual gambar tanpa mengorbankan informasi penting yang terkandung di dalamnya.

```
n = len(sub_image)
for p in range(n - 1):
    min_indeks = p
    for q in range(p + 1, n):
        if sub_image[q] < sub_image[min_indeks]:
            min_indeks = q
    sub_image_p = sub_image[p]
    sub_image[p] = sub_image[min_indeks]
    sub_image[min_indeks] = sub_image_p

median_index = kernel.shape[0] * kernel.shape[1]//2
result[i, j] = sub_image[median_index]
```



Modus Filter

Modus filter, atau mode filter, adalah teknik pengolahan citra digital yang digunakan untuk mengurangi noise dengan menggantikan setiap piksel dalam citra dengan nilai modus (nilai yang paling sering muncul) dari piksel di sekitarnya dalam jendela filter yang telah ditentukan. Modus filter sangat efektif untuk menghilangkan noise impulsif, seperti salt-and-pepper noise, karena memilih nilai yang paling sering muncul di sekitar piksel target.

```
temp_angka = [0] * 256
max_val, max_angka = -1, -1

for k in range(hKernel):
    for l in range(wKernel):
        val = img[i - half_hKernel + k][j - half_wKernel + l]
        temp_angka[val] += 1
        if temp_angka[val] > max_angka:
            max_angka = temp_angka[val]
            max_val = val

result[i][j] = max_val
```



Penjelasan Code Overall

- Sebelum Preprocessing**
- Sesudah Preprocessing**

Analisa

✓ **Akurasi Model**

Model yang dilatih dengan data yang telah diproses (grayscale, normalisasi, median filter, dan modus filter) menunjukkan peningkatan akurasi dibandingkan dengan model yang dilatih tanpa preprocessing

✓ **Kualitas Fitur**

Fitur yang diekstraksi dari gambar yang telah diproses lebih informatif dan konsisten, yang membantu model dalam belajar pola yang lebih jelas dan akurat.

✓ **Pemodelan test dan train**

Pemodelan test dan train menggunakan 3 model utama sebagai perbandingan yakni KNN, RF, SVC. Dari ketiganya dapat dilihat terjadi peningkatan akurasi didalamnya.





Kesimpulan

Dari percobaan yang telah dilakukan dapat disimpulkan bahwa percobaan dengan 5 pre processing berupa grayscale, normalisasi, median filter, dan modus filter adalah percobaan yang paling baik dengan peningkatan akurasi sebesar 10-20 %, dengan penggunaan metode tertinggi yakni RF.

Terima Kasih

