



MODUL PEMBELAJARAN PRAKTIK VISUAL ROBOT

FARIZ AL MUSTAQIM

KATA PENGANTAR

Puji syukur senantiasa dipanjatkan kehadirat Allah Yang Maha Esa karena atas berkat rahmat, hidayah serta karunianya sehingga penyusunan modul dapat diselesaikan dengan baik. Sholawat serta Salam senantiasa kita haturkana kepada junjungan kita Rasulullah Muhammad SAW, kepada Beliau kami harapkan syafaatnya. Tidak lupa pula penyusun haturkan terimakasih yang mendalam kepada Dosen Pembimbing yang sangat berperan membimbing dan mengarahkan penyusun, sehingga proses penyusunan modul dapat berjalan dengan lancar dan dapat diselesaikan dengan baik.

Adapun tujuan penyusunan modul ini adalah sebagai bahan ajar pada Mata kuliah Praktik Penginderaan Visual Robot. Materi yang dimuat dalam modul yaitu sistem pendeteksi Quick Response dan ArUco Marker.

Penyusun menyadari sepenuhnya bahwa media pembelajaran Pengembangan Aplikasi Scan QR Code dan Aruco Marker dengan OpenCV ini masih memiliki banyak kekurangan dan masih jauh dari kata sempurna. Untuk itu, penyusun sangat mengharapkan kritik, saran dan masukan yang membangun guna menyempurnakan modul ini. Penyusun sangat berharap modul ini dapat dipahami dan berguna dalam proses pembelajaran sebagai bahan ajar mata kuliah Praktik Penginderaan Visual Robot. Semoga media pembelajaran modul ini dapat bermanfaat dan menjadi referensi baru bagi pembaca.

Kami ucapkan terimakasih atas dukungan, bimbingan dan masukan yang membangun dari semua pihak sehingga modul ini dapat terselesaikan dan digunakan pada pembelajaran di lingkup Prodi Pendidikan Teknik Mekatronika, Fakultas Teknik, Universitas Negeri Yogyakarta.

Yogyakarta, 22 Agustus 2022

Penulis,

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI.....	4
DAFTAR GAMBAR	5
PENDAHULUAN	6
A. Digital Image Processing.....	6
B. Deteksi Objek	7
C. Jupyter Notebook.....	8
D. Python.....	9
E. OpenCV	10
F. Quick Respon Code	10
G. Aruco Marker.....	11
Langkah Penggunaan	12
1. Penggunaan Jupyter Notebook	12
2. Pengenalan Library OpenCV.....	19
3. Pengenalan <i>Quick Response</i>	25
4. Pengenalan Aruco Marker	33
Daftar Pustaka	40

DAFTAR GAMBAR

Gambar 1. Deteksi Objek	8
Gambar 2. Logo Jupyter Notebook	9
Gambar 3. Syntax Python	9
Gambar 4. Logo OpenCV	10
Gambar 5. Quick Respon	11
Gambar 6. Aruco Marker	11

PENDAHULUAN

A. Digital Image Processing

Digital image processing merupakan teknik pengolahan digital yang dapat membantu kita dalam manipulasi gambar digital dengan menggunakan komputer. Karena data raw (mentah) dari sensor pencitraan dari platform satelit mengandung kekurangan, untuk mengatasi kekurangan tersebut dan mendapatkan keaslian informasi, maka dia harus melalui berbagai tahap pemrosesan. Perlu untuk diketahui bahwa 3 (tiga) tahapan umum yang harus dilalui semua jenis data saat menggunakan teknik digital adalah pre-processing (pra-pemrosesan), enhancement (peningkatan) dan display (tampilan), dan information extraction (ekstraksi informasi).

Berdasarkan penjelasan pengertian image processing serta tujuan, dasar, cara kerja dan jenisnya di atas, sekarang kita pasti sudah dapat menyebutkan bahwa ini adalah salah satu metode yang berkembang untuk mengubah gambar menjadi bentuk digital. image processing juga dapat melakukan beberapa operasi untuk mendapatkan model tertentu atau mengekstrak informasi yang berguna darinya.

Macam-macam fase serta steps atau tahapan yang terjadi dalam image processing antara lain sebagai berikut:

1. *Acquicition* (akuisisi)

Fase ini bisa sesederhana ketika kita mendapatkan gambar dalam bentuk digital. Dalam tahapan ini, pekerjaan utamanya meliputi scaling (penskalaan) dan color conversion atau konversi warna, sebagai contoh misalnya RGB ke Gray atau sebaliknya.

2. *Enhancement* atau peningkatan

Fase ini adalah salah satu tahapan yang paling sederhana serta paling menarik di bidang image processing, dimana ini juga digunakan untuk mengekstrak beberapa detail tersembunyi dari sebuah gambar dan bersifat subjektif.

3. *Restoration* atau pemulihan

Tahapan ini juga berhubungan dengan daya tarik gambar, tetapi bersifat objektif, dimana pemulihannya didasarkan pada model matematika atau probabilistik atau degradasi gambar.

4. *Color image processing* (pengolahan warna gambar)

Fase ini berkaitan dengan Pseudocolor dan model warna pemrosesan gambar penuh warna berlaku untuk pemrosesan gambar digitalnya.

5. *Wavelets* dan multi-resolution processing

Ini adalah dasar untuk merepresentasikan gambar dalam berbagai derajat.

6. *Compression* (kompresi)

Tahap ini melibatkan dalam mengembangkan beberapa fungsi untuk melakukan operasi ini. Ini terutama berkaitan dengan ukuran atau resolusi gambar.

7. *Morphological processing* (pengolahan morfologi)

Fase ini berkaitan dengan alat untuk mengekstraksi komponen gambar yang berguna dalam representasi & deskripsi bentuk.

8. *Segmentation procedure*

Tahapan ini termasuk mempartisi gambar menjadi bagian atau objek penyusunnya. Perlu kalian ketahui bahwa segmentasi otonom, biasanya adalah tugas tersulit dalam image processing.

9. *Representation* (representasi) dan *description* (deskripsi)

Ini mengikuti tahap output (keluaran) segmentasi. Pemilihan representasi hanyalah bagian dari solusi untuk mengubah data raw (mentah) menjadi data yang sudah diproses.

10. *Object detection* dan recognition

Tahapan ini adalah proses yang akan memberikan label ke suatu objek berdasarkan deskriptornya.

B. Deteksi Objek

Menurut Aningtiyas et al. (2020: 423) deteksi objek merupakan teknik *Computer*

Vision untuk dapat menemukan contoh objek pada citra digital dari suatu gambar atau video. Algoritma yang digunakan dalam mendeteksi objek memanfaatkan pembelajaran mesin mendalam untuk dapat menghasilkan hasil yang baik. Gollapudi (2019: 97) mengemukakan pendapat bahwa mendeteksi objek dalam gambar adalah kemampuan penting dari aplikasi *Computer Vision*. Deteksi / pengenalan objek banyak diaplikasikan pada pelabelan adegan, navigasi robotik, mobil tanpa pengemudi, pengenalan bagian wajah dan tubuh, deteksi penyakit dan kanker, objek dalam citra satelit, pengenalan tulisan tangan, dan banyak lagi.



Gambar 1. Deteksi Objek

C. Jupyter Notebook

Jupyter Notebook adalah aplikasi web gratis untuk yang digunakan untuk membuat dan membagikan dokumen yang memiliki kode, hasil hitungan, visualisasi, dan teks. Jupyter adalah singkatan dari tiga bahasa pemrograman Julia (Ju), Python (Py), dan R. Tiga bahasa pemrograman ini adalah sesuatu yang penting bagi seorang data scientist.



Gambar 2. Logo Jupyter Notebook

Jupyter berfungsi untuk membantu kamu dalam membuat narasi komputasi yang menjelaskan makna dari data di dalamnya dan memberikan insight mengenai data tersebut. Selain itu, Jupyter juga mempermudah kerja sama antara insinyur dan data scientist karena kemudahannya dalam menulis dan berbagi teks dan kode.

D. Python

Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, Python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain.

Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. Bahasa Python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya.

Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah sibuk mencari syntax error.

```
In [1]: print("Hello World")
Hello World
```

Gambar 3. Syntax Python

Hanya dengan menuliskan kode print seperti yang diatas, anda sudah bisa mencetak apapun yang anda inginkan di dalam tanda kurung (). Dibagian akhir kode pun, anda tidak harus mengakhirnya dengan tanda semicolon (;).

E. OpenCV

OpenCV merupakan *library*/pustaka berbasis *Open Source* yang memiliki lebih dari 500 fungsi yang ditujukan untuk menangani *Computer Vision*. “*Computer Vision* sendiri adalah bidang yang berhubungan dengan upaya untuk membuat komputer dapat menangkap informasi yang berada dalam suatu gambar atau video”(Kadir, 2019, p. 2).

OpenCV adalah pustaka yang berupa perangkat lunak yang dapat diunduh di <https://opencv.org/>. Menurut situs resmi OpenCV, library/pustaka ini telah diunduh lebih dari 14 juta kali pada Oktober 2018 dan terus bertambah seiring berjalannya waktu.



Gambar 4. Logo OpenCV

F. Quick Respon Code

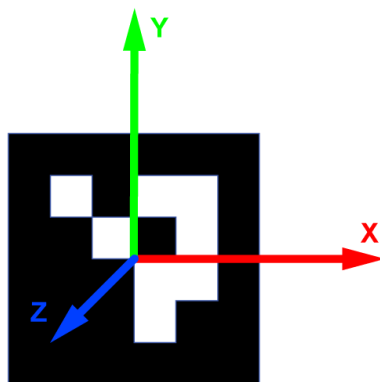
QR Code atau singkatan dari *Quick Response* adalah sebuah gambar dalam bentuk matriks dua dimensi di mana data dapat disimpan (Wijaya & Gunawan, 2016). Popularitas *QR Code* berkembang pesat di seluruh dunia. Saat ini, ponsel dengan kamera bawaan banyak digunakan untuk mengenali *QR Code* (Tiwari, 2017). *QR Code* bisa digunakan dalam berbagai hal dalam hal menyimpan informasi, seperti contoh ketika ingin mengakses suatu *website* kita tidak perlu mengetikkan *url* *website* tersebut melainkan bisa dengan cukup melakukan *scan* maka selanjutnya *smartphone/scanner* akan langsung menampilkan alamat *url* yang berada pada *QR Code* tersebut.



Gambar 5. Quick Respon

G. Aruco Marker

Aruco Marker adalah *square marker* sintetis yang terdiri dari batas hitam lebar dan matriks biner bagian dalam yang menentukan pengenalnya (id). Aruco Marker biasa digunakan dalam implementasi *AR (Augmented Reality)*. Aruco Marker memiliki kegunaan menyimpan informasi koordinat yang nantinya digunakan untuk kalibrasi kamera. Menurut (Nayak, 2020, p. 1) Aruco Marker adalah fiducial marker yang ditempatkan pada objek atau pemandangan yang dicitrakan. Ini adalah kotak biner dengan latar belakang dan batas hitam dan pola yang dihasilkan putih di dalamnya yang secara unik untuk diidentifikasi. Batas hitam membantu membuat deteksi Aruco Marker lebih mudah. Aruco Marker dapat dihasilkan dalam berbagai ukuran.

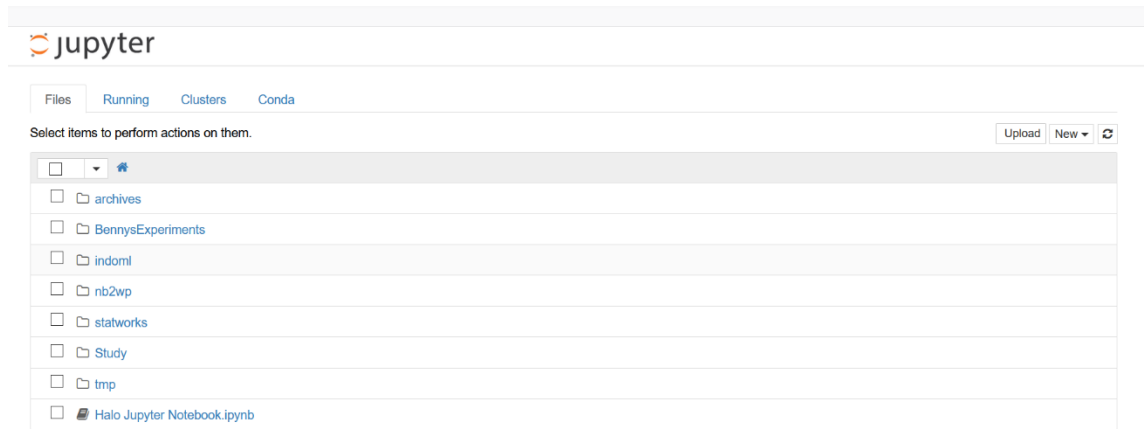


Gambar 6. Aruco Marker

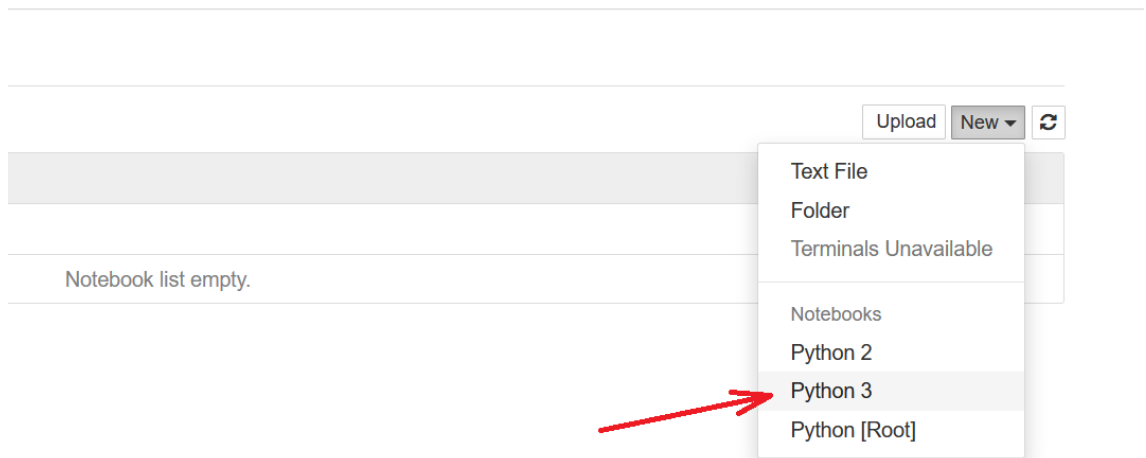
Langkah Penggunaan

1. Penggunaan Jupyter Notebook

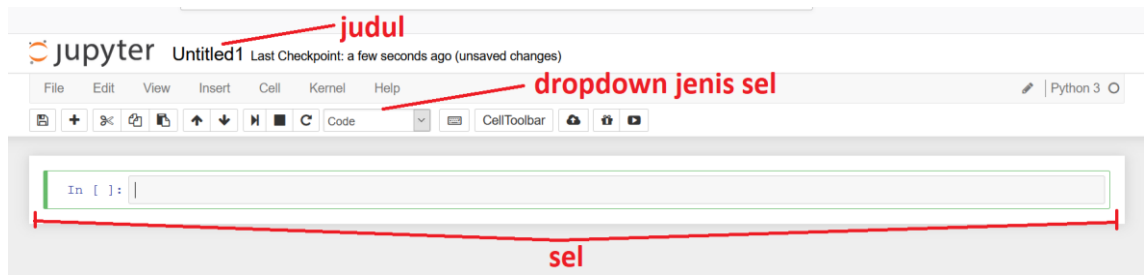
a. Membuat Notebook baru



b. Klik tombol New, lalu pilih Python 3 seperti pada gambar di bawah ini:



c. Menyunting Notebook



Judul

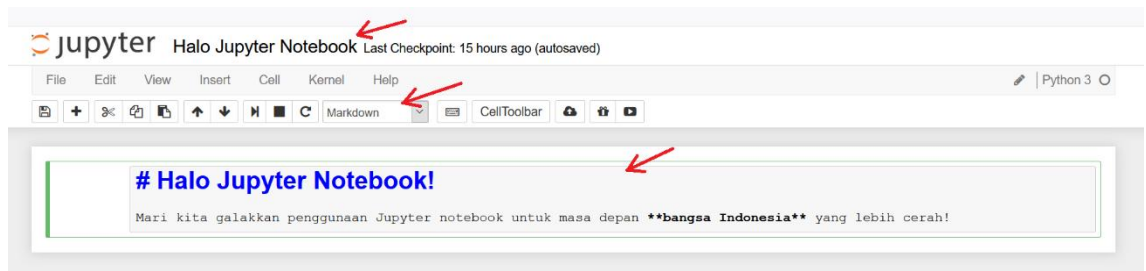
Yang pertama harus Anda sunting adalah judul. Masukkan judul dengan mengklik tulisan *Untitled* di atas. Harap diperhatikan bahwa judul ini akan menjadi nama file notebooknya juga.

Sel dan Jenis Sel

Setelah itu, bagian penting yang Anda harus ketahui adalah *sel* (Inggris: *cell*) (lihat gambar di atas). *Sel* adalah tempat Anda menuliskan karya Anda, baik berupa narasi ataupun kode. Nantinya notebook akan berisi banyak sel. Saat ini notebook hanya mempunyai satu sel. Jenis *sel* Anda pilih dengan memilih pada *dropdown* jenis sel seperti yang ditunjukkan pada gambar. Pada awalnya, jenis *sel* adalah kode.

d. Menulis Narasi dengan Markdown

Untuk menulis narasi, gantilah jenis sel menjadi **Markdown** dari pilihan *dropdown* jenis cell di atas. Lalu tulislah judul dan narasi awal untuk karya Anda, misalnya seperti ini:



Sintaks Markdown akan dijelaskan di bawah, tapi kalau Anda sekarang penasaran Anda bisa melihat detilnya di laman [Markdown cheatsheet](#).

Ketika kita menyunting sel seperti ini, maka sel disebut sedang dalam **mode edit**. Perhatikan garis kotak hijau yang mengelilingi sel. Pada mode ini, untuk sel

Markdown kode sumber Markdownnya akan ditampilkan.

Untuk menampilkan hasil tulisan Markdown kita itu, sel harus dijalankan.

e. Menjalankan *Cell*

Untuk melihat tampilan sebuah sel, maka sel itu harus kita jalankan, dengan salah satu cara berikut (kalau selnya masih belum mendapatkan fokus, maka kliklah sekali agar dia fokus):

Cara Menjalankan dari Menu	Kunci Pintas	Keterangan
Menu <i>Cell</i> -> <i>Run Cells</i>	Ctrl+Enter (atau Cmd+Enter untuk Mac)	Menjalankan sel
Menu <i>Cell</i> -> <i>Run Cells and Select Below</i>	Shift+Enter	Menjalankan sel lalu memindahkan fokus ke sel berikutnya
Menu <i>Cell</i> -> <i>Run Cells and Insert Below</i>	Alt+Enter (atau Option+Enter untuk Mac)	Menjalankan sel lalu membuat sel baru di bawahnya

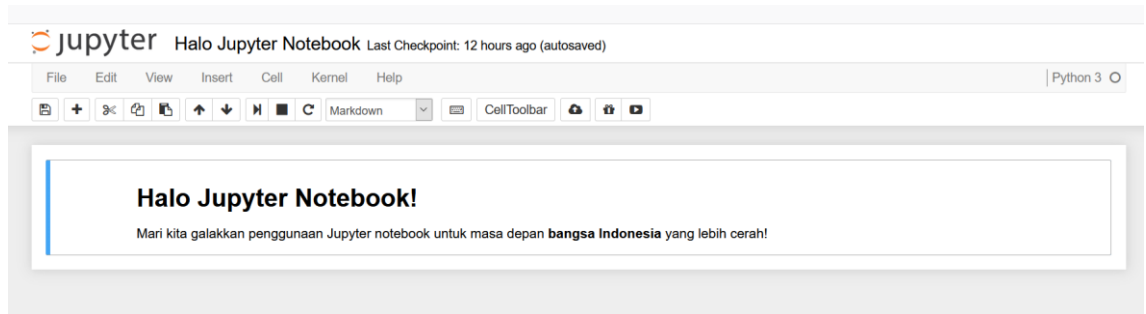
Menjalankan sel mempunyai makna yang berbeda tergantung jenis selnya. Untuk jenis sel Markdown, maka ini akan merubah kode Markdown itu menjadi tampilan yang diharapkan (dirender). Untuk jenis sel kode, maka kodenya akan dijalankan oleh kernel, perubahan *state* akan disimpan oleh kernel, dan hasilnya ditampilkan di notebook.

Sebagai tambahan informasi, ada cara untuk menjalankan semua sel yang bisa Anda lihat di menu *Cell*:

- Run All
- Run All Above
- Run All Below

f. Menjalankan Sel Markdown

Mari kita eksekusi sel Markdown kita di atas, dan hasilnya akan seperti ini:



Setelah sel dieksekusi, maka sel disebut ada dalam *mode command*. Perhatikan garis kotak biru yang mengelilingi sel.

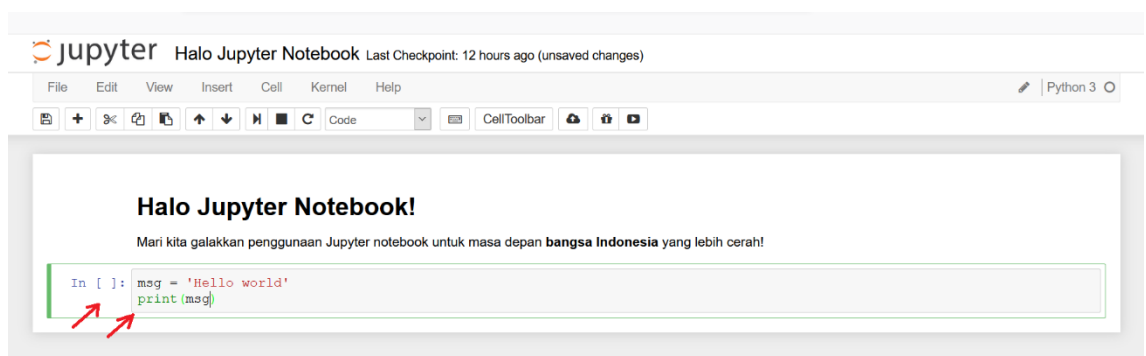
g. Menyunting Kembali

Ketika sel dalam *mode command*, untuk menyunting kembali di *mode edit* Anda bisa mengklik ganda sel itu atau mengklik sekali untuk menjadikan sel itu fokus lalu menekan Enter.

Selanjutnya mari kita membuat sel baru bertipe kode. Tambahkan sel baru dengan memilih menu *Insert* → *Insert Cell Below*, atau dengan menekan tombol + dari toolbar. Secara default, jenis sel baru adalah kode jadi kita tidak perlu merubah jenisnya.

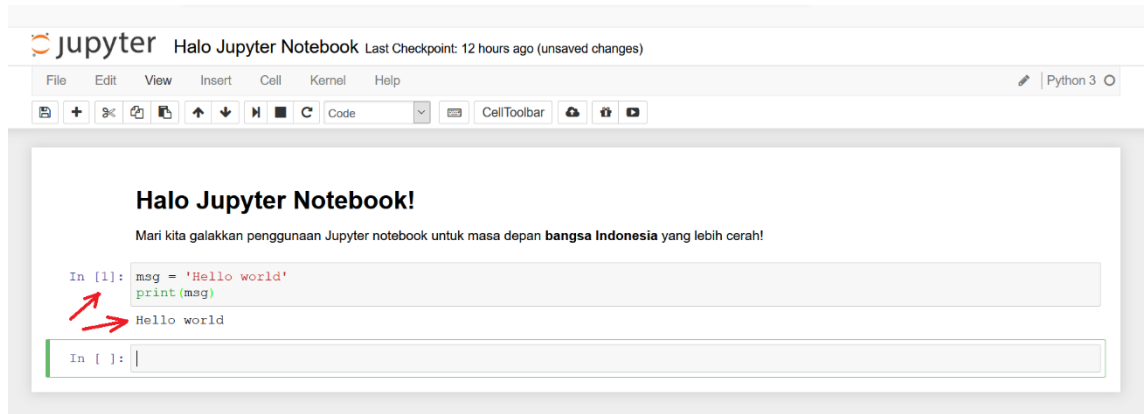
h. Menulis dan Menjalankan Kode

Mari kita sunting sel kode seperti di bawah ini:



Perhatikan juga bahwa untuk jenis sel kode, ada braket kotak di sebelah kiri sel, bertuliskan “In []:”. Braket kotak ini nanti akan diisi angka urutan eksekusi setelah selnya dieksekusi.

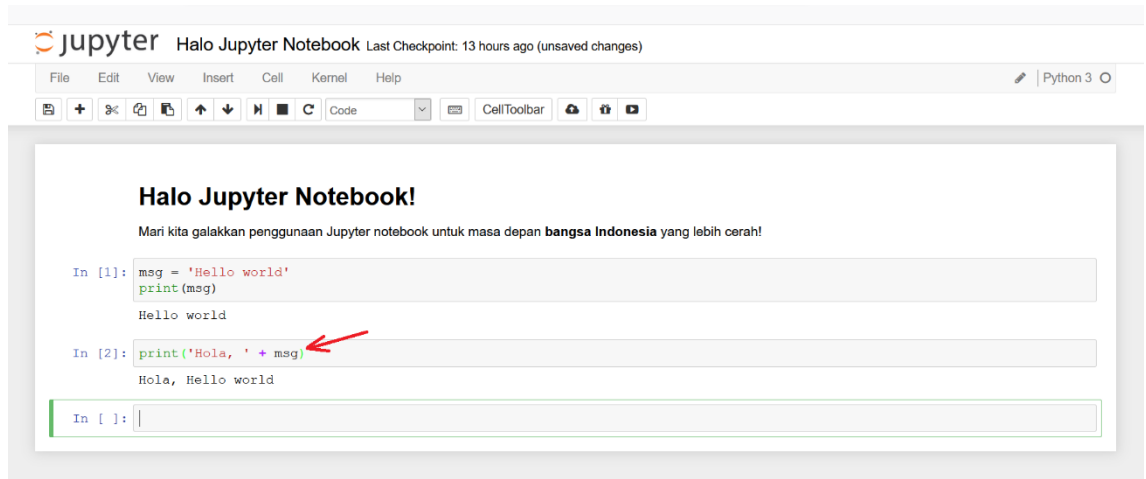
Mari kita jalankan. Hasilnya akan seperti ini:



Yaaa.. kodenya dijalankan. Perhatikan braket kotaknya sekarang terisi nomor urutan eksekusi (“In [1]:”). Nomor ini berguna karena urutan eksekusi tidak harus sama dengan urutan sel dalam notebook. Misalnya Anda punya lima sel, katakanlah sel 1 2 3 4 5, maka bisa saja Anda mengeksekusinya dengan urutan 1 4 2 5 3. Hal ini mungkin terdengar aneh sekarang. Tapi pada prakteknya ini merupakan hal yang sangat biasa Anda lakukan nantinya. Seperti kalau kita menyunting dokumen atau kode, maka sering kita menyunting bagian-bagian secara melompat-lompat. Makanya nomor urutan eksekusi ini sangat berguna untuk diperhatikan.

i. Lingkup (Scope) Variabel

Variabel-variabel dan fungsi-fungsi yang Anda deklarasikan di sel akan mempunyai lingkup global, dan bisa diakses dari sel lainnya. Misalnya, mari kita membuat sel baru yang mengakses variabel `msg` yang kita deklarasikan di sel kedua tadi (ingat, sel pertama kita adalah Markdown):

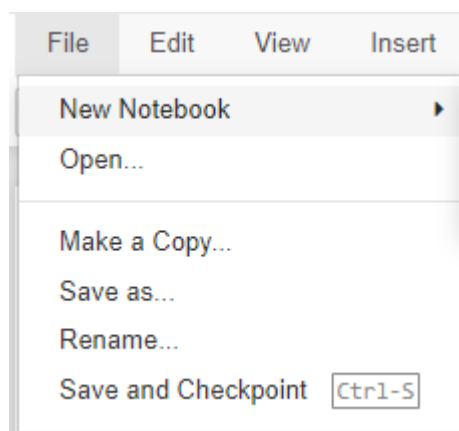


Anda bisa mengakses variabel global yang didefinisikan di sel lain, asalkan sel tempat variabel tersebut didefinisikan telah dieksekusi sebelumnya. Cara melihatnya adalah dengan melihat nomor eksekusi dalam braket kotak di sebelah kiri sel itu. Karena nomor eksekusinya lebih kecil dari nomor sekarang (nomor eksekusi sel itu 1, dan nomor sekarang 2), dan kernelnya sama, maka berarti sel tersebut telah tereksekusi sebelumnya.

j. Menyimpan Notebook

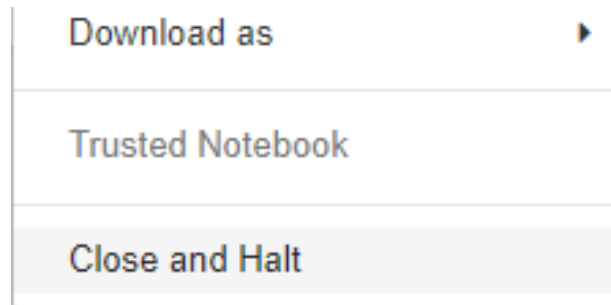
Untuk menyimpan notebook, Anda bisa melakukan salah satu di bawah ini:

- tekan kunci `Ctrl+S` (atau `Cmd+S` di Mac)
- pilih dari menu **File** -> **Save and Checkpoint**
- tekan tombol disket dari toolbar



k. Menutup Notebook

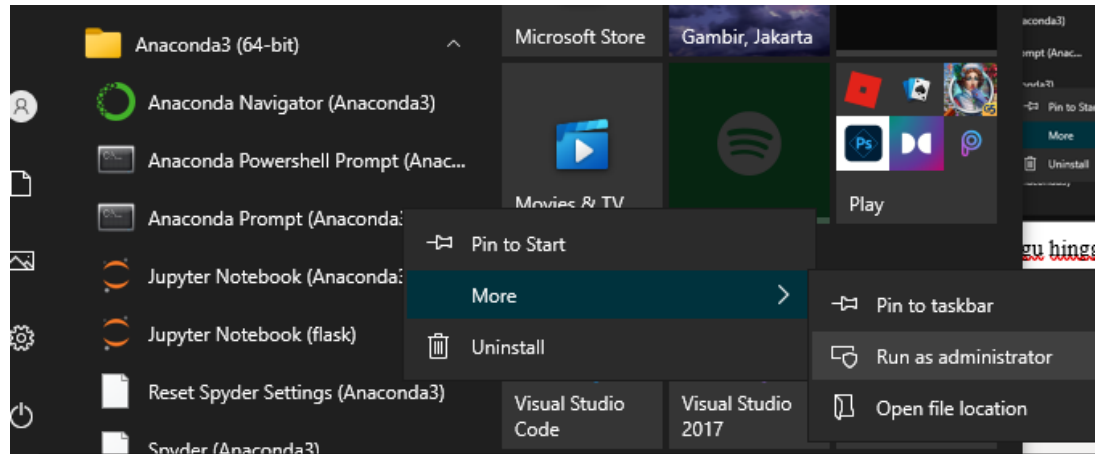
Untuk menutup notebook, pilih dari menu **File** → **Close and Halt**. Kadang-kadang window/tab browser tidak tertutup setelah Anda melakukan itu. Anda bisa menutup tab/window browser secara manual.



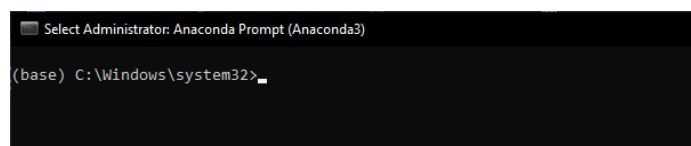
2. Pengenalan Library OpenCV

a. Instalasi OpenCV-Contrib

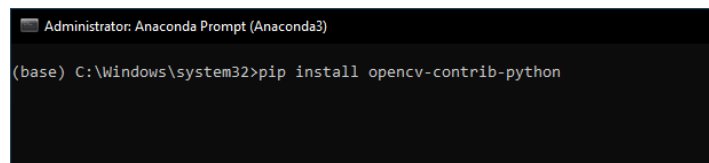
- 1) Silahkan buka **Anaconda Prompt** pada Windows bar lalu **Run as administrator** seperti berikut.



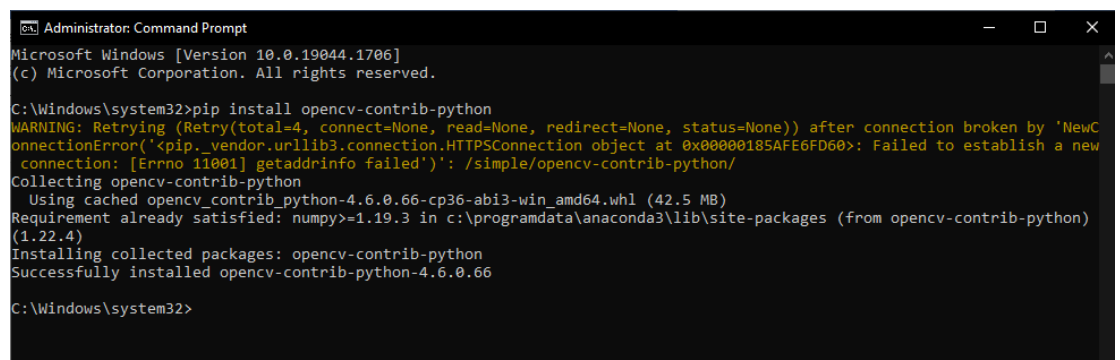
- 2) Apabila muncul Pop-Up **User Control** klik **Yes** untuk melanjutkan.
- 3) Kemudian akan terbuka Command Prompt dari Anaconda.



- 4) **Pastikan Terkoneksi dengan Internet.**
- 5) Lalu ketik pada Prompt seperti berikut: **pip install opencv-contrib-python**



- 6) Akan muncul keterangan seperti berikut apabila pernah menginstall



- 7) Lalu cek dengan mengetikan script **python** seperti berikut

```
Command Prompt
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Fariz AM>python
```

8) Ketik lagi script **import cv2**

9) Apabila sudah terinstal dengan sempurna maka tidak akan muncul error seperti berikut

```
Command Prompt - python
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Fariz AM>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

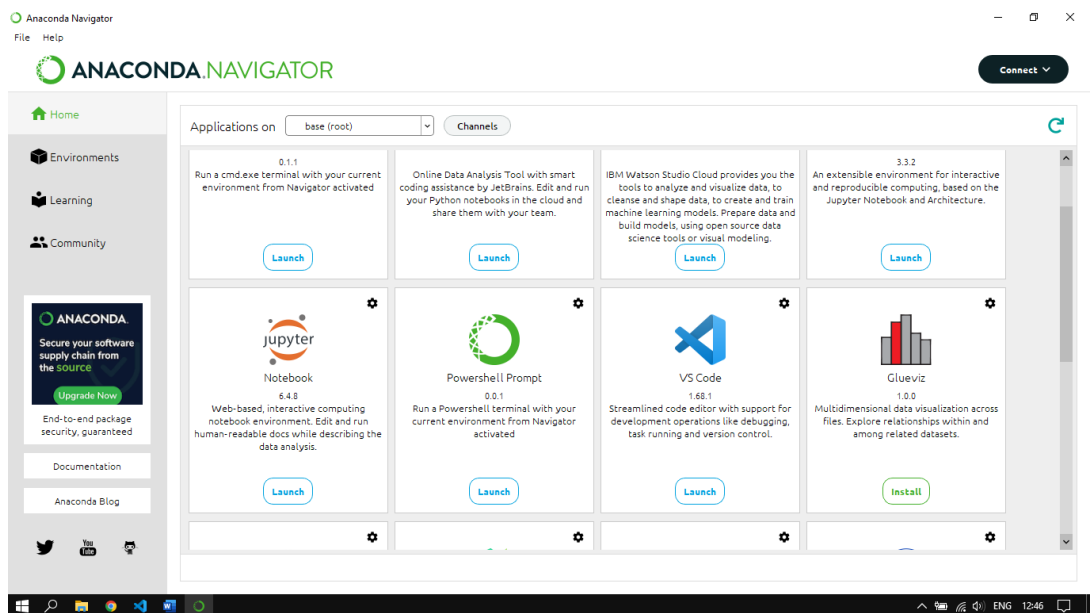
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>>
```

b. Uji Coba kode Program pada OpenCV

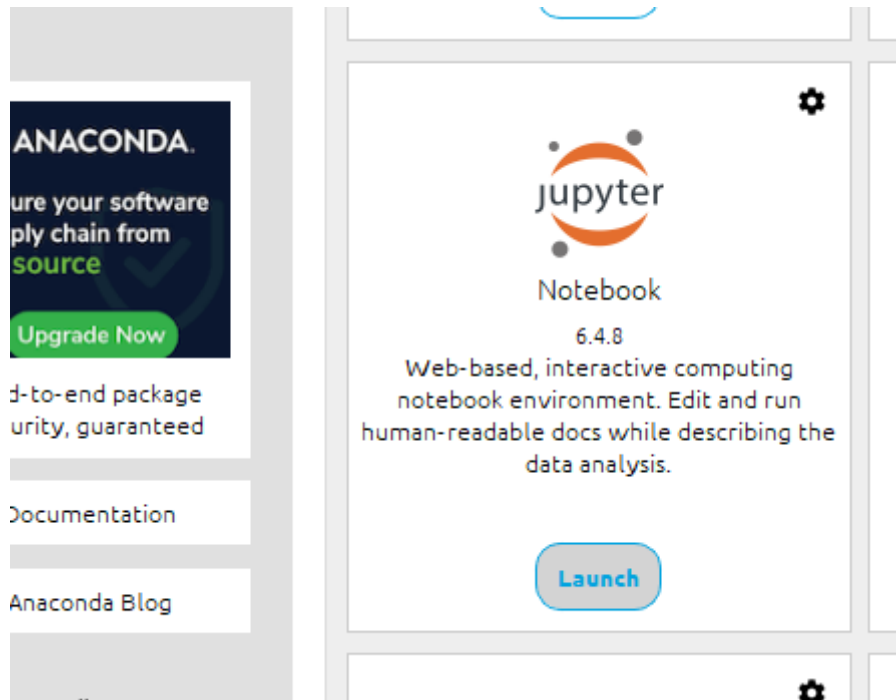
1) Persiapkan gambar yang ingin diuji coba dengan OpenCV, misal ambil contoh gambar pada link berikut.

<https://faunafocus.files.wordpress.com/2021/01/rusty-spotted-cat.png>

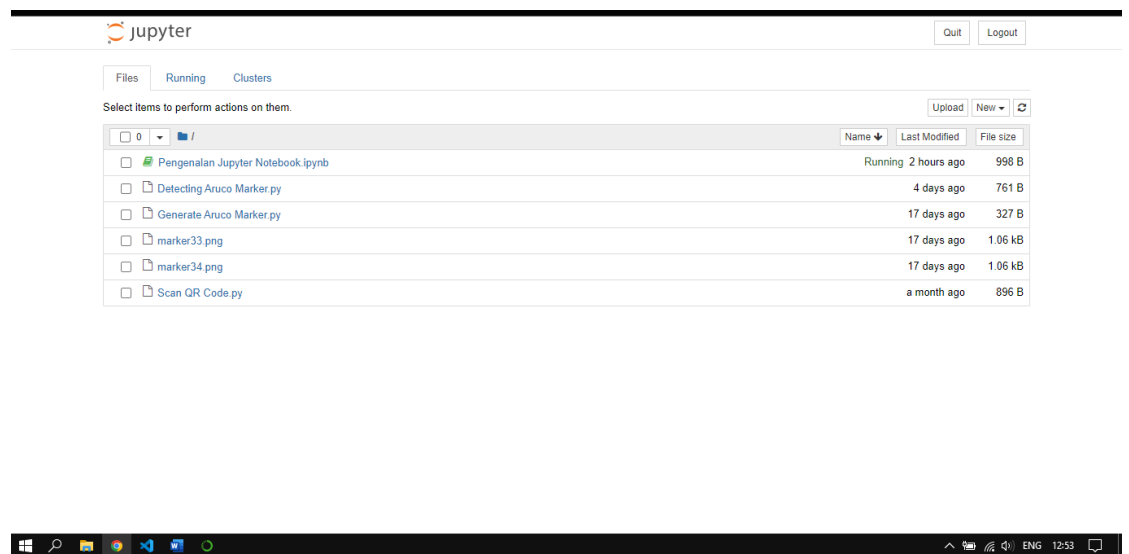
2) Jalankan Anaconda Navigator as Administrator



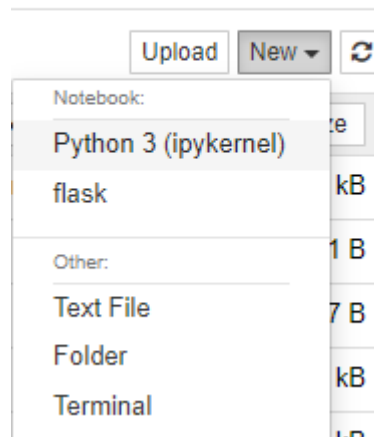
3) Untuk memulai pemrograman Python dengan Jupyter Notebook, silahkan Launch Jupyter Notebook



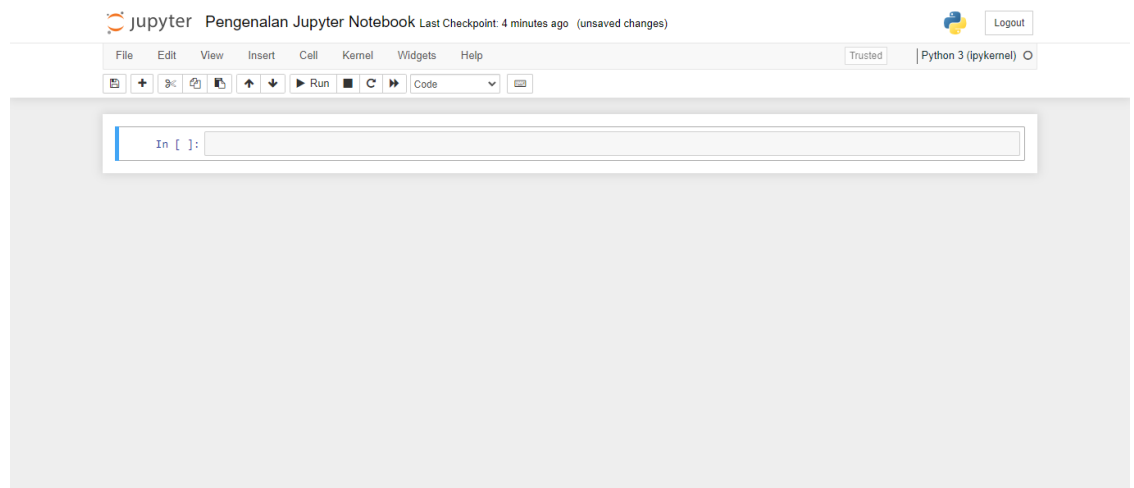
- 4) Setelah di **Launch** maka akan menuju browser default yang di gunakan pada masing-masing device.



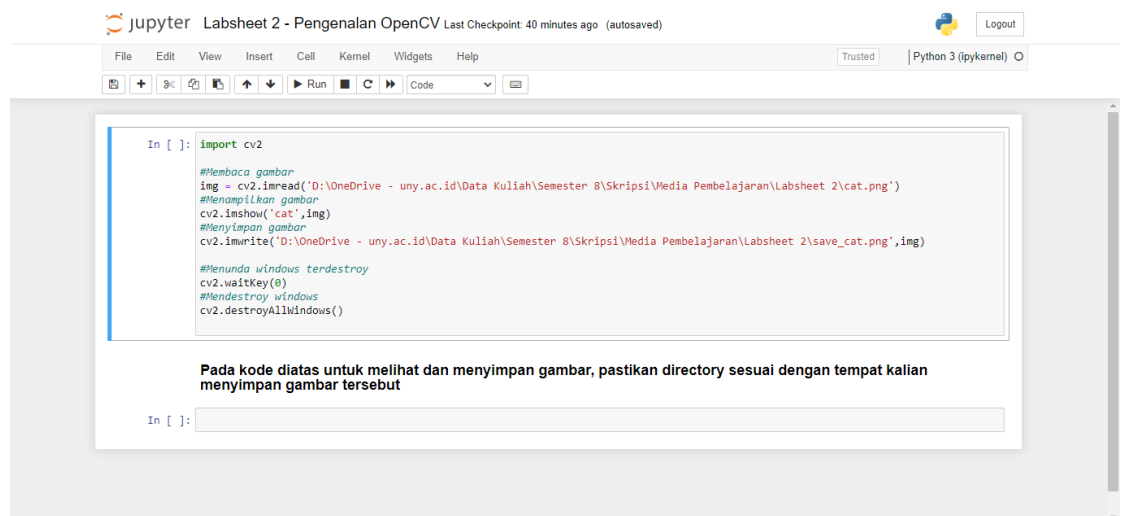
- 5) Untuk memulai menjalankan Script Python, silahkan klik tombol **New** dan arahkan ke Python 3 seperti gambar berikut.



- 6) Lalu jendela baru akan terbuka untuk membuka text editor kernel Jupyter Notebook



- 7) Kemudian ketika kode program berikut:



import cv2

```

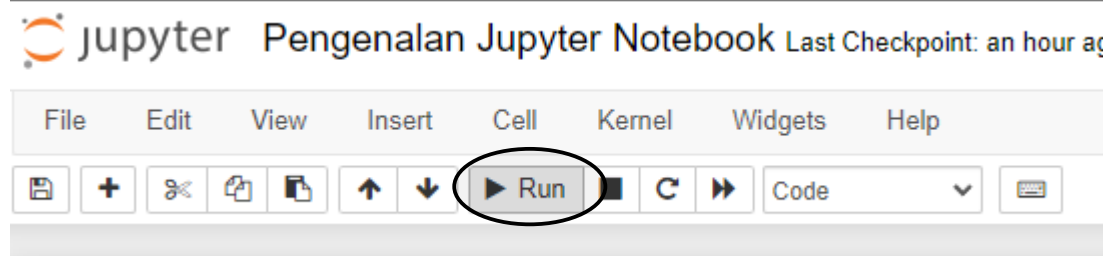
#Membaca gambar
img = cv2.imread('D:\OneDrive - uny.ac.id\Data Kuliah\Semester
8\Skripsi\Media Pembelajaran\Labsheet 2\cat.png')
#Menampilkan gambar
cv2.imshow('cat',img)
#Menyimpan gambar
cv2.imwrite('D:\OneDrive - uny.ac.id\Data Kuliah\Semester 8\Skripsi\Media
Pembelajaran\Labsheet 2\save_cat.png',img)

#Menunda windows terdestroy
cv2.waitKey(0)
#Mendestroy windows
cv2.destroyAllWindows()

```

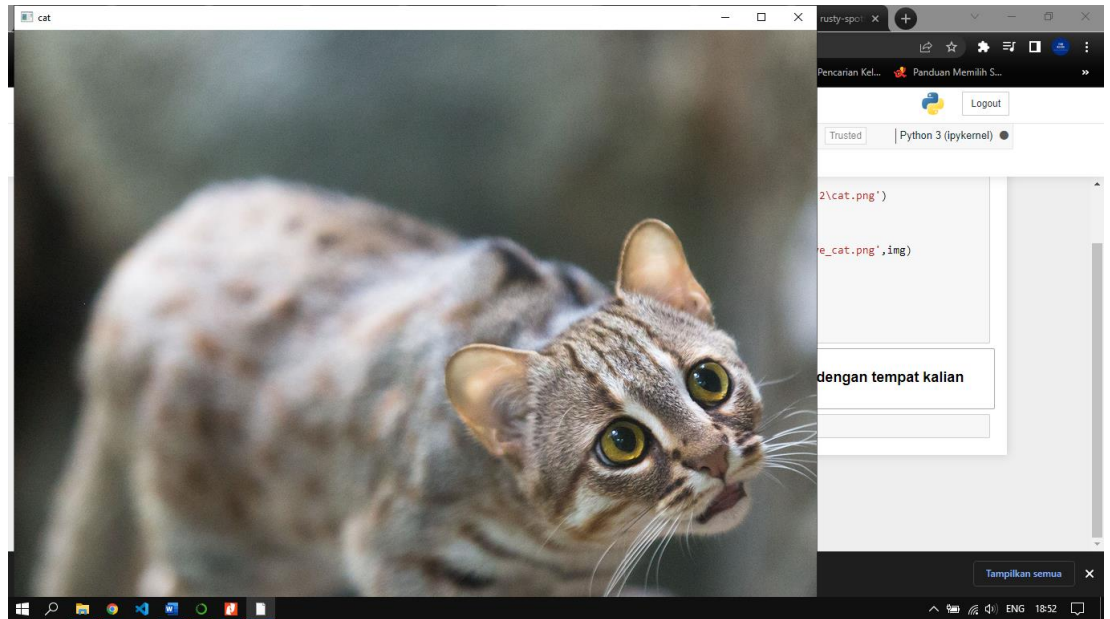
Note: Pada kode diatas untuk melihat dan menyimpan gambar, pastikan directory sesuai dengan tempat kalian menyimpan gambar tersebut.

- 8) Klik **Run** pada Jupyter Notebook.

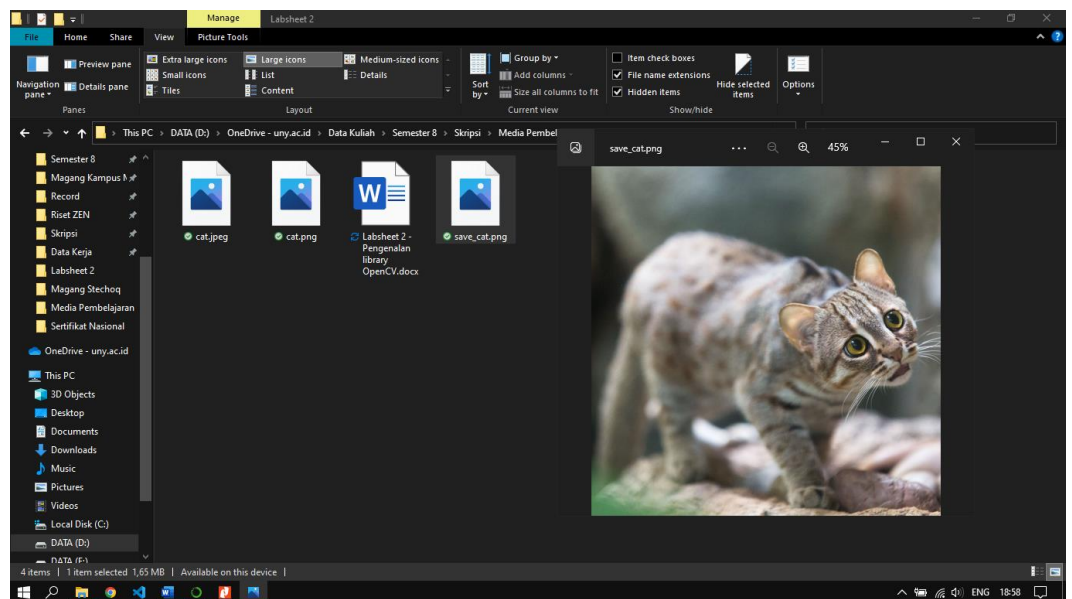


- 9) Maka akan keluar hasil pemrosesan seperti berikut: Terdapat pop-up windows gambar yang ditampilkan dan gambar yang disimpan menggunakan kode Python!

- a) Pop Up Window pembacaan image cat.png



b) Gambar yang disimpan dengan nama save_cat.png



10) Berikut keterangan *syntax* perintah pada kode diatas:

a) Membaca Gambar (Read Image)

Fungsi pada OpenCV untuk membaca gambar dengan perintah:

cv2.imread()

Contoh *syntax*:

```
cv2.imread('cat.png')
```

b) Menampilkan Gambar (Show Image)

Fungsi pada OpenCV untuk menampilkan gambar dengan perintah:
cv2.imshow()

Contoh:

```
cv2.imshow('cat.png')
```

c) Menyimpan Gambar (Save Image)

Fungsi pada OpenCV untuk menyimpan gambar dengan perintah:
cv2.imwrite()

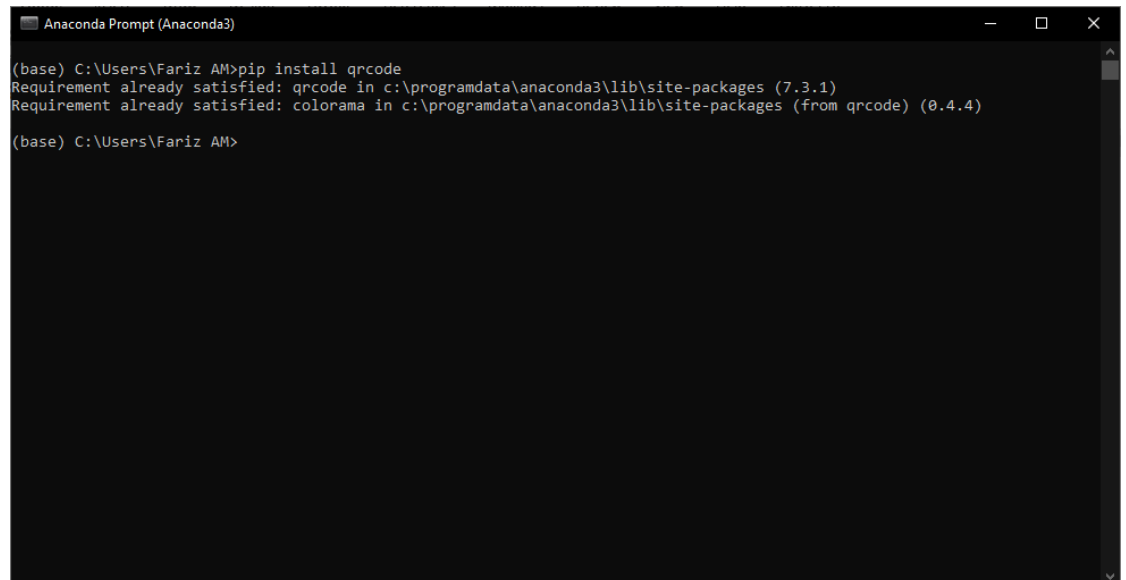
Contoh:

```
cv2.imwrite('save_cat.png', 'cat.png')
```

3. Pengenalan *Quick Response*

c. Install library qrcode

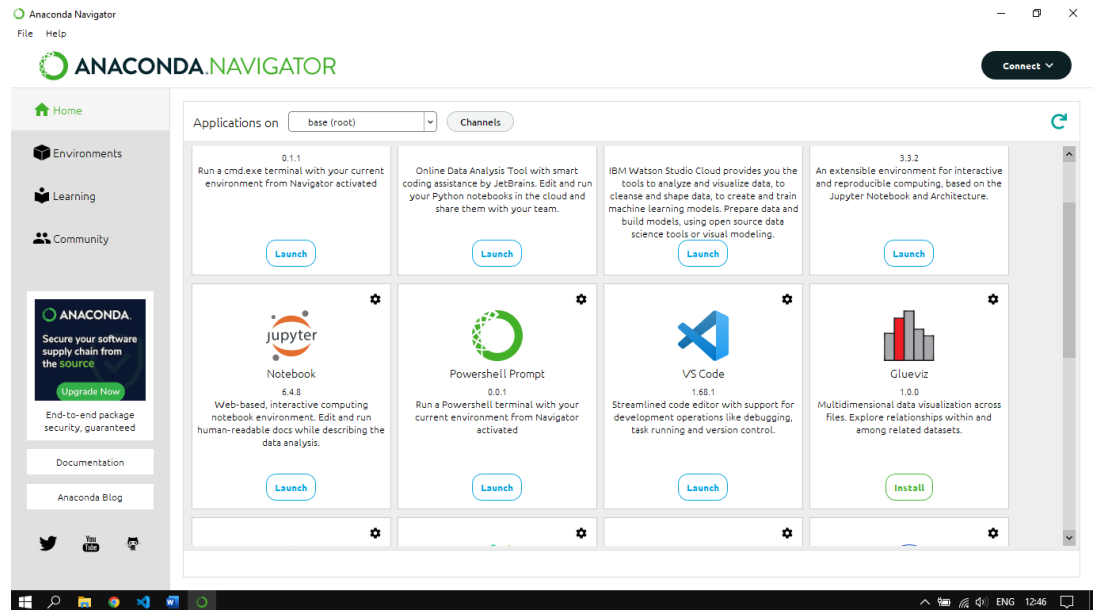
Pertama buka anaconda command prompt lalu ketik pip **install qrcode**



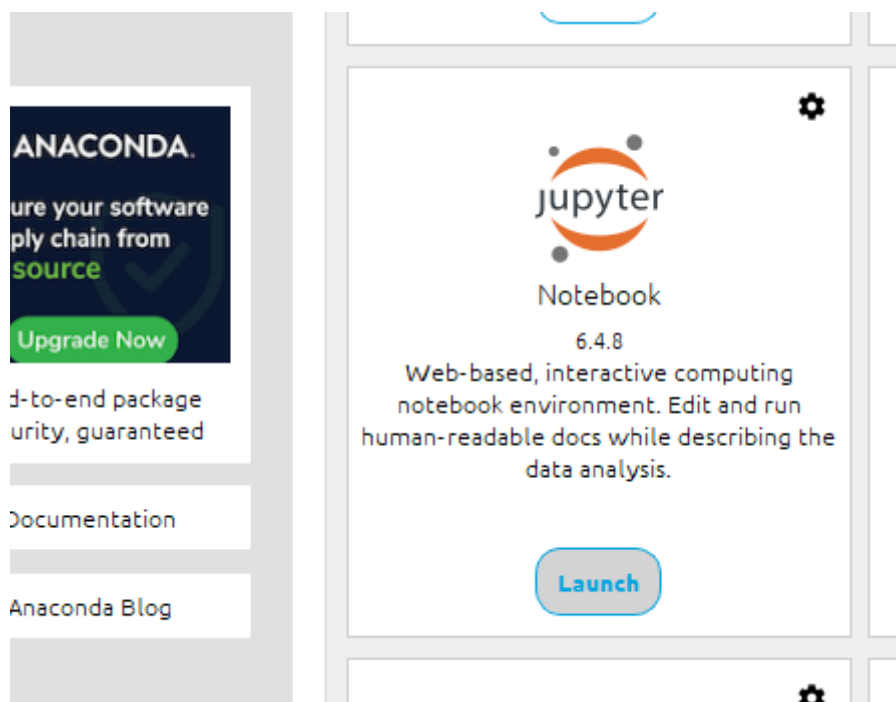
```
Anaconda Prompt (Anaconda3)
(base) C:\Users\Fariz AM>pip install qrcode
Requirement already satisfied: qrcode in c:\programdata\anaconda3\lib\site-packages (7.3.1)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from qrcode) (0.4.4)
(base) C:\Users\Fariz AM>
```

d. Mengimport library yang dibutuhkan

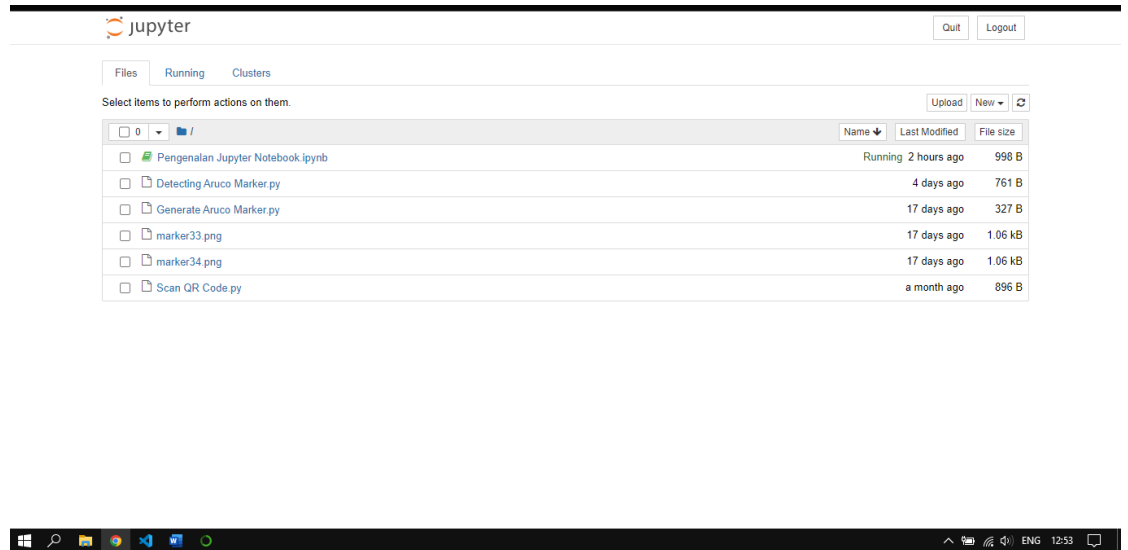
- 1) Jalankan Anaconda Navigator as Administrator**



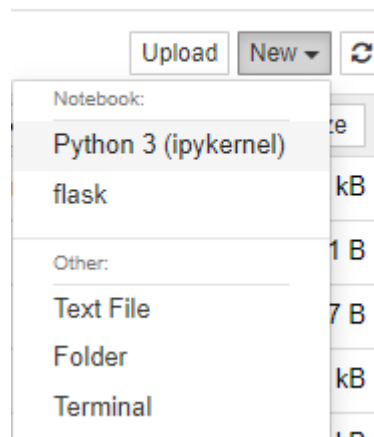
- 2) Untuk memulai pemrograman Python dengan Jupyter Notebook, silahkan Launch Jupyter Notebook



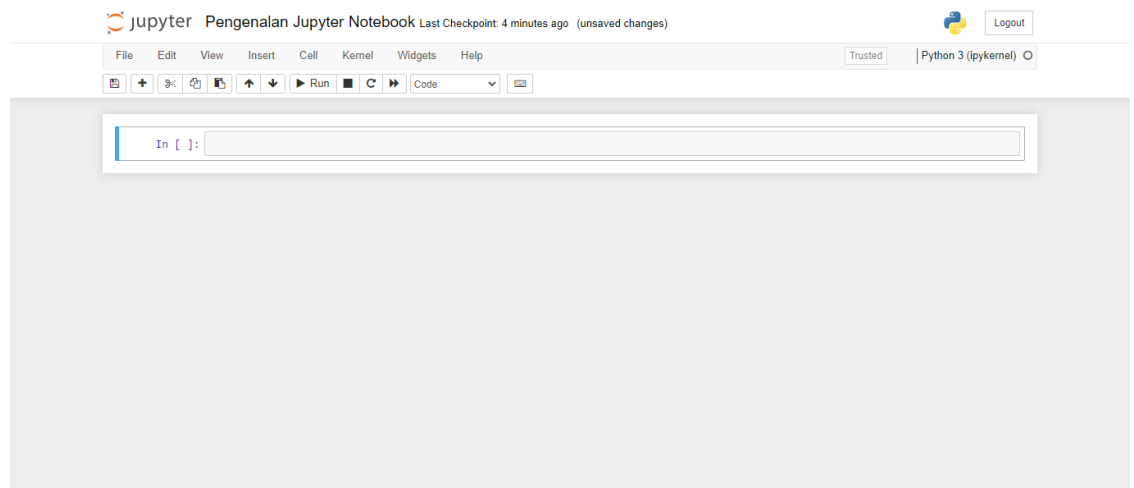
- 3) Setelah di **Launch** maka akan menuju browser default yang di gunakan pada masing-masing device.



- 4) Untuk memulai menjalankan Script Python, silahkan klik tombol **New** dan arahkan ke Python 3 seperti gambar berikut.



- 5) Lalu jendela baru akan terbuka untuk membuka text editor kernel Jupyter Notebook



- 6) Kemudian import library berikut:

Import Library

```
: import cv2
import numpy as np
import qrcode
from pyzbar.pyzbar import decode
```

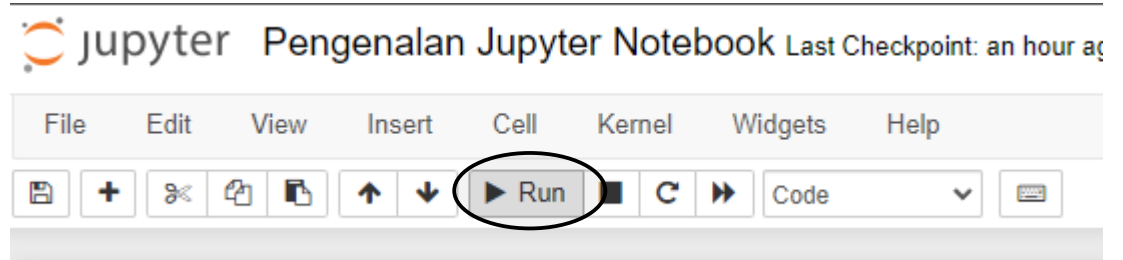
import cv2

import numpy as np

import qrcode

from pyzbar.pyzbar import decode

- 7) Klik **Run** pada Jupyter Notebook.



e. Mengenerate QR Code

- 1) Ketikan kode berikut untuk mengenerate/membuat qrcode dari sebuah link situs.

Generate QR Code

```
:  
# Link for website  
  
input_data = "https://uny.ac.id"  
  
#Creating an instance of qrcode  
qr = qrcode.QRCode(  
    version=1,  
    box_size=10,  
    border=5)  
qr.add_data(input_data)  
  
qr.make(fit=True)  
img = qr.make_image(fill='black', back_color='white')  
img.save('qrcode001.png')
```

Link for website

input_data = "https://uny.ac.id"

#Creating an instance of qrcode

qr = qrcode.QRCode(
 version=1,
 box_size=10,
 border=5)
qr.add_data(input_data)

qr.add_data(input_data)

qr.add_data(input_data)

qr.add_data(input_data)

qr.add_data(input_data)

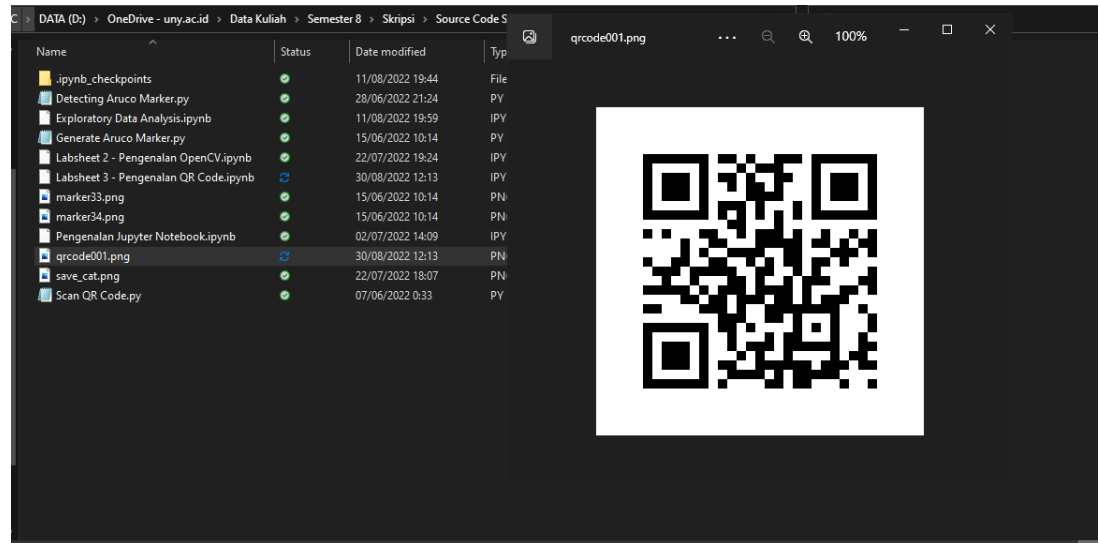
qr.make(fit=True)

img = qr.make_image(fill='black', back_color='white')

img.save('qrcode001.png')

2) Lalu klik **Run**

3) Silahkan cek pada directory jupyter notebook tersebut, seperti berikut



f. Membuat program untuk scan QR Code

- 1) Setelah dibuatnya/generate qrcode maka selanjutnya akan dibuat program untuk melakukan scan qrcode.
- 2) Ketik code berikut ini untuk membuat **function decoder**

```
def decoder(image):
    gray_img = cv2.cvtColor(image,0)
    barcode = decode(gray_img)

    for obj in barcode:
        points = obj.polygon
        (x,y,w,h) = obj.rect
        pts = np.array(points, np.int32)
        pts = pts.reshape((-1, 1, 2))
        cv2.polylines(image, [pts], True, (0, 255, 0), 3)

        barcodeData = obj.data.decode("utf-8")
        barcodeType = obj.type
        string = "Data " + str(barcodeData) + " | Type " + str(barcodeType)

        cv2.putText(frame, string, (x,y), cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,0,0), 2)
        print("Barcode: "+barcodeData + " | Type: "+barcodeType)
```

```
def decoder(image):
    gray_img = cv2.cvtColor(image,0)
    barcode = decode(gray_img)
```

```
for obj in barcode:
    points = obj.polygon
    (x,y,w,h) = obj.rect
    pts = np.array(points, np.int32)
    pts = pts.reshape((-1, 1, 2))
```

```
cv2.polylines(image, [pts], True, (0, 255, 0), 3)
```

```
barcodeData = obj.data.decode("utf-8")
```

```
barcodeType = obj.type
```

```
string = "Data " + str(barcodeData) + " | Type " + str(barcodeType)
```

```
cv2.putText(frame,string,(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,0,0), 2)
```

```
print("Barcode: "+barcodeData + " | Type: "+barcodeType)
```

- 4) Kemudian klik **Run** lalu ketik lagi code berikut pada cell berikutnya

```
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    decoder(frame)
    cv2.imshow('Image', frame)
    code = cv2.waitKey(10)
    if code == ord('q'):
        break
```

```
cap = cv2.VideoCapture(0)
```

```
while True:
```

```
    ret, frame = cap.read()
```

```
    decoder(frame)
```

```
    cv2.imshow('Image', frame)
```

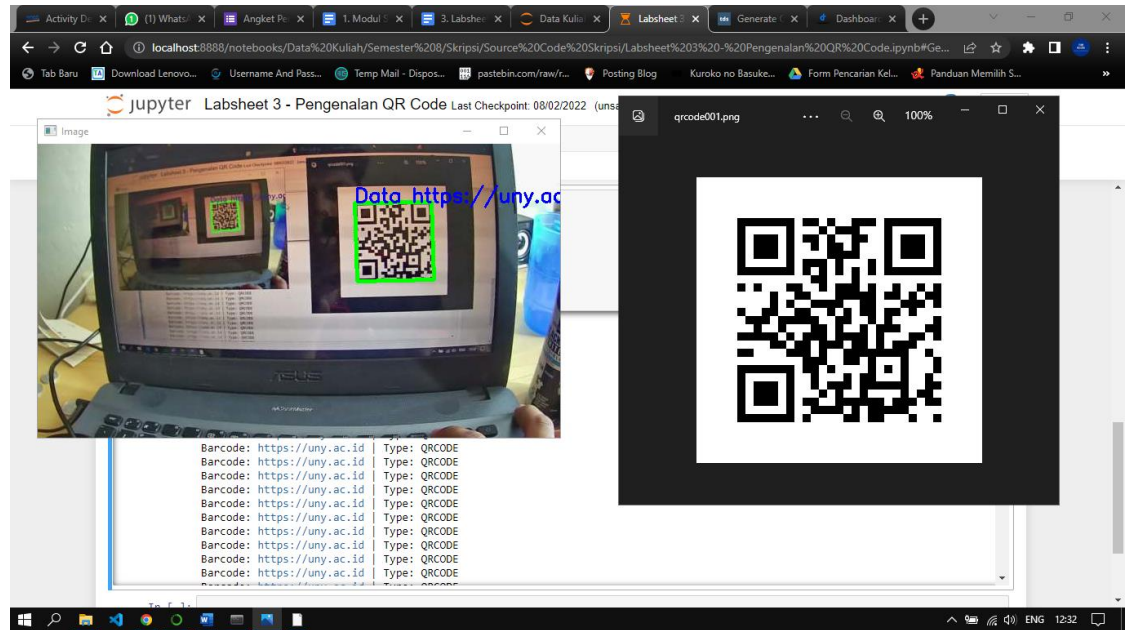
```
    code = cv2.waitKey(10)
```

```
    if code == ord('q'):
```

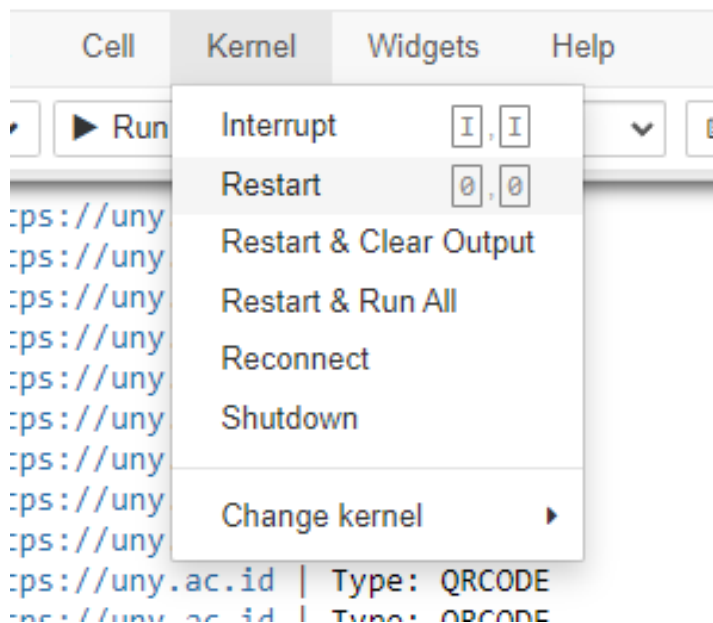
```
        break
```

Note: Apabila memiliki webcam eksternal silahkan ubah **cv2.VideoCapture(0)** menjadi **cv2.VideoCapture(1)**

- 5) Sebelum di **Run** pastikan webcam laptop/pc mu tersedia dan bisa digunakan.
- 6) Berikut akan muncul pop up ketika program dijalankan lalu arahkan untuk menscan qr code



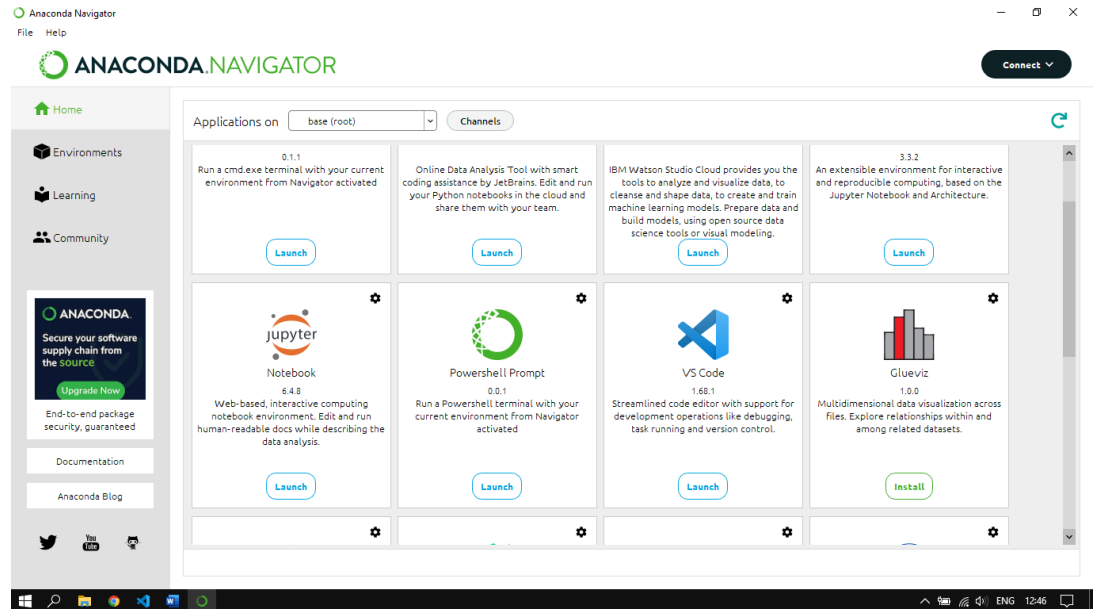
- 7) Terlihat pada output Jupyter Notebook dan tampilan scanner terlihat ekstraksi informasi yang terdapat dalam QR Code.
- 8) Untuk keluar dari program silahkan klik pop up yang muncul lalu klik di keyboard huruf 'q'.
- 9) Apabila terjadi not responding silahkan restart kernel pada Jupyter Notebook seperti berikut.



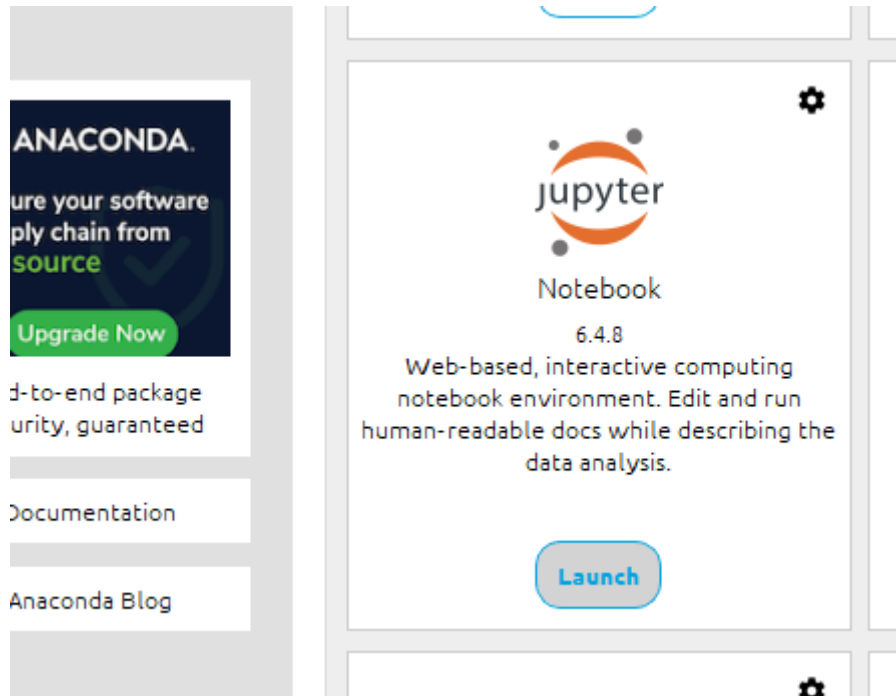
4. Pengenalan Aruco Marker

a. Mengimport library yang dibutuhkan

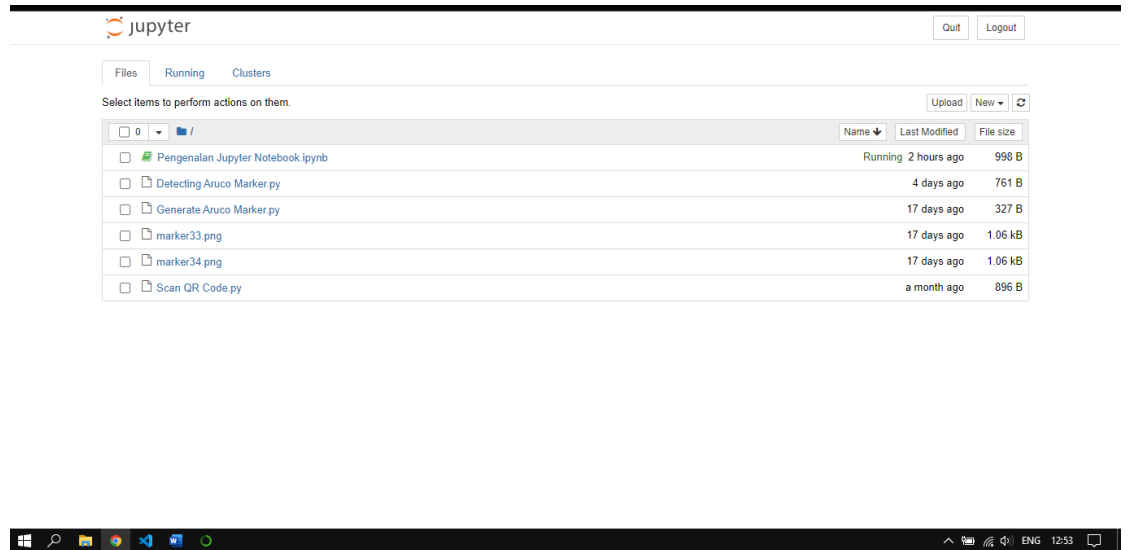
1) Jalankan Anaconda Navigator as Administrator



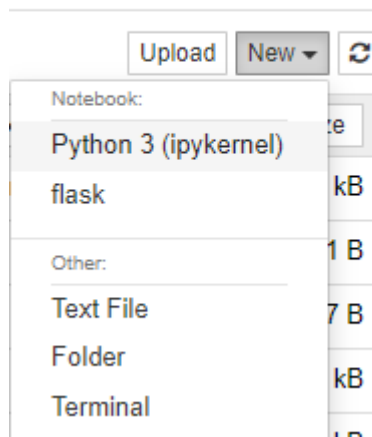
2) Untuk memulai pemrograman Python dengan Jupyter Notebook, silahkan Launch Jupyter Notebook



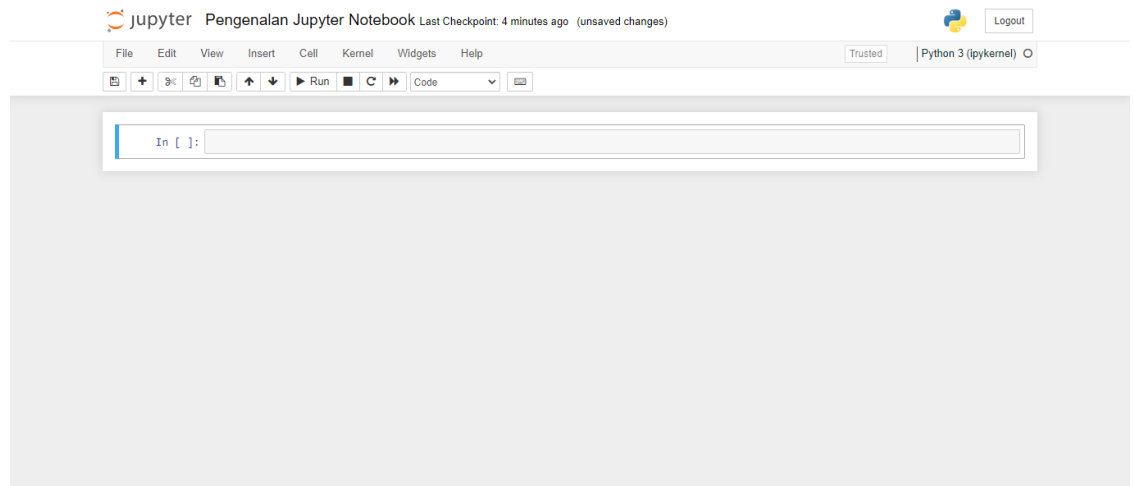
3) Setelah di **Launch** maka akan menuju browser default yang di gunakan pada masing-masing device.



- 4) Untuk memulai menjalankan Script Python, silahkan klik tombol **New** dan arahkan ke Python 3 seperti gambar berikut.



- 5) Lalu jendela baru akan terbuka untuk membuka text editor kernel Jupyter Notebook



6) Kemudian **import library** berikut:

Import Library

```
In [1]: import cv2
import cv2.aruco as aruco
import numpy as np
import os
```

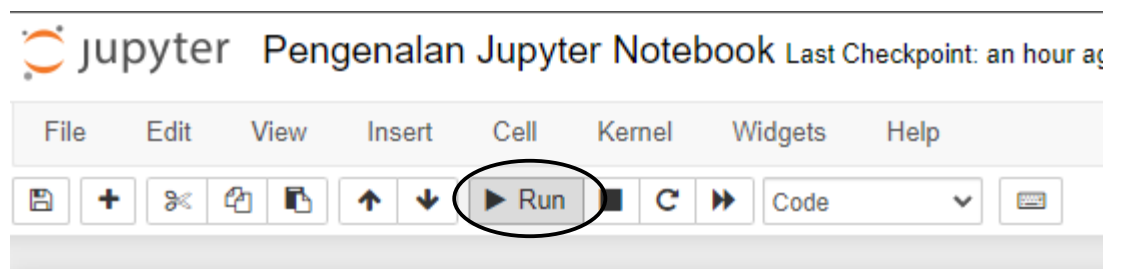
import cv2 as cv

import cv2.aruco as aruco

import numpy as np

import os

7) Klik **Run** pada Jupyter Notebook.



b. Mengenerate Aruco Marker

10) Ketikan kode berikut untuk mengenerate/membuat qrcode dari sebuah link situs.

Generate Aruco Marker

```
In [4]: # Load the predefined dictionary
dictionary = cv.aruco.Dictionary_get(cv.aruco.DICT_6X6_250)

# Generate the marker
markerImage = np.zeros((200, 200), dtype=np.uint8)
markerImage = cv.aruco.drawMarker(dictionary, 33, 200, markerImage, 1);

cv.imwrite("marker33.png", markerImage);
```

Load the predefined dictionary

```
dictionary = cv.aruco.Dictionary_get(cv.aruco.DICT_6X6_250)
```

Generate the marker

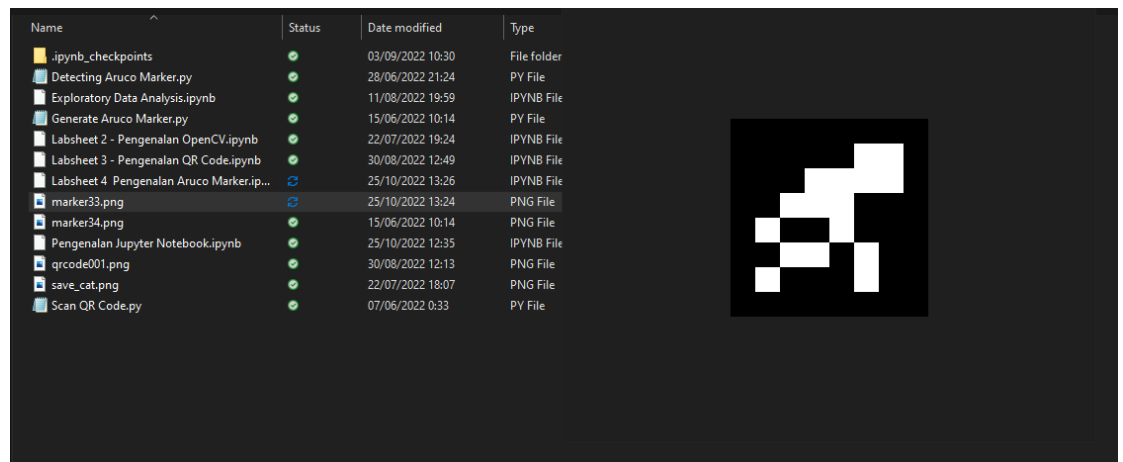
```
markerImage = np.zeros((200, 200), dtype=np.uint8)
```

```
markerImage = cv.aruco.drawMarker(dictionary, 33, 200, markerImage, 1);
```

```
cv.imwrite("marker33.png", markerImage);
```

11) Lalu klik **Run**

12) Silahkan cek pada directory jupyter notebook tersebut maka akan tergenerate Aruco Marker yang dibuat, seperti berikut



c. Membuat program untuk scan Aruco Marker

- 3) Setelah dibuatnya/generate Aruco Tag/Aruco Marker maka selanjutnya akan dibuat program untuk melakukan scan Aruco Tag/Aruco Marker.
- 4) Ketik code berikut ini untuk membuat **function generating Marker**

Detecting Aruco Marker

```
In [2]: def findArucoMarkers(img, markerSize = 4, totalMarkers=250, draw=True):
        gray = cv.cvtColor(img, cv2.COLOR_BGR2GRAY)
        key = getattr(aruco, f'DICT_{markerSize}X{markerSize}_{totalMarkers}')
        arucoDict = aruco.Dictionary_get(key)
        arucoParam = aruco.DetectorParameters_create()
        bboxes, ids, rejected = aruco.detectMarkers(gray, arucoDict, parameters = arucoParam)
        print(ids)
        if draw:
            aruco.drawDetectedMarkers(img, bboxes)

        cap = cv.VideoCapture(1)
        while True:
            success, img = cap.read()
            findArucoMarkers(img)
            cv2.imshow('img',img)
            k = cv2.waitKey(30) & 0xff
            if k == 27:
                break
        cap.release()
        cv2.destroyAllWindows()
```

```
def findArucoMarkers(img, markerSize = 4, totalMarkers=250, draw=True):
```

```
    gray = cv.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
    key = getattr(aruco,
```

```
f'DICT_{markerSize}X{markerSize}_{totalMarkers}')
```

```
    arucoDict = aruco.Dictionary_get(key)
```

```
    arucoParam = aruco.DetectorParameters_create()
```

```
    bboxes, ids, rejected = aruco.detectMarkers(gray, arucoDict, parameters =
arucoParam)
```

```
    print(ids)
```

```
    if draw:
```

```
        aruco.drawDetectedMarkers(img, bboxes)
```

```
cap = cv.VideoCapture(0)
```

```
while True:
```

```
    success, img = cap.read()
```

```
    findArucoMarkers(img)
```

```
    cv2.imshow('img',img)
```

```
    k = cv2.waitKey(30) & 0xff
```

```
    if k == 27:
```

```
        break
```

```
cap.release()
```

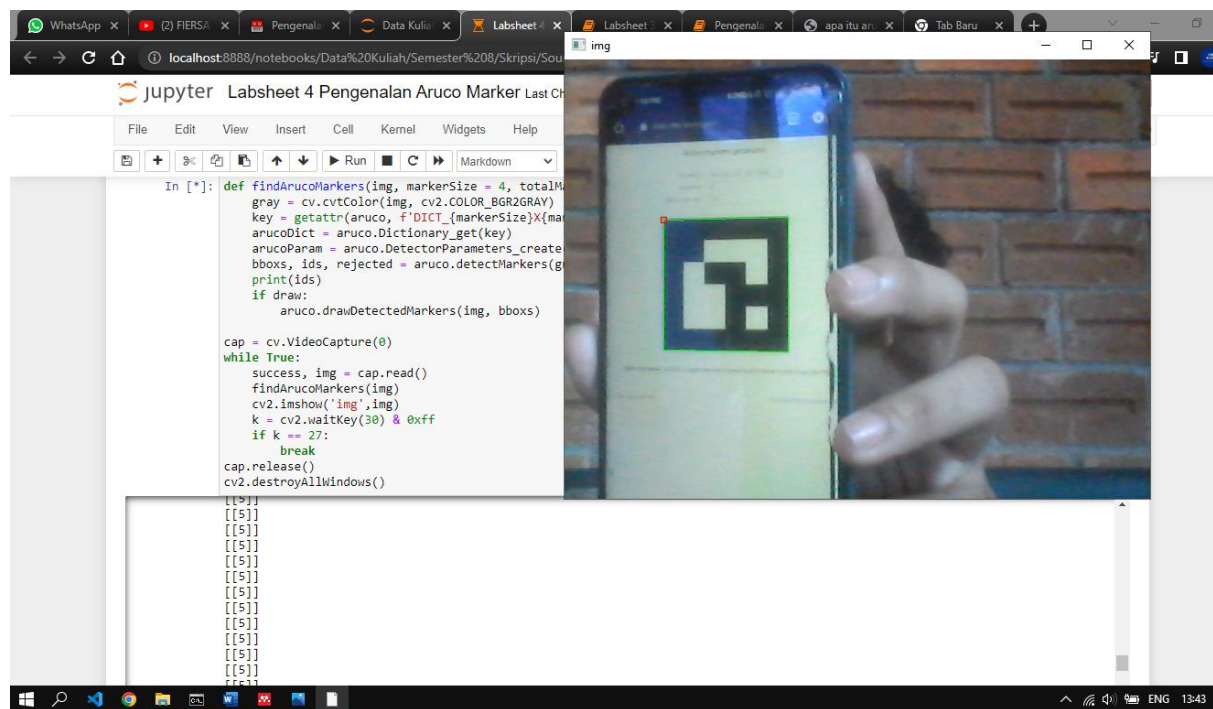
```
cv2.destroyAllWindows()
```

Note: Apabila memiliki webcam eksternal silahkan ubah `cv2.VideoCapture(0)` menjadi `cv2.VideoCapture(1)`

13) Kemudian klik **Run**

14) Sebelum di **Run** pastikan webcam laptop/pc mu tersedia dan bisa digunakan.

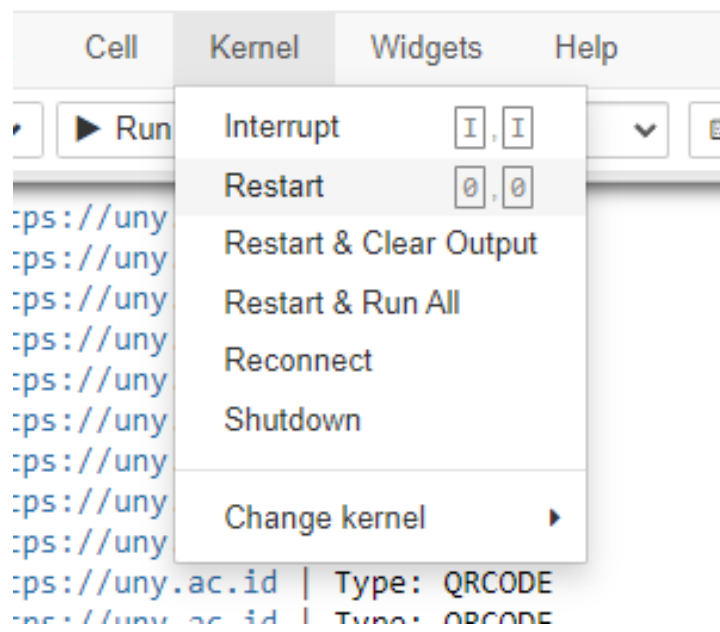
15) Berikut akan muncul pop up ketika program dijalankan lalu arahkan untuk menscan ArTag/Aruco Marker



16) Terlihat pada output Jupyter Notebook dan tampilan scanner terlihat ekstraksi informasi yang terdapat dalam Aruco Marker yaitu id .

17) Untuk keluar dari program silahkan klik pop up yang muncul lalu klik di keyboard `'esc'`.

18) Apabila terjadi not responding silahkan restart kernel pada Jupyter Notebook seperti berikut.



Daftar Pustaka

- Kadir, A. (2019). Langkah Mudah Pemrograman OpenCV & Python. In *Langkah Mudah Pemrograman OPenCV & Python*. PT Elex Media Komputindo.
- Nayak, S. (2020). *Augmented Reality using ArUco Markers in OpenCV (C++ / Python)*. <https://learnopencv.com/augmented-reality-using-aruco-markers-in-opencv-c-python/>
- Tiwari, S. (2017). An introduction to qr code technology. *Proceedings - 2016 15th International Conference on Information Technology, ICIT 2016, December 2016*, 39–44. <https://doi.org/10.1109/ICIT.2016.38>
- Wijaya, A., & Gunawan, A. (2016). *Penggunaan qr code sarana penyampaian promosi dan informasi kebun binatang berbasis android*. 4(1), 16–21.