

PRAKTIKUM 1

OpenCV

Materi

- Instalasi OpenCV
- Kode program sederhana

Tujuan Praktikum

- Mahasiswa dapat melakukan instalasi OpenCV dan menjalankan program sederhana pada OpenCV

A. Penyajian

Computer Vision (CV) adalah salah satu cabang dari ilmu pengolahan citra digital (*digital image processing*) yang memungkinkan komputer memahami citra dunia nyata yang diperoleh. Berdasarkan pemahaman tersebut, program diharapkan dapat mengambil keputusan yang tepat. Beberapa kasus umum yang dapat diselesaikan dengan teknik CV adalah deteksi dan pengenalan wajah, pelacakan objek (*tracking*), navigasi, desain interaksi, dan proses manufaktur. Di Institut Pertanian Bogor, teknik CV juga diterapkan di bidang pertanian, antara lain untuk mengidentifikasi tanaman obat berdasarkan ciri morfologi dan tekstur, identifikasi penyakit pada tanaman pangan, penilaian keparahan penyakit, serta inspeksi kualitas buah dan sayuran. Sebagian besar teknik tersebut diterapkan dengan menggunakan pustaka OpenCV.

OpenCV (*Open Source Computer Vision*) adalah pustaka sumber terbuka yang dikembangkan untuk keperluan *computer vision*. OpenCV dikembangkan oleh Intel Corp dan bersifat *cross-platform*. OpenCV telah mendukung bahasa C, C++, Python, Java, dan Matlab.

OpenCV saat ini terdiri atas beberapa modul utama:

- **core**. Berisi definisi struktur-struktur data.
- **imgproc**. Berisi teknik pemroses citra seperti filter, transformasi, ruang warna, dan histogram.
- **video**. Berisi teknik analisis video seperti estimasi gerak, substraksi latar belakang, dan pelacakan objek.
- **calib3d**. Berisi teknik dasar untuk pengaturan *viewing window*, kalibrasi kamera, dan rekonstruksi objek 3D.
- **features2d**. Berisi teknik untuk ekstraksi fitur dan deskriptor.
- **objdetect**. Berisi teknik untuk deteksi objek dari kelas yang telah didefinisikan sebelumnya (seperti wajah dan mata).
- **highgui**. Berisi *interface* untuk akuisisi video dan gambar serta berisi fitur antarmuka sederhana.

- **gpu.** Berisi algoritme-algoritme yang lebih diefisiensikan dengan menggunakan *graphical processing unit* (GPU).
- dan modul pendukung lainnya.

Selain itu OpenCV juga dilengkapi dengan pustaka *Machine Learning* yang berisi algoritme seperti *naïve bayes*, *k-nearest neighbor*, *support vector machine*, *decision tree*, dan *neural network*.

Instalasi OpenCV

Untuk menggunakan OpenCV pada python harus menginstall beberapa *package* diantaranya adalah *opencv-python* dan *numpy*. Dalam menggunakan Bahasa *python* dapat menggunakan PyCharm sebagai *text editor* dan compilernya. Adapun langkah-langkah yang perlu dilakukan dalam menginstall OpenCV pada PyCharm adalah sebagai berikut:

- Buka PyCharm, setelah itu buka *file* kemudian pilih *setting*.
- Setelah jendela *setting* terbuka, pilih projek yang sedang dibuat, lalu pilih *project interpreter*.
- Pada jendela tersebut, terdapat *icon* “+” untuk menginstall *package-package* yang dibutuhkan. Pilih *icon* tersebut kemudian cara *package* yang dibutuhkan (*opencv-python* dan *numpy*), kemudian *install package* tersebut.

Kode Program Sederhana

Untuk menguji apakah sudah bisa atau tidak PyCharm kalian menggunakan OpenCV. Buatlah *project console* baru dalam bahasa Python, kemudian ketikkanlah Program 1.

```
#Program1
#ProgramOpenCVsederhanauntukmembacadanmenampilkancitradaun.

#Importpackagecv2untukopencv-pythondannumpy
importnumpyasnp          #untukmeringkasnumpymenjadinpagartidakterlalupanjang
importcv2

#Membacafilebernamadaun.jpgimage=c
v2.imread('daun.jpg')

#Menampilkan
citracv2.imshow("Gambar",image)

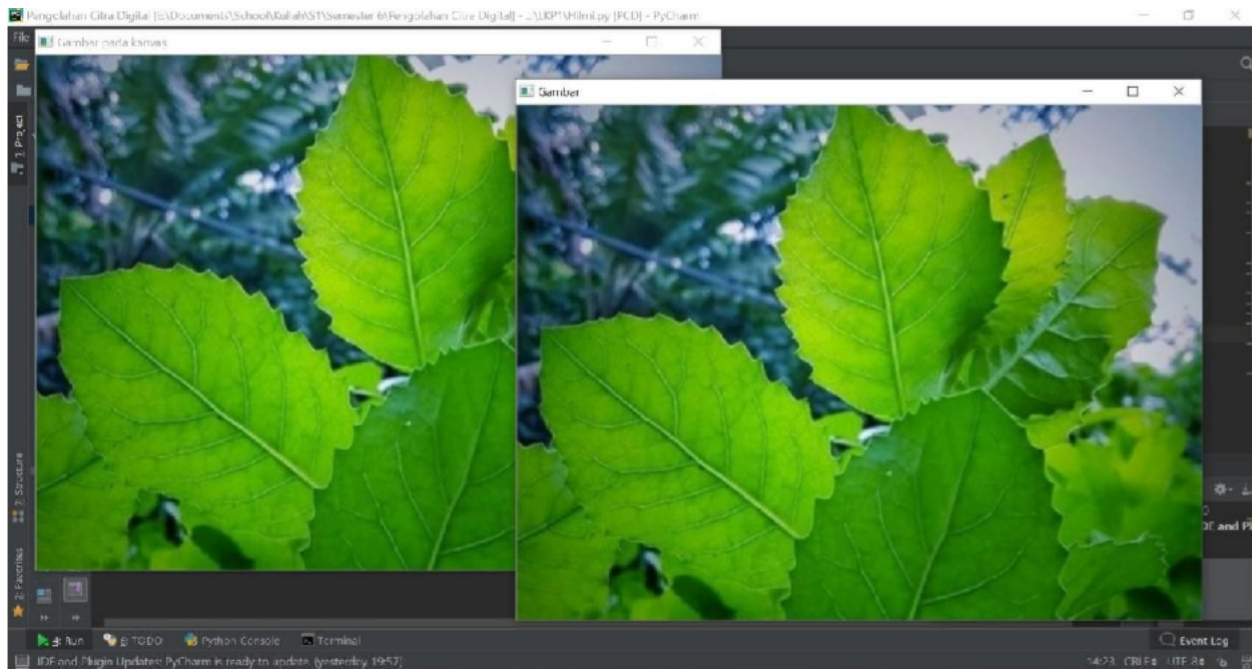
#Mengambilnilaidimensi (panjangdanlebar) danchannelcitrarow,col,ch
=
image.shape#Membuatkanv
askosong
kanvas=np.zeros((row,col,3),np.uint8)

#Menyimpannilaicitrapadakanvasforiinrange(
0,row):
    forjinrange(0,col):kanvas[i,j]
        =image[i,j]

#Menampilkan citra
padakanvascv2.imshow("Gambarpadakanvas",kanvas)

#Menghentikangambarhinggapenggunamenekantombolcv2.waitKey(0)
```

Sebelum menjalankan program, pastikan berkas gambar daun.jpg telah Anda salin pada direktori *project*. Setelah itu, *compile* dan jalankan *project* tersebut. Hasil keluaran program dapat dilihat pada Gambar 1.



Gambar 1 Keluaran Program 1 yang berupa citra daun

Penjelasan Kode Program

Bagian pertama program merupakan *package* yang dibutuhkan oleh program OpenCV standar.

```
import numpy as np
import cv2
```

Hal pertama yang dilakukan adalah membaca citra dengan menggunakan fungsi `cv2.imread()` ke dalam variabel `image`. Fungsi ini dapat membaca citra dalam berbagai format seperti BMP, DIB, JPEG, JPE, PNG, PBM, PGM, PPM, SR, RAS, and TIFF.

```
image = cv2.imread('daun.jpg')
```

Setelah itu, jika ingin langsung menampilkan citra yang sebelumnya sudah dibaca dapat dilakukan dengan menggunakan fungsi `cv2.imshow()`. Parameter pertama untuk nama citra tersebut sedangkan parameter kedua variabel yang menyimpan citra tersebut.

```
cv2.imshow("Gambar", image)
```

Jika ingin menampilkan sebuah citra pada suatu kanvas kosong hal pertama yang harus dilakukan adalah mengambil nilai dimensi citra (panjang dan lebar) dan channel yang ada pada citra tersebut agar canvas yang dibuat dimensinya sama dengan citra yang akan ditampilkan pada kanvas tersebut. Caranya buatlah tiga variabel untuk menampung nilai panjang, lebar, dan *channel* citra dari *shape* citra yang sudah

dibaca. Kemudian buat kanvas yang dimensi dan channelnya sama dengan citra yang ingin ditampilkan dimana isi tiap pixel kanvas tersebut bernilai 0 dengan menggunakan fungsi `numpy.zeros()`.

```
row,col,ch=image.shape
kanvas=np.zeros((row,col,3),np.uint8)
```

Untuk meletakkan nilai setiap pixel citra pada kanvas dapat dilakukan dengan cara *looping* setiap pixel pada citra dan letakkan pada nilai pixel pada kanvas. Untuk menampilkan kanvas yang berisi citra caranya sama dengan sebelumnya.

```
foriinrange(0,row):forj inrange(0,col):kanv
    as[i,j]
    =image[i,j]

#Menampilkan citra
padacanvascv2.imshow("Gambarpadakanvas",kanvas)
```

Fungsi ini akan meminta program untuk berhenti dan menunggu *keystroke*. Jika argumen yang diberikan positif, program akan menunggu selama milidetik argumen. Jika argumen diatur ke 0 atau ke angka negatif, program akan menunggu sampai suatu tombol ditekan.

```
#Menghentikangambarhinggapenggunamenekantombolcv2.waitKey(0)
```

Nama :Muhammad Farhadh
 NRP :G6401231080
 Nama Dosen :Dr. Ir. Yeni Aryati
 Nama Asisten :Quwwamulhaaq

B. Lembar Kerja Praktikum

1. Buatlah program sederhana untuk membuat matrix gambar

Kode:

```
# Nomor 1
kanvas1 = np.zeros((row, col, 3), np.uint8)
for i in range(row):
    for j in range(col):
        kanvas1[i, j] = image[i, j]
```

```
case '0':
    for i in range(10):
        for j in range(10):
            print(str(kanvas1[i, j]) + " ", end=" ")
        print("\n")
    exit()
```

Output:

```
[157 190 186] [156 189 185] [157 190 186] [159 192 188] [163 194 191] [164 195 192] [165 195 190] [164 194 189] [163 190 186] [163 190 186]
[155 188 184] [155 188 184] [156 189 185] [158 191 187] [162 193 190] [163 194 191] [165 194 191] [164 194 189] [163 190 186] [163 190 186]
[153 185 184] [153 185 184] [155 186 185] [157 188 187] [160 191 190] [162 193 192] [164 192 192] [163 192 189] [164 191 188] [164 191 188]
[150 181 182] [150 182 181] [152 182 183] [155 186 185] [158 188 189] [160 191 190] [162 190 191] [161 189 189] [163 189 189] [163 190 187]
[149 170 102] [140 170 179] [149 170 102] [152 102 103] [150 105 109] [160 100 109] [162 107 191] [161 106 100] [161 106 100] [161 107 107]
[147 176 181] [147 176 180] [147 176 181] [150 179 183] [155 181 187] [157 184 188] [159 183 189] [158 183 187] [158 183 185] [158 183 185]
[146 174 101] [140 174 100] [140 174 101] [150 176 102] [155 178 106] [157 101 107] [156 179 107] [155 179 105] [157 179 105] [157 179 104]
[147 172 182] [147 173 180] [147 172 182] [150 176 183] [154 177 185] [156 179 187] [155 178 186] [154 177 185] [154 175 183] [155 177 183]
[151 173 185] [151 173 184] [152 174 186] [154 176 187] [156 178 189] [156 178 189] [156 176 187] [153 173 184] [151 171 182] [152 173 181]
[150 172 184] [150 172 184] [151 173 185] [153 175 187] [154 176 187] [154 176 187] [154 174 185] [152 172 183] [150 170 181] [150 170 181]
```

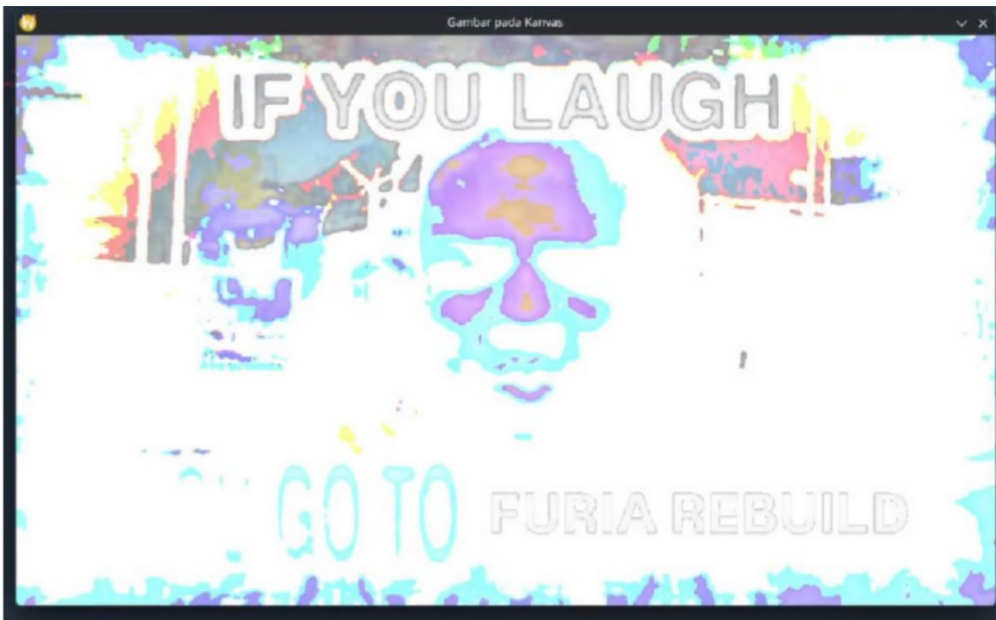


2. Buatlah program pencari nilai pixel (<150) dan mengganti dengan nilai 255.

Kode:

```
# Nomor 2
kanvas2 = kanvas1.copy()
for i in range(row):
    for j in range(col):
        for k in range(3):
            if (kanvas2[i, j, k] < 150):
                kanvas2[i, j, k] = 255
```

Output:



3. Buatlah program untuk membalik matrix (transpose)

Kode:

```
# Nomor 3
kanvas3 = np.zeros((col, row, 3), np.uint8)
for i in range(row):
    for j in range(col):
        kanvas3[j, i] = kanvas1[i, j]
```

Output:

