

MODUL 2

PENGOLAHAN CITRA DIGITAL
OPERASI ARITMATIKA DAN LOGIKA PADA CITRA

D3/D4 TEKNIK INFORMATIKA
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG



WILDAN SETYA NUGRAHA 032 | PENGOLAHAN CITRA
DIGITAL | JANUARI, 24 2023

```
In [ ]: pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.7.0.68-cp37-abi3-win_amd64.whl (38.2 MB)
Collecting numpy>=1.21.2
  Using cached numpy-1.24.1-cp311-cp311-win_amd64.whl (14.8 MB)
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.24.1 opencv-python-4.7.0.68
Note: you may need to restart the kernel to use updated packages.
```

```
In [ ]: pip install pillow
Collecting pillow
  Using cached Pillow-9.4.0-cp311-cp311-win_amd64.whl (2.5 MB)
Installing collected packages: pillow
Successfully installed pillow-9.4.0
Note: you may need to restart the kernel to use updated packages.
```

```
In [ ]: pip install numpy
Requirement already satisfied: numpy in d:\sekolah\kuliah\semester 4\pengolahan
citra digital\praktek\meeting 2\.venv\lib\site-packages (1.24.1)
Note: you may need to restart the kernel to use updated packages.
```

```
In [ ]: import cv2
from PIL import Image
import numpy as np
from IPython.display import display
```

```
In [ ]: def loadImage032(path):
    img = cv2.imread(path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    return img

def showImage032(img, text=""):
    display(Image.fromarray(img), text)

def addImage032(img1, img2):
    img = cv2.add(img1, img2)
    return img

def subImage032(img1, img2):
    img = cv2.subtract(img1, img2)
    return img

def maxImage032(img1, img2):
    img = cv2.max(img1, img2)
    return img

def minImage032(img1, img2):
    img = cv2.min(img1, img2)
    return img

def numpyInverseImage032(img):
    img = 255 - img
    return img

def cvInverseImage032(img):
```

```
    img = cv2.bitwise_not(img)
    return img

def cvResizeImage032(img1, img2):
    height, width, channel = img1.shape
    img2 = cv2.resize(img2, (width, height))
    return img2
```

```
In [ ]: foto = loadImage032("Images/foto.jpg")
label = loadImage032("Images/label.png")

showImage032(label)
showImage032(foto, "foto")
```

JTK 2021 - Wildan Setya Nugraha



'foto'

```
In [ ]: hasilAdd = addImage032(foto, label)
showImage032(hasilAdd)
```

```
-----
error                                         Traceback (most recent call last)
Cell In[6], line 1
----> 1 hasilAdd = addImage032(foto, label)
      2 showImage032(hasilAdd)

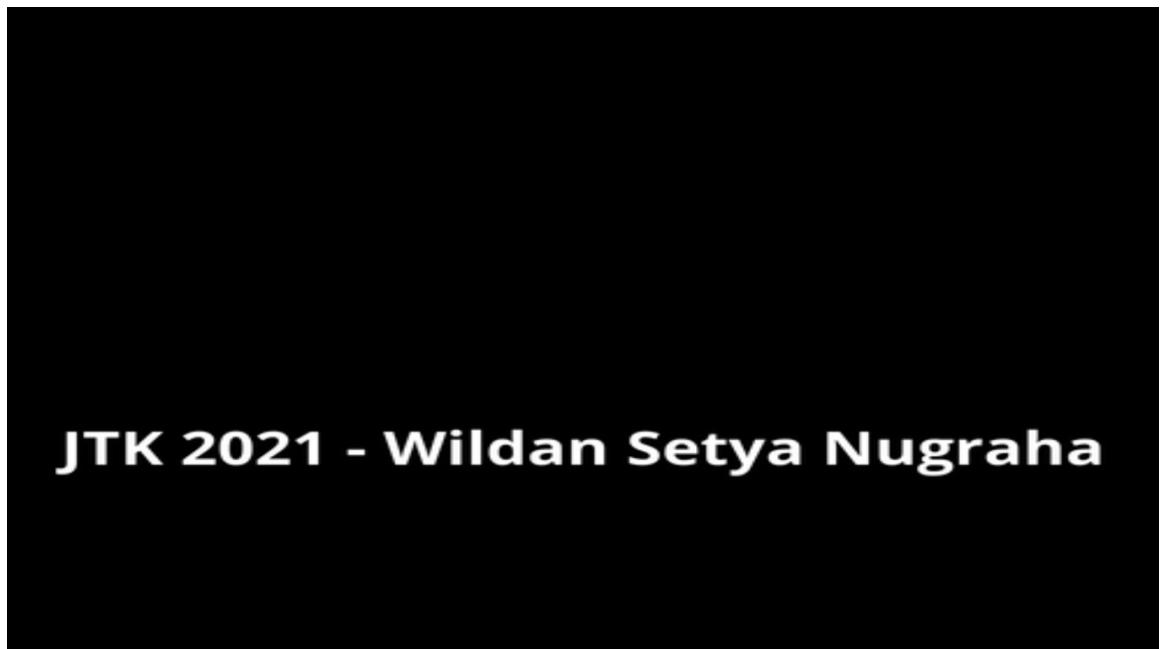
Cell In[4], line 10, in addImage032(img1, img2)
     9 def addImage032(img1, img2):
--> 10     img = cv2.add(img1, img2)
     11     return img

error: OpenCV(4.7.0) D:\a\opencv-python\opencv-python\opencv\modules\core\src\arithm.cpp:650: error: (-209:Sizes of input arguments do not match) The operation is neither 'array op array' (where arrays have the same size and the same number of channels), nor 'array op scalar', nor 'scalar op array' in function 'cv::arithm_op'
```

```
In [ ]: label = cvResizeImage032(foto, label)
hasilAdd = addImage032(foto, label)
showImage032(hasilAdd)
```

JTK 2021 - Wildan Setya Nugraha

```
In [ ]: labelInverse = cvInverseImage032(label)
showImage032(labelInverse)
```



In []: `showImage032(cvInverseImage032(label))`



In []: `hasilAdd = addImage032(foto, labelInverse)`
`showImage032(hasilAdd)`



```
In [ ]: showImage032(addImage032(foto, cvInverseImage032(label)))
```



```
In [ ]: showImage032(minImage032(foto, label))
# minImage ini akan membandingkan pixel per pixel
# dari dua buah image, Jika nilai pixel dari citra pertama
# lebih kecil dari citra kedua, maka nilai pixel pada citra
# hasil akan diambil dari citra pertama. Jika nilai pixel
# dari citra pertama lebih besar dari citra kedua, maka nilai
# pixel pada citra hasil akan diambil dari citra kedua.

showImage032(maxImage032(foto, labelInverse))
# maxImage ini akan membandingkan pixel per pixel
# dari dua buah image, jika nilai pixel dari citra pertama
# lebih kecil dari citra kedua, maka nilai pixel pada citra
# hasil akan diambil dari citra kedua. Jika nilai pixel dari
# citra pertama lebih besar dari citra kedua, maka nilai
# pixel pada citra hasil akan diambil dari citra pertama.
```

```
showImage032(subImage032(foto, labelInverse))
# subImage ini merupakan operasi pengurangan array,
# jika hasil pengurangan lebih dari nilai min yaitu 0
# maka nilai pixel pada citra hasil akan bernilai 0.
```





```
In [ ]: def bitwise_and032(img1, img2):
    img = cv2.bitwise_and(img1, img2)
    return img
# Maksud dari bitwise_and adalah akan menampilkan area yang beririsan,
# jadi operator ini akan membandingkan per pixel dari dua buah image
# nantinya pixel tersebut akan di bandingkan, jika kedua pixel tersebut
# bernilai 1 maka akan menghasilkan nilai 1 dan hal tersebut akan
# menghasilkan warna, jika salah satu bernilai 0 maka akan menghasilkan
# nilai 0 dan hal ini akan menghasilkan warna hitam.

def bitwise_or032(img1, img2):
    img = cv2.bitwise_or(img1, img2)
    return img
# Maksud dari bitwise_or adalah akan menampilkan semua area,
# jadi operator ini akan membandingkan per pixel dari dua buah image
# nantinya pixel tersebut akan di bandingkan, jika kedua pixel tersebut
# bernilai 1 atau pun 0 maka hasilnya akan 1 dan hal ini akan menghasilkan
# warna dimana jika warna yang dibandingkan adalah lebih putih maka yang
# akan tampil adalah warna putihnya.

def bitwise_xor032(img1, img2):
    img = cv2.bitwise_xor(img1, img2)
    return img
# Maksud dari bitwise_XOR akan menampilkan diluar area yang beririsan,
# jadi operator ini akan membandingkan per pixel dari dua buah image
# nantinya pixel tersebut akan di bandingkan, jika kedua pixel tersebut
# bernilai 1 maka akan menghasilkan nilai 0 dan hal ini akan menghasilkan
# warna hitam, jika salah satu bernilai 0 maka akan menghasilkan nilai 1
# dan hal ini akan menghasilkan warna. Pada hasil gambar yang dihasilkan
# ada warna biru, itu karena warna biru merupakan hasil dari perbedaan
# antara channel red and green. Namun hal itu akan terjadi ketika memiliki
# warna yang sama pada channel red dan green
```

```
In [ ]: showImage032(bitwise_and032(foto, label), "bitwise_and")
showImage032(bitwise_or032(foto, label), "bitwise_or")
showImage032(bitwise_xor032(foto, label), "bitwise_xor")
```



'bitwise_and'

JTK 2021 - Wildan Setya Nugraha



'bitwise_or'

'bitwise_xor'

In []:

```
# Membuka citra
img = loadImage032("Images/foto.jpg")

def pencahayaan032(img, light_factor):
    img = img * light_factor
    img[img > 255] = 255
    return img

showImage032(pencahayaan032(img, 2), "pencahayaan")

# Jadi perkalian dengan konstan ini sangat berguna ketika
# dalam situasi dimana citra asli terlalu gelap atau terang
# dan perlu diubah intensitas cahanya agar terlihat lebih baik.
# Jika nilai konstan yang digunakan adalah lebih dari 1 maka akan menghasilkan
# citra yang lebih terang, jika nilai konstan yang digunakan adalah kurang dari
# maka akan menghasilkan citra yang lebih gelap.
# Namun jika kita ingin menambahkan atau mengurangi
# intensitas cahaya pada citra dengan nilai yang berbeda pada
# setiap pixel maka kita bisa menggunakan operasi penjumlahan
# atau pengurangan pada array.
```



'pencahayaan'

Nama : Wildan setya Nugraha

Lesson Learn pertemuan 2

• What went well ?

Dari pertemuan kali ini ada beberapa hal yang dapat saya pelajari terutama dalam hal Operasi aritmatika.

1. min Image

ini akan membandingkan pixel per pixel dari dua buah image, jika nilai pixel dari citra pertama lebih kecil dari citra kedua, maka nilai citra yang diambil adalah citra pertama. begitupun sebaliknya.

2. max Image

ini akan membandingkan pixel per pixel dari dua image, jika nilai pixel dari citra pertama lebih besar dari citra kedua. maka nilai citra yang diambil adalah citra pertama. begitupun sebaliknya.

3. Add Image

Operasi penambahan array, jika hasil penambahan lebih dari nilai max yaitu 255, maka nilai pixel yang diambil pada citra adalah 255.

4. Sub Image

Operasi aritmatika pengurangan array, jika hasil pengurangan kurang dari 0, maka nilai citra adalah 0

5. Bitwise and

Akan menampilkan semua area yang beririsan, jadi operator ini akan membandingkan per pixel dari dua buah image, jika kedua pixel bernilai 1 maka akan menghasilkan warna jika salah satu 0 maka hasilnya 0 dan berwarna hitam.

6. Bitwise or

Akan menampilkan semua area. jadi jika kedua pixel dibandingkan bernilai 1 atau pun 0 maka hasilnya akan 1 dan berwarna. dan jika warna yang dibandingkan lebih putih maka yang akan tampil adalah warna putihnya

7. Bitwise XOR

Menampilkan diluar area yang beririsan jika kedua pixel bernilai 1 maka akan menghasilkan nilai 0 dan hal ini akan menghasilkan warna hitam.

jika salah satu bernilai 0 maka akan menghasilkan nilai 1 dan hal ini akan menghasilkan warna.

Pada hasil gambar ada yang berwarna biru, itu karena warna biru merupakan hasil dari perbedaan antara channel red and green. Namun hal itu akan terjadi ketika memiliki warna yang sama pada channel red dan green.

- What didn't go well ?

Improvisasi terhadap penggunaan syntax library masih banyak belum tereksplore.

- What might have been better handled if done differently ?

Kalau menurut saya lebih baik kita juga tahu apa yang terjadi di belakang layar saat proses aritmatika itu dilakukan dalam pengolahan citra digital.

- What recommendations would you give to others who might be involved in future projects of a similar type ?

1. Memahami warna dari 3 channel

R, G, B

2. Eksplor terkait library pengolahan citra digital!

3. Mengetahui proses perhitungan yang dilakukan dalam pengolahan citra digital