

## Modul 1 – Pengantar Citra Digital: Representasi, Karakteristik, dan Aplikasi dalam Visi Komputer

### A. TUJUAN

1. Mampu membuat *Repository* di *Github*
2. Mampu membuat *Python Notebook* di *Google Colaborator*
3. Mampu menyimpan dan mengakses *Python Notebook* dari *Google Colaborator*
4. Mampu membuka *File* Citra dari *Code Python*
5. Mampu mengakses pixel pada citra
6. Mampu membuka file image dari Google Drive pribadi
7. Mampu memahami dasar-dasar library OpenCV di Python
8. Mampu memahami channel warna pada OpenCV dan konversinya

### B. ALAT DAN BAHAN

1. PC/LAPTOP
2. Github
3. *Google Colaborator*

### C. ULASAN TEORI

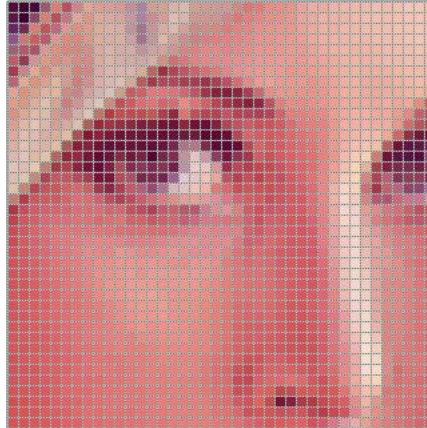
#### C1. Pengantar Citra Digital dan Visi Komputer

##### a) Konsep dasar citra digital dan visi computer

Citra digital adalah representasi visual dari objek dunia nyata dalam bentuk numerik, tersusun atas unit terkecil bernama **piksel**. Setiap piksel memiliki nilai intensitas (pada citra grayscale) atau kombinasi warna (pada citra RGB, RGBA, HSV, dll.). Karakteristik citra digital ditentukan oleh **resolusi** (jumlah piksel horizontal × vertikal), **bit depth** (kemampuan merepresentasikan jumlah warna), serta **format penyimpanan**. Aspek-aspek ini berpengaruh terhadap kualitas dan ukuran citra, yang menentukan kecocokan penggunaannya pada aplikasi tertentu, misalnya citra medis, penginderaan jauh, hingga aplikasi media sosial [Gonzalez & Woods, 2018] .

Citra Digital (Citra Raster) adalah representasi numerik dari citra dua dimensi. Nilai numerik yang direpresentasikan umumnya adalah nilai biner 8 bit. Nilai biner ini

disimpan pada elemen citra yang sering disebut sebagai pixel. Citra digital berisi pixel yang jumlah baris dan kolomnya tetap. Pixel adalah elemen gambar terkecil dari citra digital. Pixel disimpan pada memory computer sebagai map raster, yaitu array dua dimensi bertipe integer.



Citra Raster diakuisisi menggunakan berbagai macam perangkat input dan teknik, seperti digital camera, scanner, radar, camera infra merah, dan lain sebagainya. Pengolahan Citra Digital adalah ilmu yang mempelajari algoritma transformasi citra.

**Visi komputer (Computer Vision)** adalah cabang AI yang memungkinkan mesin meniru kemampuan manusia dalam melihat dan memahami citra. Perkembangannya dipacu oleh **GPU berdaya tinggi**, **dataset skala besar**, dan **algoritma deep learning**. Pada 2025, visi komputer telah diaplikasikan luas, antara lain:

- Kendaraan otonom (Tesla, Waymo, Hyundai Ioniq 2025).
- Analisis citra medis (AI multimodal seperti **Med-PaLM 2**).
- AgriTech (deteksi penyakit tanaman dan ikan berbasis YOLOv9).
- AR/VR melalui perangkat seperti **Apple Vision Pro 2 (2025)**.

Perkembangan mutakhir menekankan **model berbasis transformer** (Vision Transformer, SAM 2) dan **YOLOv9** yang efisien untuk deteksi real-time. Paradigma pun bergeser dari *handcrafted features* (HOG, SIFT) menuju pendekatan **end-to-end deep learning** yang adaptif dan presisi tinggi 【Dosovitskiy et al., 2024; Wang et al., 2024】.

### b) Pengenalan tools pendukung

#### 1. Python

Python adalah bahasa pemrograman yang saat ini menjadi standar de facto dalam bidang data science, machine learning, deep learning, serta pengolahan citra digital. Python populer karena memiliki sintaks yang sederhana, mendukung

pemrograman berorientasi objek, dan dilengkapi dengan ribuan pustaka open-source yang memudahkan pengembangan aplikasi. Dibandingkan dengan bahasa lain seperti C++ atau Java, Python memiliki kurva belajar yang lebih ramah bagi mahasiswa, tetapi tetap powerful untuk riset maupun implementasi industri.

Bahasa Python termasuk bahasa pemrograman trend di beberapa tahun terakhir ini. Pada awalnya Python merupakan kelanjutan bahasa pemrograman ABC yang dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam. Python memiliki ciri khusus jika dibandingkan dengan bahasa lain dalam hal penulisan kode program yang memiliki aturan mengenai indentasi, tipe data, tuple, dan dictionary. Kelebihan Python dibandingkan dengan bahasa pemrograman lain terutama terlihat dalam hal penanganan modul, serta keunggulan Python yang merupakan produk yang opensource, free, dan multiplatform. Python memiliki beberapa keunggulan antara lain : Terdapat modul-modul yang telah disediakan oleh Python; tata bahasa lebih mudah dipahami; layout yang dimiliki lebih mudah untuk ditinjau ulang dan dikembangkan; berorientasi obyek; pengolahan memori dilakukan secara modular; dapat dibangun dengan bahasa lain Python sendiri ataupun C/C++.modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++; serta kelebihan lain.

Dalam konteks pengolahan citra dan visi komputer, Python menyediakan berbagai pustaka populer seperti:

- **NumPy** untuk manipulasi array numerik.
- **OpenCV** untuk pengolahan citra, deteksi objek, dan tracking.
- **Matplotlib dan Seaborn** untuk visualisasi data dan citra.
- **scikit-image** untuk pemrosesan citra tingkat lanjut (filtering, edge detection, transformasi morfologi).
- **TensorFlow dan PyTorch** untuk deep learning yang digunakan dalam pengembangan model klasifikasi dan deteksi citra.

**NumPy (Numerical Python)** merupakan library Python untuk data scientist dengan kemampuan untuk membentuk objek N-dimensional array. NumPy merupakan bagian ekosistem Python dari tool open source SciPy. Kemampuan yang dimiliki ini memiliki kemiripan dengan pada Python. Akan tetapi konsumsi memory dan NumPy array lebih kecil dan runtime yang lebih cepat dibandingkan list pada Python. Oleh karena itu NumPy juga memudahkan dalam mengeksekusi operasi terkait Vector (1-d array) dan Matrix (2-d array).



Sebelum menggunakan NumPy terlebih dahulu harus melakukan import package Numpy menggunakan pip install numpy. Akan tetapi jika Numpy sudah terinstall atau kita sudah menggunakan fasilitas google Colab maka NumPy bisa langsung di import menggunakan `import numpy as np`.

Pembuatan array dengan NumPy diterapkan dengan penambahan tipe data dengan parameter dtype. Berikut ini adalah beberapa tipe yang terdapat pada NumPy :

|            |  |
|------------|--|
| bool_      | Boolean (True or False) stored as a byte   |
| int_       | Default integer type   |
| intc       | Identical to C int e.g int32 in64  |
| intp       | Integer used for indexing  |
| int8       | Byte (-128 to 127)   |
| int16      | Integer (-32768 to 32767)  |
| int32      | Integer (-2147483648 to 2147483647)  |
| int64      | Integer (-9223372036854775808 to 9223372036854775807)                            |
| uint8      | Unsigned integer (0 to 255)  |
| uint16     | Unsigned integer (0 to 65535)  |
| uint32     | Unsigned integer (0 to 4294967295)   |
| uint64     | Unsigned integer (0 to 18446744073709551615)                                     |
| float16    | Half precision float: sign bit, 5 bits exponent, 10 bits mantissa                |
| float32    | Single precision float: sign bit, 8 bits exponent, 23 bits mantissa              |
| float64    | Double precision float: sign bit, 11 bits exponent, 52 bits mantissa             |
| complex64  | Complex number, represented by two 32-bit floats (real and imaginary components) |
| complex128 | Complex number, represented by two 64-bit floats (real and imaginary components) |

(Sumber : <https://vsvaibhav2016.medium.com/basics-of-numpy-python-for-data-analysis-45b0c43f591b>)

Array NumPy disebut sebagai ndarray, atau dikenal juga dengan sebutan array. Untuk mengatasi permasalahan dengan array khususnya array multidimensi dengan NumPy, dimensi NumPy dimensi disebut sebagai axis, dan banyaknya axis disebut

rank. Untuk ruang 3D, koordinat titik diruang 3D 1 axis dengan panjang 3 dapat dinyatakan seperti [2,1,1]. Sedangkan panjang axis tersebut adalah 3. Selain contoh tersebut array seperti ini [[ 1. , 1. , 1.], [ 2. , 2. , 2.]] adalah contoh array dengan rank 2 yang memiliki 2 axis dimana axis pertama panjangnya 2, dan axis kedua memiliki panjang 3. Berikut ini adalah beberapa atribut penting dari ndarray :

| Atribut          | Kegunaan  |
|------------------|---|
| ndarray.ndim     | Jumlah dimensi array. Pada bahasa Python biasa disebut rank.  |
| ndarray.shape    | Ukuran array di setiap dimensi. Matriks dengan n baris dan m kolom atau matriks (n,m).  |
| ndarray.size     | Jumlah elemen suatu array atau merupakan perkalian elemen dari shape  |
| ndarray.dtype    | Merupakan tipe data Python standar suatau elemen dalam array  |
| ndarray.itemsize | Ukuran dalam byte dari setiap elemen array. Contoh : Sebuah array yang memiliki elemen dengan tipe float64 memiliki itemsize 8 (= 64/8) |
| ndarray.data     | Buffer berisikan elemen array   |

Terdapat beberapa metode utama NumPy array yang digunakan untuk memanipulasi data sebagai berikut :

### 1. Indexing

Metode untuk mengakses suatu elemen array. Contohnya : `a[5]`. A adalah suatu array NumPy, sedangkan 5 artinya elemen ke-5 pada array tersebut.

### 2. Slicing

Metode untuk mengakses array khususnya subarray (beberapa elemen sekaligus). Akses subarray dilakukan dengan karakter khusus (:). Contohnya : `a[2:6]`, artinya akses elemen array dilakukan mulai dari index 2 sampai dengan 5(6-1). `a[:5]`, artinya akses 5 elemen pertama dari suatu array. `a[::2]`, artinya akses array dilakukan mulai index 0, sampai dengan index akhir, dengan jarak 2 elemen setiap aksesnya.

### 3. Iterating

Metode akses array secara iteratif. Contohnya :

```
for row in b:
    print (row )
```

Artinya setiap array diakses dan ditampilkan per baris

```
for element in b.flat :
    print ( element )
```

Artinya setiap array diakses dan ditampilkan per elemen di setiap baris

## 4. Reshaping

Metode untuk mengubah bentuk array. Contohnya :

```
a=np.array ([ 3. , 6. , 1. , 5. , 3. , 2. , 0. , 0. , 2. , 5. , 1. , 4.])
```

```
a. reshape (6 ,2)
```

Artinya array a akan diubah bentuk menjadi 6 baris 2 kolom menjadi

```
array([[3., 6.],  
       [1., 5.],  
       [3., 2.],  
       [0., 0.],  
       [2., 5.],  
       [1., 4.]])
```

**OpenCV (Open Source Computer Vision Library)** dikembangkan oleh Intel Corporation dan merupakan library open source yang disediakan untuk programming terkait citra digital dan banyak metode AI (Artificial Intellegence). OpenCV memiliki banyak fitur terkatit citra digital seperti pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, algoritma sederhana pada Computer Vision untuk low level API, dll. OpenCV awalnya merupakan library untuk bahasa pemrograman C/C++ dan kini dikembangkan ke bahasa lain juga seperti Phytion, java, dan juga matlab.



| Versi      | Tahun Rilis | Fitur Utama  | Fokus Utama                      |
|------------|-------------|--|----------------------------------|
| OpenCV 1.x | 2000 – 2006 | - Fungsi dasar citra (filter, transformasi, edge detection)- Deteksi wajah dengan Haar Cascade | Pengolahan citra dasar (C / C++) |

| Versi             | Tahun Rilis | Fitur Utama  | Fokus Utama                         |
|-------------------|-------------|--|-------------------------------------|
| <b>OpenCV 2.x</b> | 2009 – 2015 | - API modern berbasis C++- Modul machine learning (cv::ml)- Dukungan CUDA (GPU) awal   | Peralihan ke C++ dan akselerasi GPU |
| <b>OpenCV 3.x</b> | 2015 – 2018 | - Modularisasi (core, imgproc, video, ml, dll.)- Contrib modules (SIFT, SURF, Face Recognition)- Video analysis (optical flow, tracking KCF/MIL/TLD)- Dukungan Python lebih baik | Computer vision klasik + tracking   |
| <b>OpenCV 4.x</b> | 2018 – 2023 | - Modul DNN (Deep Neural Networks)- Dukungan TensorFlow, PyTorch, Caffe, ONNX- Optimisasi hardware (Intel OpenVINO, NVIDIA TensorRT)- Full support Python 3                      | Integrasi deep learning             |
| <b>OpenCV 5.x</b> | 2024 – 2025 | - Optimisasi penuh untuk GPU NVIDIA & Intel Arc- Dukungan Vision Transformer (ViT), YOLOv9, SAM 2- Dukungan WebAssembly & JavaScript- Integrasi visualisasi (Plotly, Streamlit)  | Jembatan CV tradisional ↔ AI modern |

Versi pertama OpenCV dikenalkan pada tahun 1999 oleh Gary Bradsky dan mulai dan mulai dirilis pada tahun 2000. OpenCV versi pertama ini memerlukan library dari Intel Image Processing dalam penggunaannya. Akan tetapi ketergantungan akan library tersebut sudah dihilangkan dan sekarang berdiri sebagai library standalone dan multiplatform sehingga dapat digunakan pada berbagai operating system baik window, linux, macOSX maupun android. OpenCV memiliki 5 (lima) library dasar sebagai berikut :

1. CV  
Mendukung algoritma image processing dan juga computer vision
2. ML  
Mendukung algoritma machine learning
3. Highgui  
Mendukung GUI, image, video I/O
4. CXCORE  
Mendukung struktur data, XML,
5. CvAux

## Khusus modul computer vision

Berdasarkan library utama yang dimiliki OpenCV terdapat beberapa fitur utama sebagai berikut :

1. Image and video I/O
2. Computer Vision secara umum dan pengolahan citra digital ( untuk low dan mid level API)
3. Modul computer vision high level
4. Metode untuk AI dan machine learning
5. Sampling gambar dan transformasi
6. Metode untuk menciptakan dan menganalisa gambar biner
7. Metode untuk memperhitungkan pemodelan 3D
8. Pendeteksi gerak
9. Analisis Struktural
10. Kalibrasi Kamera
11. Image Labelling
12. Dll

Tren terbaru 2025 menunjukkan bahwa Python semakin terintegrasi dengan teknologi multimodal AI (gambar, teks, video, suara) dan komputasi GPU/TPU. Versi terbaru Python 3.12 dan 3.13 juga meningkatkan performa eksekusi hingga 20–30% dibanding versi sebelumnya, menjadikannya semakin efisien untuk eksperimen berbasis big data. Dengan fleksibilitas tinggi dan dukungan komunitas global, Python adalah fondasi utama yang wajib dikuasai mahasiswa dalam bidang ini 【Gonzalez & Woods, 2018; Szeliski, 2022】 .

## 2. GitHub

GitHub adalah platform berbasis cloud untuk version control dan kolaborasi kode menggunakan sistem Git. Dalam pengembangan aplikasi AI dan visi komputer, GitHub sangat penting karena memungkinkan tim untuk bekerja secara paralel, melacak perubahan kode, dan menggabungkan kontribusi dari berbagai peneliti atau pengembang. GitHub juga berfungsi sebagai repositori publik terbesar di dunia, sehingga mahasiswa dapat mengakses jutaan proyek open-source sebagai referensi atau dasar eksperimen.





Perkembangan terbaru pada tahun 2025 adalah hadirnya GitHub Copilot X, sebuah asisten pemrograman berbasis AI (didukung oleh model GPT-4/5 multimodal) yang dapat membantu menulis kode, memberikan rekomendasi perbaikan, hingga menghasilkan dokumentasi otomatis. Selain itu, fitur GitHub Codespaces memudahkan pengembangan aplikasi langsung di cloud tanpa harus menginstal IDE atau pustaka di komputer lokal. Hal ini sangat bermanfaat bagi mahasiswa yang memiliki keterbatasan perangkat keras.

GitHub juga sering digunakan untuk berbagi dataset citra dan model terlatih. Banyak riset mutakhir dalam pengolahan citra, misalnya proyek YOLOv9, SAM (Segment Anything Model), atau CLIP, semuanya dirilis pertama kali melalui GitHub agar dapat digunakan dan dikembangkan lebih lanjut oleh komunitas global [GitHub, 2025] .

### **3. Google Colaboratory (Colab)**

Google Colaboratory atau lebih dikenal dengan Google Colab adalah platform cloud berbasis Jupyter Notebook yang memungkinkan eksekusi kode Python secara interaktif tanpa perlu instalasi lokal. Colab menjadi favorit mahasiswa dan peneliti karena menyediakan akses GPU (Graphics Processing Unit) dan TPU (Tensor Processing Unit) secara gratis, sehingga eksperimen machine learning dan visi komputer dapat dilakukan tanpa perlu perangkat keras mahal.



Google Collaboratory atau Google Colab merupakan tools yang berbasis cloud dan bersifat free. Google Colab dibuat dengan environment jupyter dan mendukung banyak pustaka (library) yang dibutuhkan dalam lingkungan pengembangan Artificial Intelligence (AI). Google Colab memungkinkan penggunaanya untuk menulis dan mengeksekusi Python di browser (Chrome, Firefox dan Safari) tanpa memerlukan konfigurasi, dapat mengakses GPU secara gratis, serta dapat berbagi kode program (kolaborasi tim) dengan mudah. Google Colab dapat mengeksekusi, menulis, menyimpan bahkan membagikan kode program yang telah dibuat melalui google drive. Penulisan kode program Python serta eksekusi kode program tersebut tidak memerlukan proses instalasi dengan mendownload installer pada komputer. Instalasi tidak perlu dilakukan karena semua proses dilakukan di cloud. Selain itu fungsionalitas tambahan Python dapat juga memanfaatkan built-in library yang ada pada Google Colab.

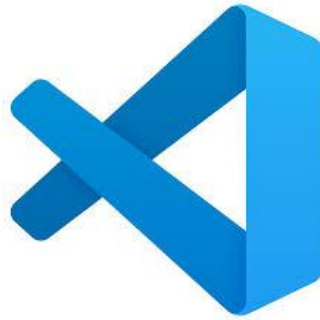
Pada tahun 2025, Google telah merilis Colab Pro++, yang memberikan akses ke GPU generasi terbaru seperti NVIDIA A100 dan H100, serta integrasi langsung dengan Google Cloud Vertex AI. Hal ini memungkinkan mahasiswa tidak hanya menjalankan eksperimen sederhana, tetapi juga melatih model deep learning skala besar untuk aplikasi visi komputer yang kompleks.

Kelebihan utama Colab adalah integrasinya dengan Google Drive, yang memudahkan penyimpanan dataset dan hasil eksperimen. Colab juga mendukung integrasi dengan pustaka mutakhir seperti TensorFlow, PyTorch, dan OpenCV tanpa perlu konfigurasi manual. Dengan antarmuka yang sederhana dan aksesibilitas tinggi, Colab menjadi pintu masuk utama mahasiswa untuk mempraktikkan teori pengolahan citra digital dalam bentuk kode nyata [Bisong, 2024] .

#### 4. Software Desktop untuk Python

Selain menggunakan platform cloud seperti Google Colab, mahasiswa juga dapat mengembangkan program Python menggunakan berbagai software desktop (IDE – Integrated Development Environment). Beberapa pilihan populer:

- Visual Studio Code (VS Code) → ringan, fleksibel, mendukung banyak ekstensi untuk Python, GitHub, dan Jupyter Notebook.



- PyCharm → IDE khusus Python dengan fitur debugging dan manajemen proyek yang lengkap. Cocok untuk proyek besar.



- Jupyter Notebook (Anaconda) → berbasis notebook interaktif, ideal untuk eksplorasi data dan pembelajaran interaktif.



- Spyder → IDE berbasis Anaconda dengan tampilan mirip MATLAB, populer di kalangan akademisi.



Pemilihan software tergantung kebutuhan. Misalnya, untuk riset akademis, Jupyter Notebook dan VS Code sangat cocok karena mendukung dokumentasi interaktif dan visualisasi data. Untuk proyek industri atau penelitian yang kompleks, PyCharm memberikan fitur manajemen proyek yang lebih terstruktur.

Pada 2025, perkembangan terbaru menunjukkan integrasi IDE dengan AI assistant coding (misalnya Copilot di VS Code atau ChatGPT Plugin di PyCharm), yang semakin mempermudah mahasiswa dalam menulis kode dan memahami pustaka baru. Dengan adanya dukungan lintas platform (Windows, Linux, MacOS), mahasiswa dapat memilih software yang sesuai dengan perangkat masing-masing.

## C2. Representasi dan Karakteristik Citra Digital

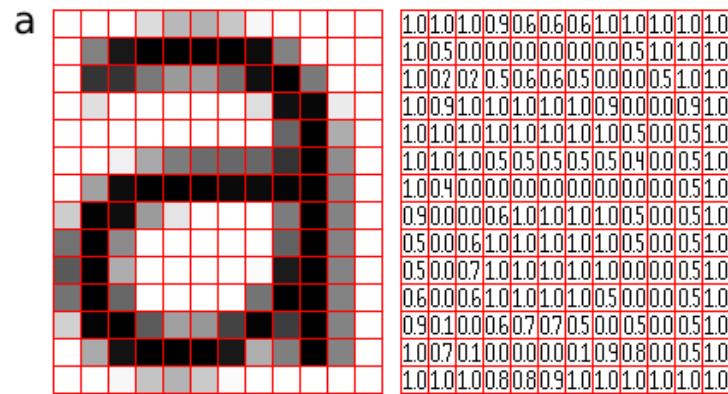
### a) Format citra dan model warna

Citra digital yang umum digunakan pada pengolahan citra adalah Citra biner, Citra Keabuan, dan Citra Berwarna RGB (Red, Green, dan Blue). Citra digital dapat disimpan dalam berbagai format file, yang masing-masing memiliki keunggulan dan keterbatasan. Format populer antara lain:

- **JPEG/JPG (Joint Photographic Experts Group):** menggunakan kompresi lossy, ukuran file lebih kecil, cocok untuk citra fotografi.
- **PNG (Portable Network Graphics):** menggunakan kompresi lossless, mendukung transparansi (RGBA), ideal untuk grafis dan web.
- **TIFF (Tagged Image File Format):** mendukung kualitas tinggi dengan kompresi lossless, sering digunakan dalam pencitraan medis dan penelitian ilmiah.
- **BMP (Bitmap Image File):** tanpa kompresi, ukuran besar, digunakan untuk keperluan grafis sederhana.
- **HEIF/HEIC (High Efficiency Image Format):** format modern berbasis HEVC, ukuran kecil dengan kualitas tinggi, kini digunakan pada perangkat mobile generasi baru (iOS/Android 2024–2025).

Citra biner adalah citra yang memiliki 2 warna saja, yaitu hitam dan putih. Jika direpresentasikan dengan nilai biner 8bit adalah warna hitam bernilai 0000 0000, dan putih bernilai 1111 1111. biasa ditampilkan dengan nilai normalisasi 0 dan 1, atau decimal 0 dan 255, atau heksadesimal 00x dan FFx.

Citra keabuan adalah citra yang memiliki derajat keabuan sebanyak 256 warna. Dimulai dengan warna terkecilnya yaitu hitam, dan warna terbesarnya adalah putih. Pada gambar berikut ditunjukkan representasi nilai normalisasi citra keabuan dimana 1.0 menyatakan warna putih, 0.0 menyatakan warna hitam, dan nilai antara 0.0 – 1.0 menyatakan warna derajat keabuannya.

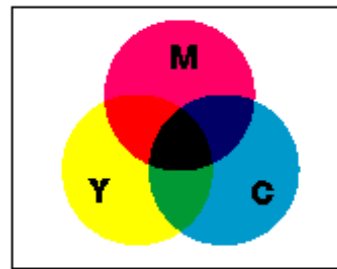


Gambar 1 Konsep Pixel dan Representasi nilai normalisasi citra keabuan

Selain format file, representasi citra juga sangat bergantung pada model warna. Model warna adalah cara matematis untuk merepresentasikan warna dalam ruang multidimensi. Model yang sering digunakan dalam visi komputer:

- **RGB (Red, Green, Blue):** standar tampilan monitor, setiap warna direpresentasikan sebagai kombinasi tiga kanal.
- **HSV (Hue, Saturation, Value):** lebih dekat dengan persepsi manusia, memudahkan segmentasi warna.
- **CMYK (Cyan, Magenta, Yellow, Black):** digunakan pada percetakan.
- **YCbCr:** digunakan dalam kompresi video (MPEG, JPEG).
- **Lab Color Space (CIELAB):** model perseptual yang berusaha meniru cara mata manusia melihat warna.

Citra berwarna RGB adalah citra yang memiliki 3 level / channel warna direpresentasikan dengan resolusi citra 3 dimensi. Pada citra digital, level pertama digunakan untuk menyimpan warna R (Red / merah), Level kedua digunakan untuk menyimpan warna G (Green / hijau), dan level ketiga digunakan untuk menyimpan warna B (Blue / biru). Pada perangkat keluaran seperti LCD Monitor, RGB disusun secara array dan berukuran sangat kecil. Representasi nilainya disimpan dalam nilai biner. Warna Hitam direpresentasikan dengan R = 0000 0000, G = 0000 0000, dan B = 0000 0000. Warna Merah direpresentasikan dengan R = 1111 1111, G = 0000 0000, dan B = 0000 0000. Karena memiliki 3 level, maka ukuran file citra RGB dibandingkan dengan citra keabuan adalah 3:1 dengan jumlah pixel (resolusi citra) yang sama. Warna lain selain Merah, hijau, dan biru adalah campuran dari ketiga warna tersebut. Perhatikan gambar berikut, warna kuning adalah campuran dari warna merah dan hijau. Kuning direpresentasikan dengan nilai R = 1111 1111, G = 1111 1111, dan B = 0000 0000.



CMY: Color Printing Press  
Use Subtractive Color



Pemilihan format citra dan model warna sangat penting karena memengaruhi efisiensi penyimpanan, kualitas visual, serta keakuratan dalam pengolahan citra [Gonzalez & Woods, 2018] .

### **b) Resolusi citra dan video**

Resolusi citra didefinisikan sebagai jumlah piksel dalam dimensi horizontal × vertikal. Semakin tinggi resolusi, semakin detail informasi yang ditampilkan. Contoh standar resolusi gambar:

- HD (1280×720)
- Full HD (1920×1080)
- 4K UHD (3840×2160)
- 8K UHD (7680×4320)

Dalam video digital, resolusi menjadi lebih kompleks karena selain dimensi piksel, juga terdapat frame rate (fps – frames per second). Misalnya:

- Video 1080p @ 30fps (standar media sosial).
- Video 4K @ 60fps (gaming & sinematografi).
- Video 8K @ 120fps (aplikasi VR/AR dan riset medis).

Perkembangan teknologi 2025 menunjukkan meningkatnya penggunaan video resolusi ultra-tinggi (8K–16K) pada riset medis dan kendaraan otonom, serta frame rate tinggi (240fps) pada aplikasi slow motion dan analisis ilmiah. Selain itu, konsep super-resolution berbasis AI juga berkembang pesat, memungkinkan peningkatan resolusi citra/video secara otomatis dengan model deep learning (misalnya ESRGAN dan Real-ESRGAN 2024) [Wang et al., 2024] .

### **c) Aplikasi Pengolahan Citra Digital**

Pengolahan citra digital memiliki spektrum aplikasi yang luas di berbagai bidang, antara lain:

- Kesehatan: deteksi kanker melalui MRI/CT, analisis X-ray paru (COVID-19, pneumonia), rekonstruksi 3D organ.
- Industri & Otomasi: sistem inspeksi kualitas produk berbasis vision, deteksi cacat produksi.
- Pertanian & Perikanan: klasifikasi tanaman, deteksi penyakit, sistem penghitung benih ikan lele berbasis YOLO + MobileNetV2 (riset 2023–2025).
- Transportasi: kendaraan otonom (deteksi objek jalan, pejalan kaki).

- Keamanan: pengenalan wajah (face recognition), pengawasan berbasis video.
- Media & Hiburan: augmented reality (AR), virtual reality (VR), dan efek visual.

Tren 2025 menunjukkan integrasi pengolahan citra dengan AI multimodal, di mana gambar, teks, suara, dan data sensor digabungkan untuk menghasilkan pemahaman yang lebih komprehensif. Contohnya adalah Segment Anything Model 2 (Meta, 2025) yang mampu melakukan segmentasi objek universal pada berbagai domain, serta penggunaan foundation models seperti CLIP dan DINOv2 yang dapat menghubungkan citra dengan bahasa alami [Khan et al., 2024] .

### D. PRAKTIKUM

#### D1. Pengenalan Github dan Google Colab

1. Bagi yang belum mempunyai akun Github, bisa membuat akun baru di Github ([https://github.com/join?ref\\_cta=Sign+up&ref\\_loc=header+logged+out&ref\\_page=%2F&source=header-home](https://github.com/join?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home))

Join GitHub

## Create your account

Username \*

Email address \*

Password \*

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.  
[Learn more.](#)

Email preferences

☒ Send me occasional product updates, announcements, and offers.

Verify your account



School work and student projects

Use the GitHub API

Other

I am interested in:

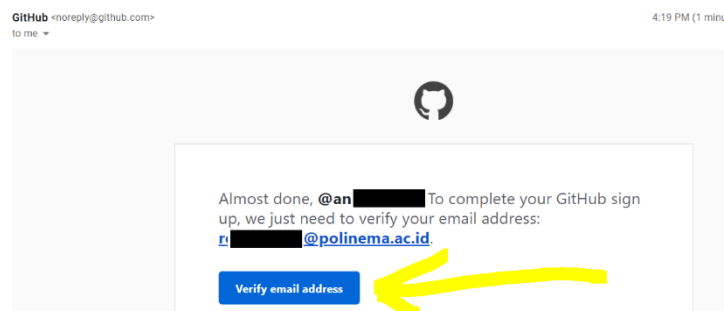
languages, frameworks, industries

We'll connect you with communities and projects that fit your interests.

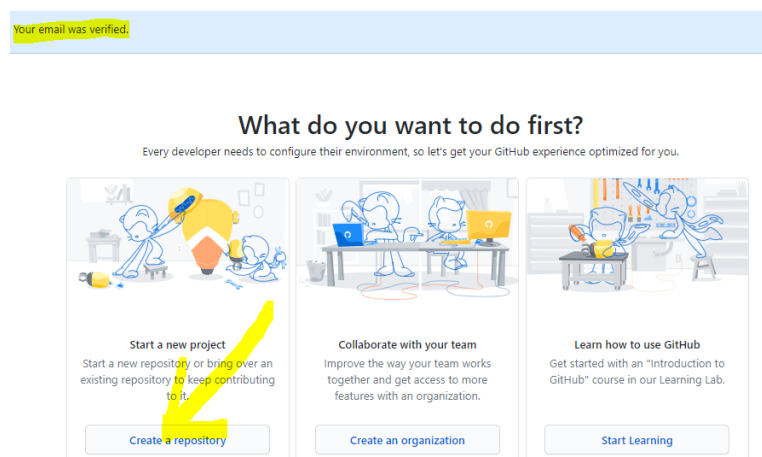
For example: lineageos actions producthunt

Complete setup

Setelah melakukan pengisian semua *mandatory field*, klik “complete setup”. Langkah selanjutnya adalah verifikasi alamat email yang digunakan untuk membuat akun baru.



2. Masuk ke dalam akun github yang telah dibuat, dan buatlah repositori baru, dengan memilih “Create a Repository”.





Isikan nama repositori, deskripsi (opsional), dan pilih apakah repositori yang akan dibuat bersifat *public* atau *private*. Pilih *Private* jika project tidak ingin

diakses oleh publik. Setelah itu klik “*Create repository*” untuk membuat repositori baru.

### Create a new repository

Owner <sup>\*</sup> Repository name <sup>\*</sup>

 andrie2000 / PCVK\_Genap\_2021 

Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-disco](#)?

Description (optional)

Berisi Project-project untuk praktikum mata kuliah PCVK Genap 20/21 JTI Polinema

☐ Public  
Anyone on the internet can see this repository. You choose who can commit.

☒ Private  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

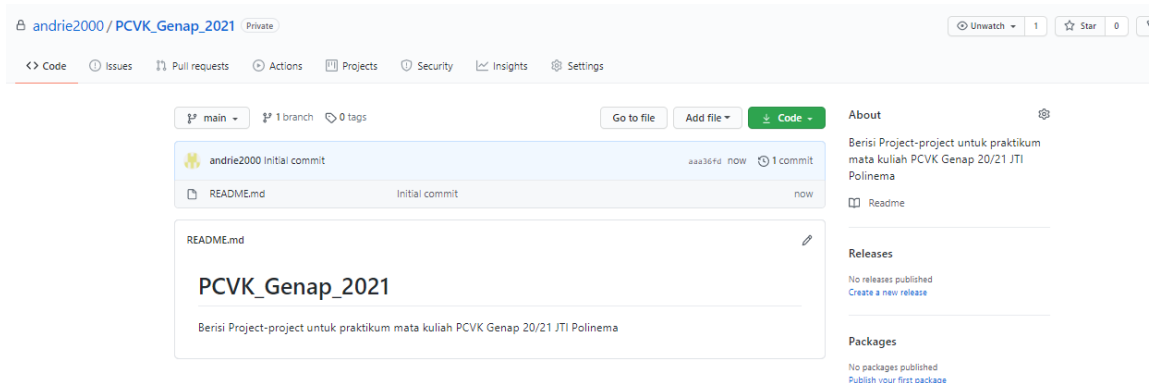
☐ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

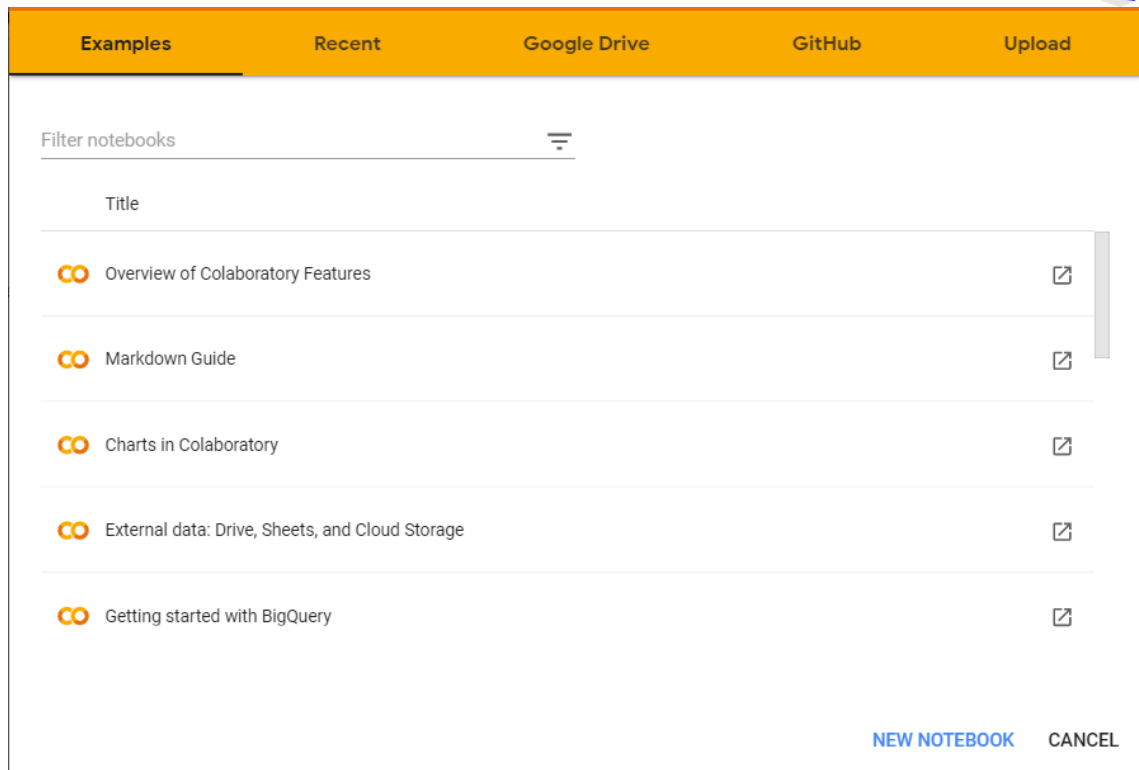
[Create repository](#)

- Repository telah siap diisikan dengan project-project untuk perkuliahan PCVK Genap 20/21.

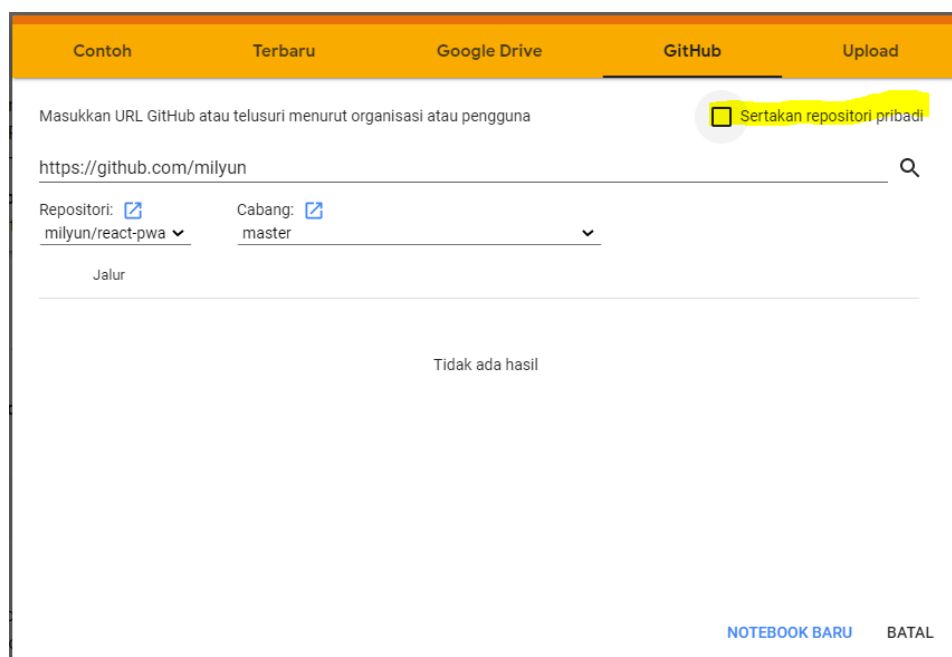


- Pada praktikum pengolahan citra dan visi komputer, kita akan melakukan *editing* dan *build code* untuk pengolahan citra menggunakan *Google Colaboratory*. *Google Colaboratory* dapat dicari menggunakan *search engine* atau dapat langsung dibuka pada link berikut: <https://colab.research.google.com/>

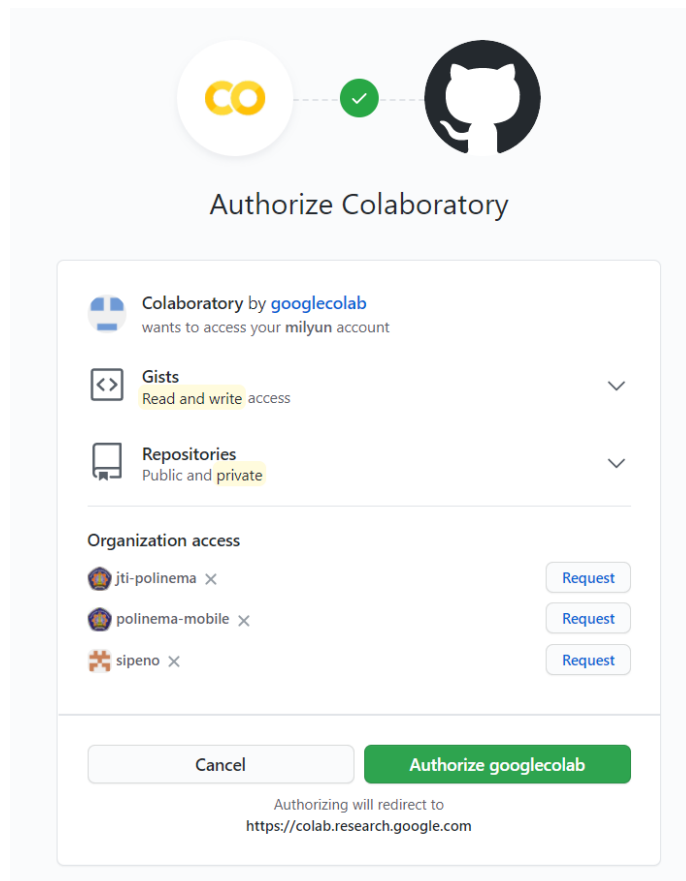
Tampilan dari jendela utama adalah sebagai berikut:



5. Aplikasi baru akan dibuat dalam format notebook python (tipe file adalah *\*.ipynb*), dimana file tersebut dapat disimpan pada drive cloud Google Colab, Google Drive, ataupun dapat terhubung pada Github secara langsung. Pada Kuliah ini kita akan menggunakan Github. Pilih menu Github pada jendela utama, kemudian lanjutkan untuk terhubung dengan Github personal anda.



Masukkan URL akun github Anda ([https://github.com/nama\\_akun](https://github.com/nama_akun)), kemudian tekan enter atau tekan icon search, kemudian akan muncul daftar repositori dan cabang yang terdapat pada akun Anda. Jika repository yang dibuat pada tahap 2 bersifat private, maka pilih checkbox “Sertakan repositori pribadi”, sehingga muncul jendela untuk memberikan otorisasi pada Google Colab, dan tekan tombol “*Authorize googlecolab*”



6. Setelah google colab terhubung dengan Github, Anda bisa memilih repositori dan membuat notebook baru.

ContohTerbaruGoogle DriveGitHubUpload

Masukkan URL GitHub atau telusuri menurut organisasi atau pengguna

☒ Sertakan repositori pribadi

milyun

Repositori: [mil](#)

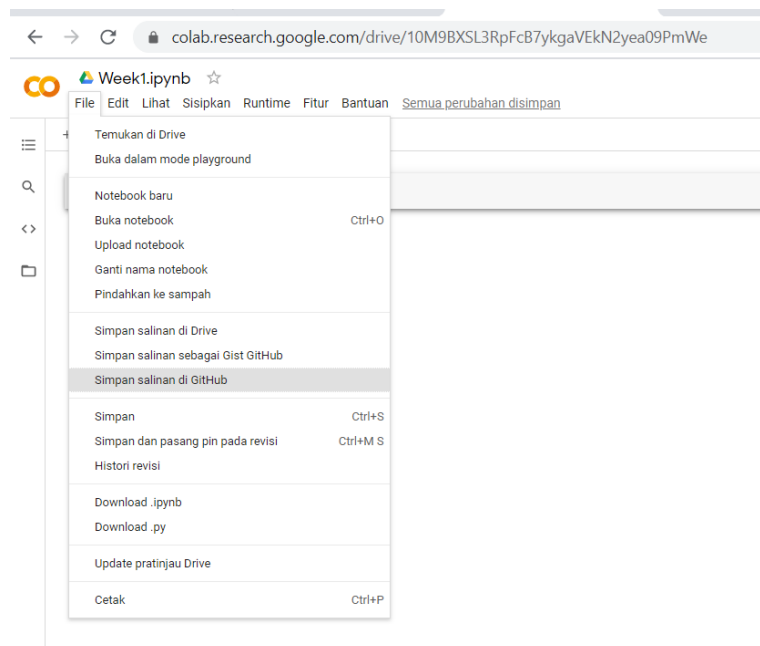
Cabang: [main](#)

Jalur

Tidak ada hasil

NOTEBOOK BARUBATAL

Ganti nama file (\*.ipynb), kemudian pilih File → Simpan Salinan ke Github



Sehingga akan muncul jendela seperti berikut:

Salin ke GitHub

Repositori: [mil](#)

Cabang: [main](#)

Jalur file

Week1.ipynb

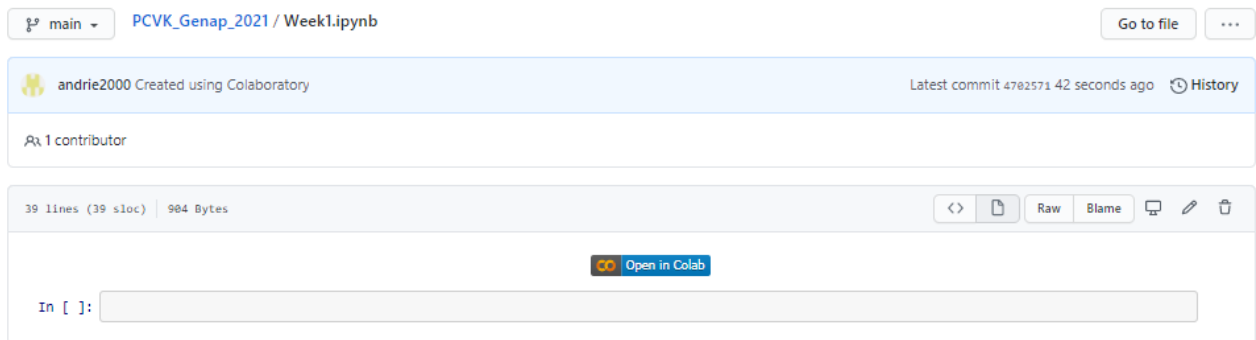
Pesan perubahan

Dibuat menggunakan Colaboratory

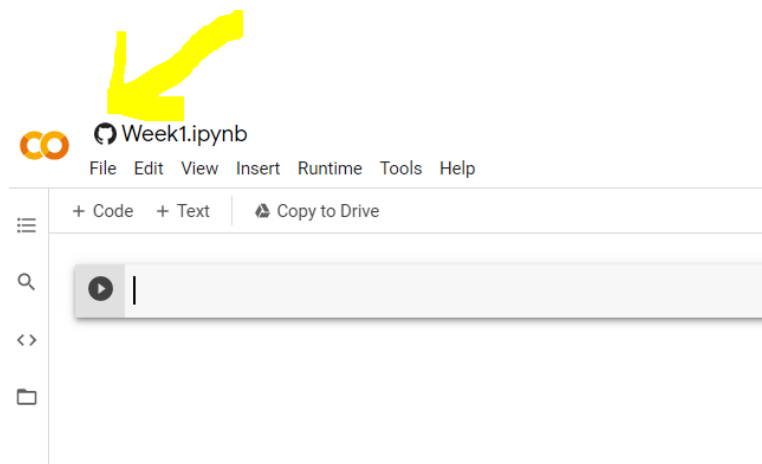
☒ Sertakan link ke Colaboratory

BATALOKE

- Setelah menekan tombol “OKE” maka akan langsung terbuka halaman file notebook yang baru saja dibuat pada repositori Github.



Di pojok kiri atas dari Google Colab perhatikan jika icon Github sudah muncul, tandanya file tersebut berhasil diakses oleh Google Colab.



File notebook dapat berisi catatan terformat dan berisi code python yang bisa langsung di running secara langsung. Tombol “+ Code” untuk menambahkan code program, dan “+ Text” digunakan untuk menambahkan catatan terformat.

## D2. Pengenalan Python untuk membaca dan menampilkan image dengan pengenalan macam-macam Library (Numpy, pandas, cv2, skimage, PIL dan matplotlib)

- Gunakan beberapa library berikut sebagai langkah pertama:

## ▼ Langkah (Step) 1:

Pada code awal, beberapa library yang akan digunakan adalah: **numpy**, **pandas**, **cv2**, **skimage**, **PIL**, **matplotlib**

--Bahasa--

--English--

- [Numpy](#) is an array manipulation library, used for linear algebra, Fourier transform, and random number capabilities.
- [Pandas](#) is a library for data manipulation and data analysis.
- [CV2](#) is a library for computer vision tasks.
- [Skimage](#) is a library which supports image processing applications on python.
- [Matplotlib](#) is a library which generates figures and provides graphical user interface toolkit.

```
[9] import numpy as np
import pandas as pd
import cv2 as cv
from google.colab.patches import cv2_imshow # for image display
from skimage import io
from skimage import transform
from PIL import Image
import matplotlib.pyplot as plt
```

## 2. Langkah 2 untuk membaca dan menampilkan image

### Langkah (Step) 2:

Baca Image dari URLs - Read Image From URLs

--Bahasa--

Pada Langkah ini kita akan membaca image dari URLs, menampilkannya menggunakan OpenCV, perhatikan ada perbedaan ketika membuka image dalam RGB dan BGR. Channel Warna default untuk OpenCV adalah BGR.

--English--

In this step we will read images from urls, and display them using openCV, please note the difference when reading image in RGB and BGR format. The default input color channels are in BGR format for openCV. RGB?

```
# Membuat list untuk menyimpan url dari beberapa image
urls = ["https://i11f.lib.ncsu.edu/iiif/0052574/full/800,/0/default.jpg",
        "https://i11f.lib.ncsu.edu/iiif/0016007/full/800,/0/default.jpg",
        "https://placekitten.com/800/571"]

# baca dan tampilkan image
# loop pada tiap url image, beberapa image dapat disimpan pada list
for url in urls:
    image = io.imread(url) #read image
    image = cv.resize(image, (0,0), fx=0.5, fy=0.5) #resize image to half size
    image_2 = cv.cvtColor(image, cv.COLOR_BGR2RGB) #convert color to RGB
    final_frame = cv.hconcat((image, image_2)) #concatenate image
    cv2_imshow(final_frame) #show image
    print('\n')
```

Pada tahap 2 kita akan membuat sebuah list untuk menyimpan URL beberapa citra, dimana untuk setiap citra akan dilakukan: pembacaan citra, resize ukuran citra menjadi setengahnya, konversi citra berwarna menjadi format RGB, menggabungkan citra asli dan citra hasil konversi, dan yang terakhir adalah menampilkan citra tersebut. Untuk url citra yang digunakan silahkan copy paste URL berikut, atau Anda juga bisa menggunakan URL citra yang lain:

urls = ["<https://iiif.lib.ncsu.edu/iiif/0052574/full/800,/0/default.jpg>", "<https://iiif.lib.ncsu.edu/iiif/0016007/full/800,/0/default.jpg>", "<https://placekitten.com/800/571>"]

Hasil dari potongan kode di atas adalah sebagai berikut:



3. Langkah 3 pada praktikum ini adalah melihat ukuran file image, dengan cara sebagai berikut:



### Langkah (Step) 3

melihat ukuran File image

```
[51] tinggi = image_2.shape[0]
    lebar = image_2.shape[1]
    print("resolusi image: tinggi x lebar = ",tinggi," x ",lebar)
    cv2_imshow(image_2)
```

resolusi image: tinggi x lebar = 286 x 400



4. Langkah 4 berikut digunakan untuk mengakses pixel dengan memberikan garis horizontal berwarna putih di tengah image

### Langkah (Step) 4

mengakses pixel dengan memberi garis diagonal menyilang

```
image_2 = cv.cvtColor(image, cv.COLOR_BGR2RGB)
image_3 = cv.cvtColor(image, cv.COLOR_BGR2RGB)

#membuat garis horizontal ditengah image
for y in range(lebar):
    image_3[int((tinggi)/2),y] = [255,255,255]

final_frame = cv.hconcat((image_2, image_3))
cv2_imshow(final_frame)
```



## PERTANYAAN PRAKTIKUM D2

1. Jelaskan, mengapa pada modul praktikum ini eksekusi kode Python dilakukan menggunakan Google Colab?
2. Jelaskan mengenai kegunaan setiap library pada praktikum langkah ke delapan? Apakah semua library tersebut harus digunakan dalam praktikum sesi ini?

3. Pada uji coba langkah ke-9 terdapat potongan kode program sebagai berikut :

```
image = cv.resize(image, (0,0), fx=0.5, fy=0.5)
```

Apa kegunaan kode program tersebut? dan apa pengaruhnya jika tidak dilakukan?

4. Perhatikan potongan kode program berikut :

```
#membuat garis horizontal ditengah image  
for y in range (lebar):  
    image_3[int((tinggi)/2),y] = [255,255,255]
```

Apakah kegunaan kode [255,255,255] ? Jelaskan!

5. Jelaskan keterkaitan antara pixel dan juga resolusi gambar yang tinggi ataupun rendah!

## TUGAS PRAKTIKUM D2

1. Lakukan langkah-langkah praktikum seperti diatas
2. Buat garis vertikal dan garis menyilang diagonal pada image keluaran

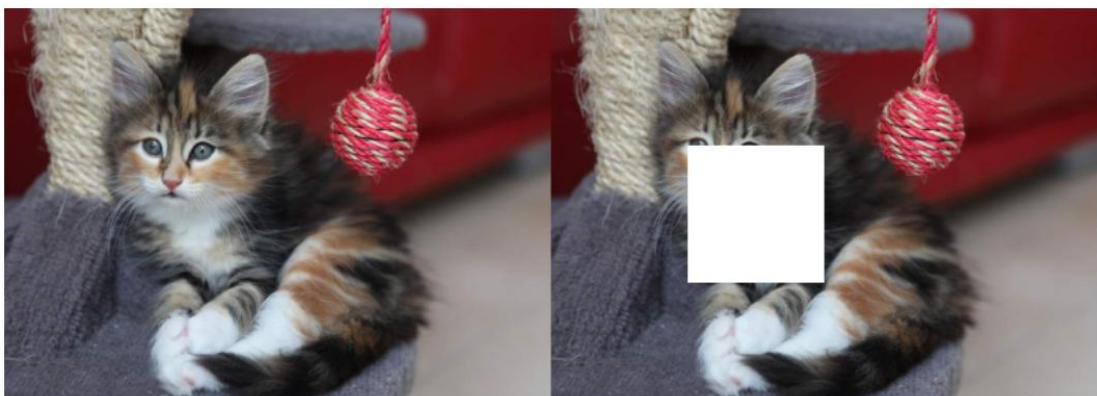




3. Buat garis horisontal berwarna putih dibagian tengah gambar dengan panjang tertentu

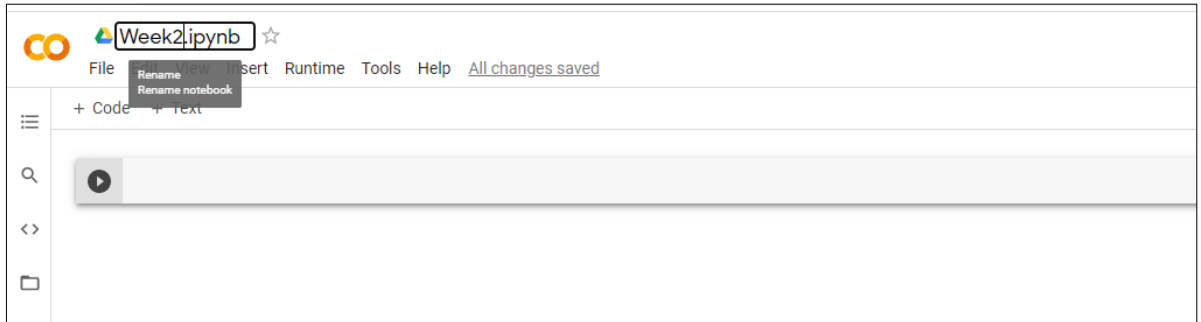


4. Buat kotak menggunakan kumpulan pixel warna putih di sembarang tempat dalam gambar

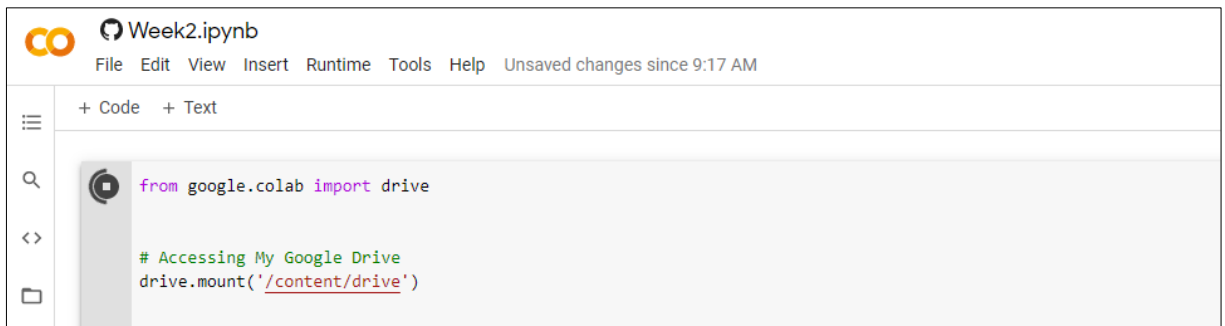


### D3. Pengenalan Python: Pengolahan Citra Dasar dan memahami channel warna pada OpenCV dan konversinya

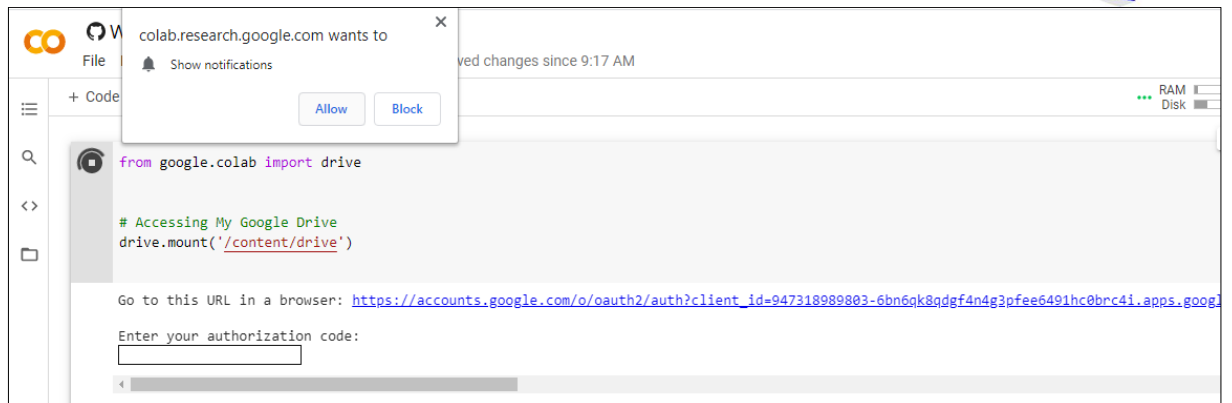
1. Buka <https://colab.research.google.com/>. Setelah dipastikan bahwa google Colab terhubung dengan Github Anda, lanjutkan dengan memilih repository yang telah digunakan pada praktikum D2, rename file menjadi “Week1b.ipynb”.



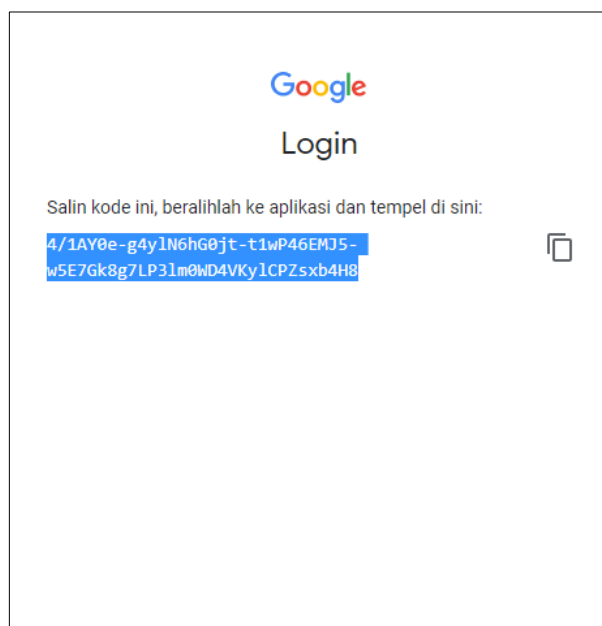
Kemudian import folder yang ada di Drive Anda dengan cara sebagai berikut.



Proses sinkronisasi google colab untuk terhubung dengan gdrive membutuhkan sedikit perubahan setting notifikasi seperti gambar berikut. Kemudian akan muncul suatu URL yang akan mengarah ke new tab untuk login akun google. Pada tahap ini dibutuhkan kode otorisasi yang muncul setelah proses login dilakukan.



Setelah proses login dilakukan, maka salin kode autorisasi dan tempelkan pada field yang sudah disediakan pada google Colab.



Jika berhasil, maka Google Colab sudah dapat mengakses folder gdrive Anda dengan keterangan output program "Mounted at /content/drive".



```
Week2.ipynb
File Edit View Insert Runtime Tools Help Unsaved changes since 9:17 AM

+ Code + Text

[11] from google.colab import drive

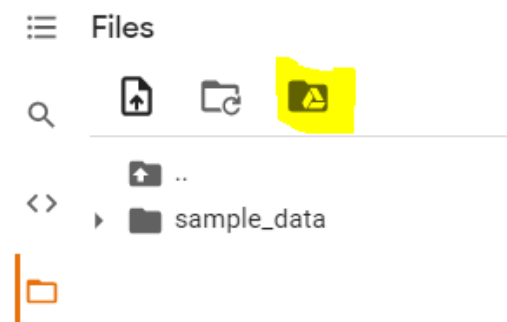
# Accessing My Google Drive
drive.mount('/content/drive')

Mounted at /content/drive
```

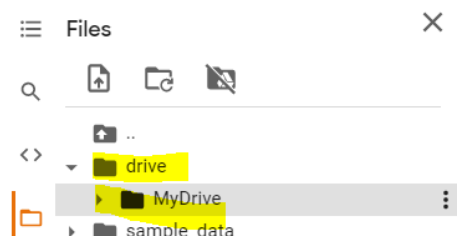
Cara yang kedua dengan mengklik tombol folder di sebelah kiri



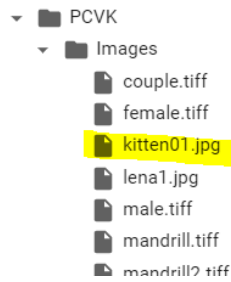
Pilih tombol gdrive -> Connect to Google Drive



Drive akan muncul dalam list folder



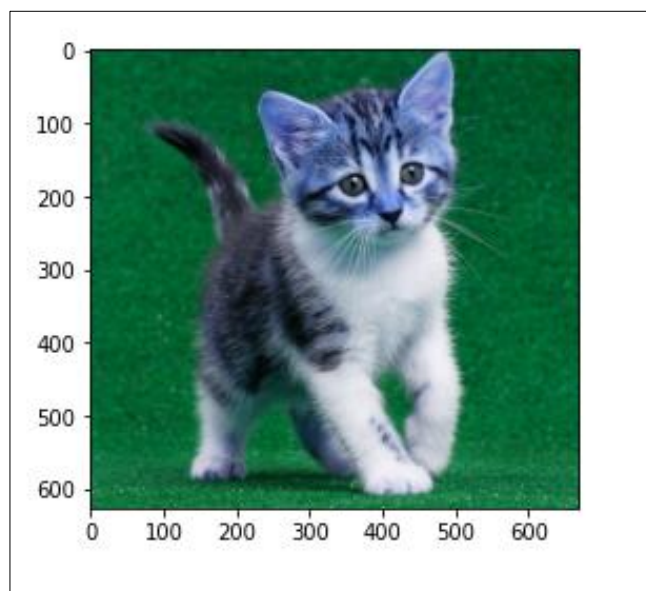
Pilih file dan klik icon titik 3 disebelah kanan file tersebut. klik copy path:



Kemudian kode program dapat dilanjutkan dengan membuka file Google Drive yang sudah ada. Contohnya pada kode program di bawah file image dengan nama kitten01.jpg akan dibuka untuk diproses lebih lanjut.

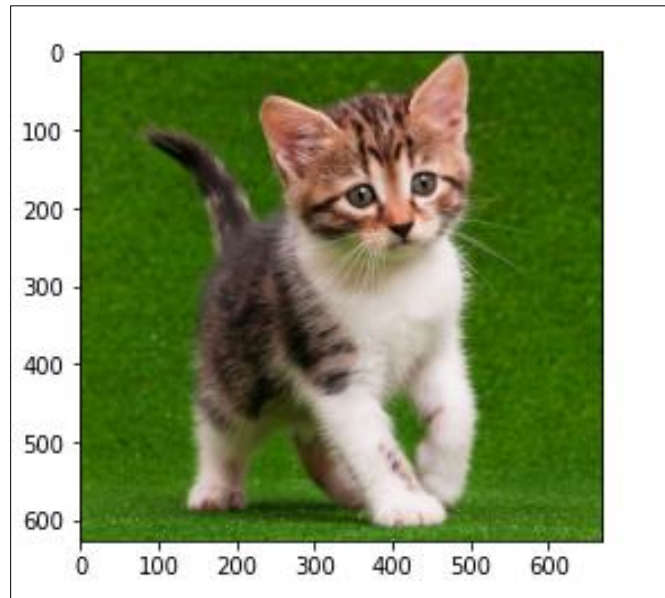
```
import cv2 as cv
from google.colab.patches import cv2_imshow
from skimage import io
import matplotlib.pyplot as plt
import numpy as np
img = cv.imread('/content/drive/MyDrive/PCVK_MM/Images/kitten01.jpg')
plt.imshow(img) #perhatikan hasilnya adalah citra dgn channel warna BGR
```

Hasil nya berupa output gambar yang sudah di plot dengan matplotlib untuk mengetahui ukuran panjang dan lebar dari gambar tersebut.



2. OpenCV membaca image dan menyimpan dalam channel warna BGR (Blue Green Red) Tampilkan nilai pixel dan image

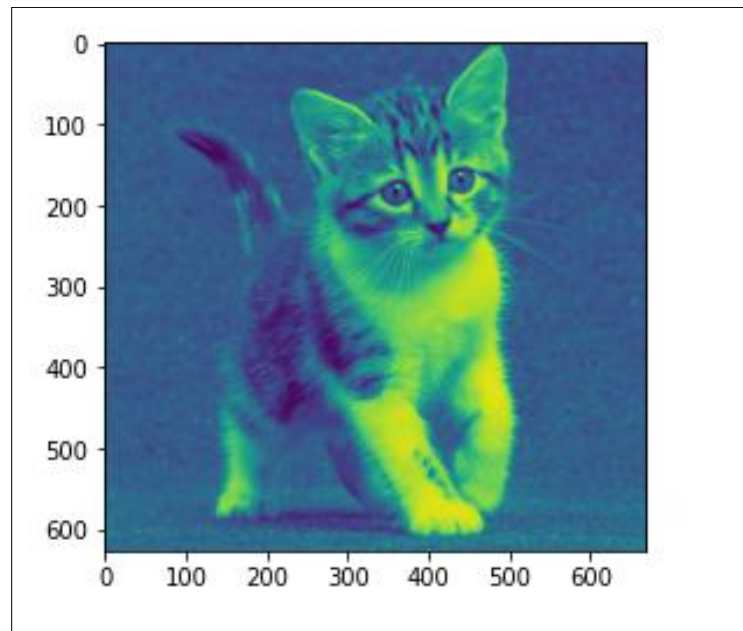
```
img2 = img  
img3 = cv.cvtColor(img,cv.COLOR_BGR2RGB)    #konversi channel BGR -> RGB  
plt.imshow(img3)
```



3. Menampilkan citra Grayscale, melakukan resizing, melakukan Flipping,  
Menyimpan citra hasil

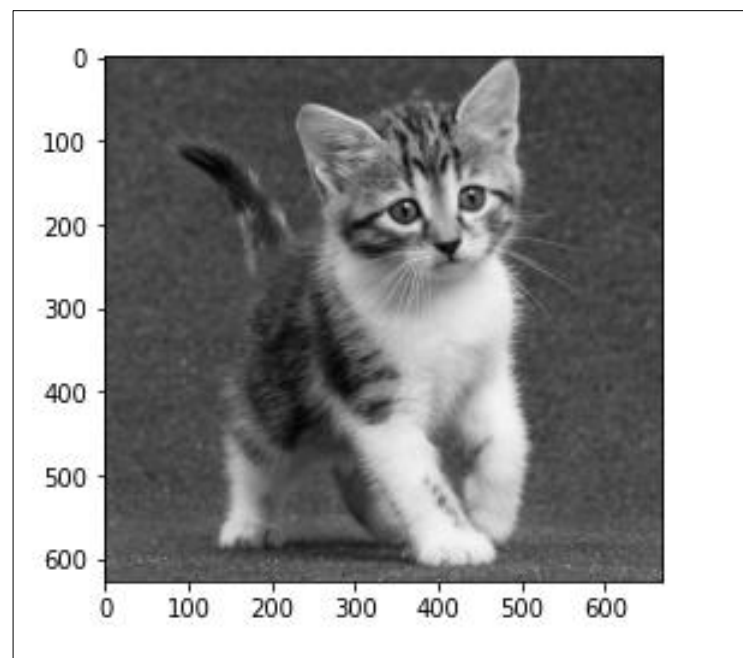
```
img_gray = cv.imread('/content/drive/MyDrive/PCVK_MM/Images/kitten01.jpg', cv.IMREAD_GRAYSCALE)  
plt.imshow(img_gray)
```





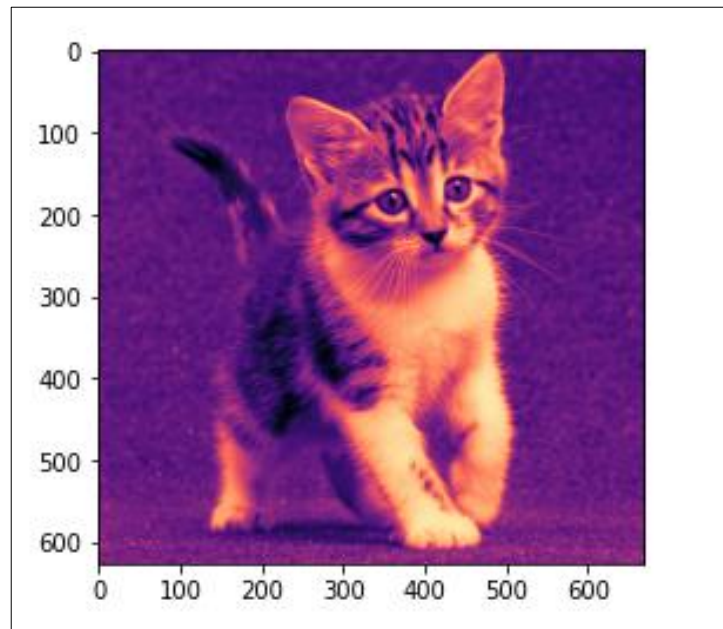
Dari citra grayscale, selanjutnya dipilih untuk ditampilkan colormap dengan warna 'gray'

```
plt.imshow(img_gray, cmap='gray')
```



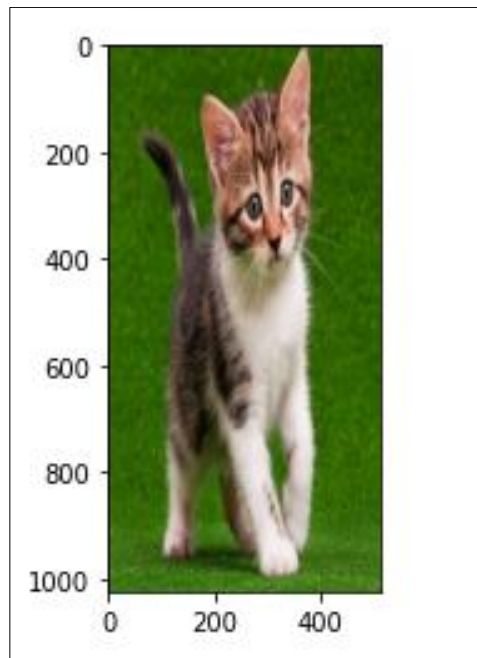
Dari citra grayscale, selanjutnya dipilih untuk ditampilkan colormap dengan warna 'magma'

```
plt.imshow(img_gray, cmap='magma')
```



Citra RGB di resize dalam ukuran panjang 512 dan lebar 1024

```
img4 = cv.resize(cv.cvtColor(img,cv.COLOR_BGR2RGB), (512,1024))  
plt.imshow(img4)
```



Citra RGB ditampilkan dalam ukuran yang lebih besar dengan posisi gambar terbalik

```
img5 = cv.flip(cv.cvtColor(img,cv.COLOR_BGR2RGB),0)

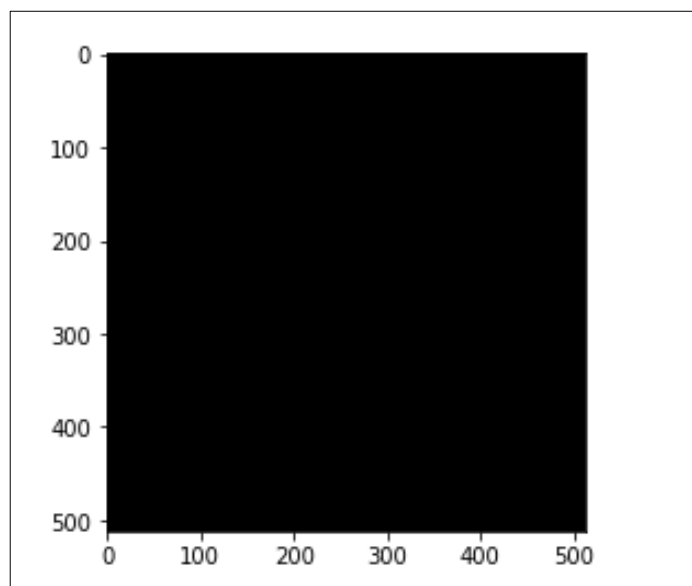
#simpan File image
#cv.imwrite('/content/drive/MyDrive/PCVK_MM/Images/mandrill2.tiff',img5)

#tampilkan plot dengan ukuran canvas yg lebih besar
fig = plt.figure(figsize=(10,10))      #ubah-ubah ukuran (10,10) sesuai kebutuhan
ax = fig.add_subplot(111)
ax.imshow(img5)
```



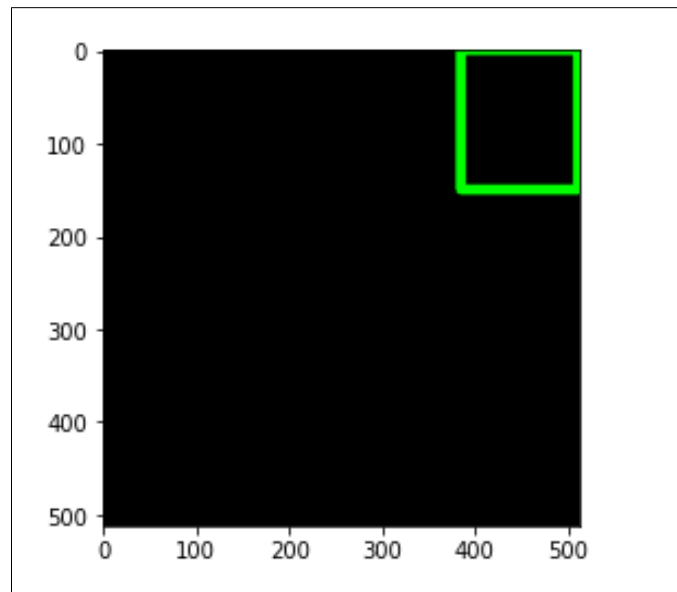
4. Membuat bentuk Geometri 2D dari OpenCV. Diawali dengan pembuatan black image dengan tipe data int16.

```
black_img = np.zeros(shape=(512,512,3),dtype=np.int16)  
plt.imshow(black_img)
```



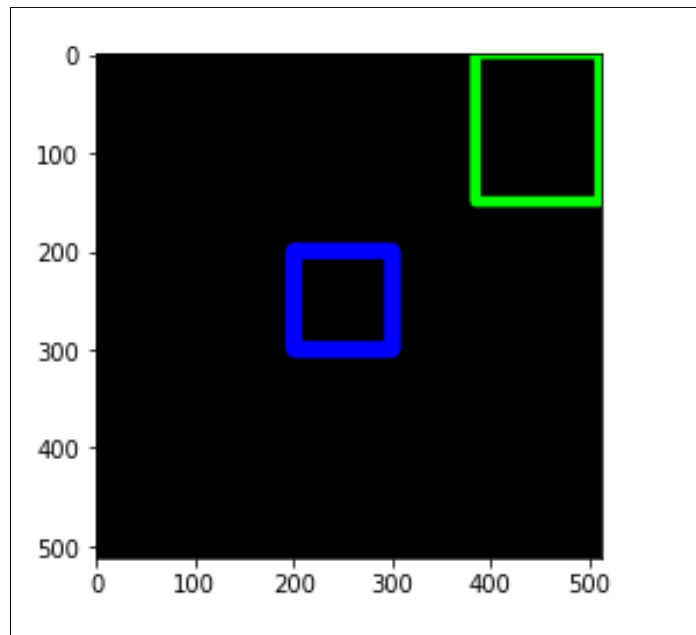
Kemudian menambahkan bentuk persegi panjang sesuai koordinat pt1 dan pt2

```
#perhatikan koordinat titik2 pt1 dan pt2  
cv.rectangle(black_img,pt1=(384,0),pt2=(510,150),color=(0,255,0),thickness=10)  
plt.imshow(black_img)
```



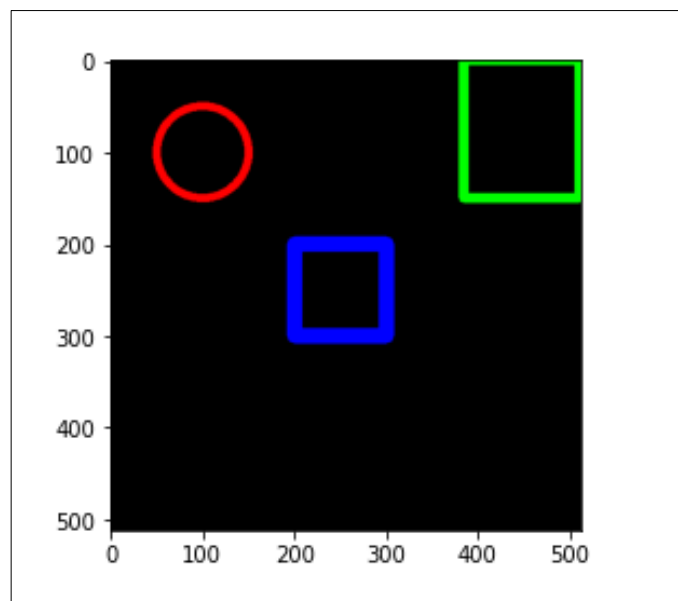
Selanjutnya ditambah menambahkan bentuk persegi sesuai koordinat pt1 dan pt2 yang tertulis pada kode program.

```
cv.rectangle(black_img,pt1=(200,200),pt2=(300,300),color=(0,0,255),thickness=15)  
plt.imshow(black_img)
```



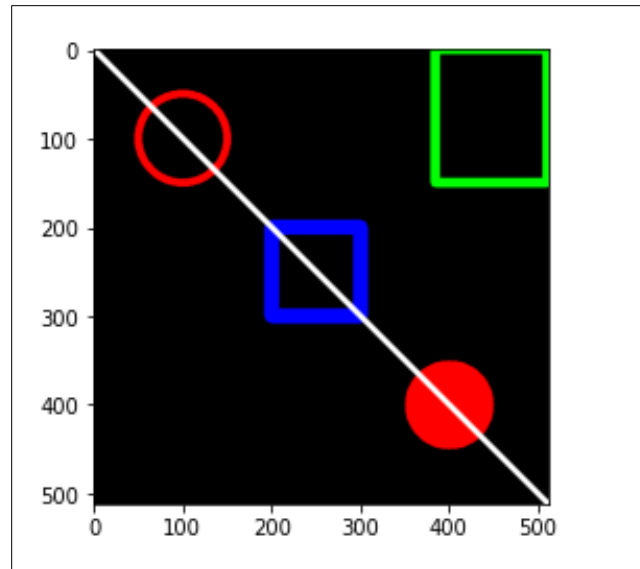
Tahap selanjutnya ditambah menambahkan bentuk lingkaran sesuai radius yang tertulis pada kode program.

```
cv.circle(black_img,center=(100,100),radius=50,color=(255,0,0),thickness=8)  
plt.imshow(black_img)
```



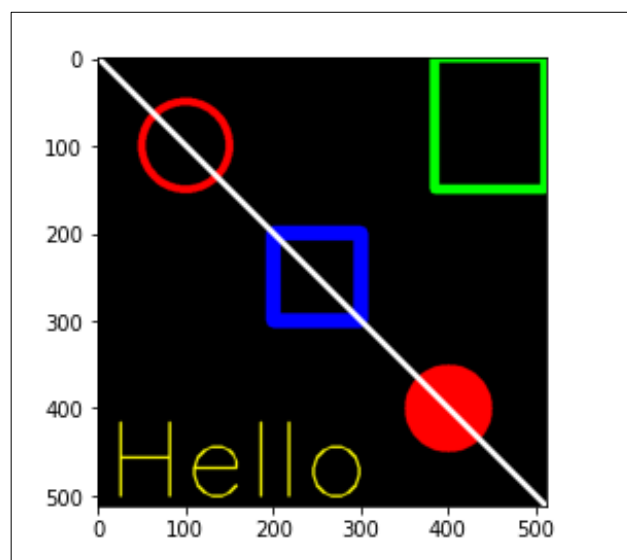
Kemudian dilakukan penambahan garis sesuai koordinat pt1 dan pt2 sebagai berikut.

```
cv.line(black_img,pt1=(0,0),pt2=(512,512),color=(255,255,255),thickness=5)  
plt.imshow(black_img)
```



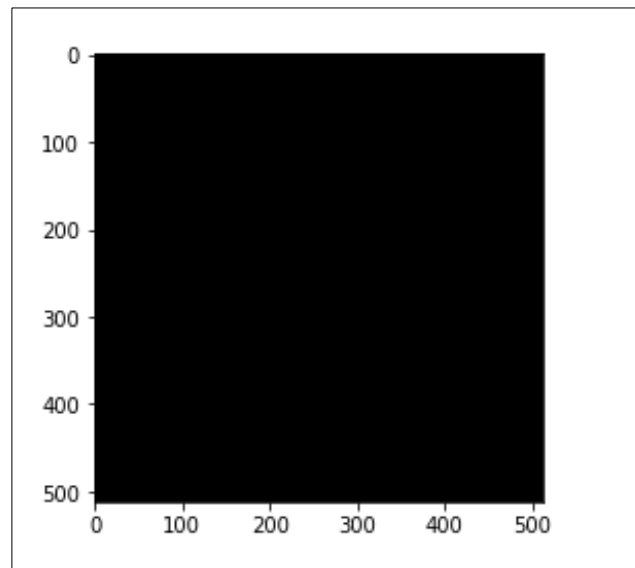
Penambahan text dengan font yang telah tertulis dengan ukuran yang sudah ditentukan.

```
font = cv.FONT_HERSHEY_SIMPLEX  
cv.putText(black_img,text='Hello',org=(10,500),fontFace=font,fontScale=4,color=(255,255,0),thickness=2,lineType=cv.LINE_AA)  
plt.imshow(black_img)
```



```
black_img2=np.zeros(shape=(512,512,3),dtype=np.int32)
plt.imshow(black_img2)
```

Pembuatan black image kembali dilakukan dengan tipe data int32



Berikut adalah kode program untuk inisialisasi NumPy array dengan tipe data int32

```
vertices = np.array([[100,300],[200,200],[400,300],[200,400]],dtype=np.int32)
vertices
```

```
array([[100, 300],
       [200, 200],
       [400, 300],
       [200, 400]], dtype=int32)
```

Array tersebut kemudian di reshape sebagai berikut

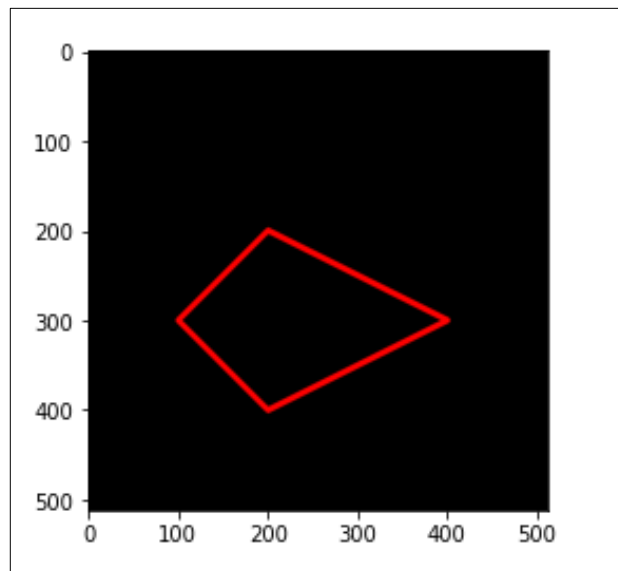
```
pts = vertices.reshape((-1,1,2)) # nilai 2 untuk menunjukkan bahwa tiap titik dibuat 3 channel yg mewakili R, G, dan B
pts
```



```
array([[100, 300],
       [200, 200],
       [400, 300],
       [200, 400]], dtype=int32)
```

Penambahan polyline pada black image kedua yang telah dibuat.

```
cv.polylines(black_img2,[pts],isClosed=True,color=(255,0,0),thickness=5)
plt.imshow(black_img2)
```



- Setelah semua kode selesai simpan “Week2.ipynb” pada GitHub Anda dengan memilih File kemudian “Save a copy in GitHub”.

### PERTANYAAN PRAKTIKUM D3

- Apakah perbedaan gambar yang ditampilkan tanpa dan dengan matplotlib?
- Apakah perbedaan dan pengaruhnya pembuatan black image antara tipe data int16 dan int32?
- Apakah kegunaan “google.colab.patches import cv2\_imshow” pada potongan kode berikut

```
from google.colab.patches import cv2_imshow
from skimage import io
```

4. Apakah kegunaan “skimage import io” pada potongan kode soal nomor 3

## TUGAS PRAKTIKUM D3

Berdasarkan praktikum bagian 1 dan 2 kerjakan beberapa tugas berikut :

1. Dengan menggunakan figsize, perhatikan apakah ukuran image pixelnya juga berubah?
2. Tampilkan image dalam channel Red-Blue dan Green-Blue saja!
3. Tampilkan image baris ke 20-115, kolom 25-120!
4. Tampilkan image baris ke 5-30, semua kolom, channel Red saja!
5. Buat 5 kotak berbagai ukuran dan warna yang berbeda dalam satu image. disarankan menggunakan bilangan acak/random!
6. Tampilkan image dengan posisi terbalik!

Berdasarkan praktikum bagian 3 dan 4 kerjakan beberapa tugas berikut :

7. Buat rectangle dan circle pada bagian wajah dari image foto s anda saat beraktifitas (bukan pasfoto).
8. Buat rectangle pada bagian sudut bawah kiri channel B pada color space RGB dari citra kitten/ lena/ mandrill/ male/ female/ couple/ sailboat/ peppers!
9. Lengkapi tulisan nama file pada file citra dari soal no.8. gunakan font, ukuran font, dan warna font yang sesuai keinginan anda.
10. Tunjukkan code program anda pada bapak/ibu dosen

## TUGAS PRAKTIKUM KELOMPOK

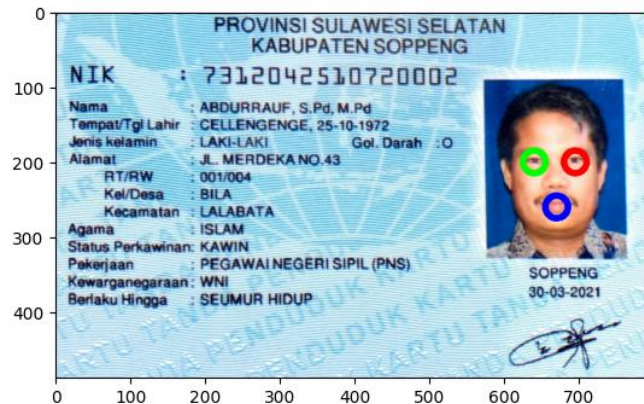
Bentuk kelompok masing-masing terdiri dari 2-3 orang. tiap kelompok lakukan langkah berikut:

1. Akses file image lokal (KTM) dan tampilkan menggunakan OpenCV dan Matplotlib

2. Tutup tiap bagian tertentu dari KTM tersebut menggunakan fungsi yang telah Anda pelajari. Kreasikan terkait dengan warna dan ukuran dari bentuk-bentuknya.

- Kelompok 1 bagian NIM dan nama
- Kelompok 2 bagian TTL dan PRODI
- Kelompok 3 bagian tulisan POLITEKNIK NEGERI MALANG dan DIREKTUR
- Kelompok 4 tulisan Kartu Tanda Mahasiswa dan POLINEMA
- Kelompok 5 bagian Nama dan LOGO POLINEMA
- Kelompok 6 logo Polinema dan tulisan POLINEMA
- Kelompok 7 bagian Malang, bulan tahun dan NIP Direktur
- Kelompok 8 bagian FOTO dan tanda tangan
- Kelompok 9 bagian Kementrian dan seluruh identitas mahasiswa
- Kelompok 10 tulisan kode barcode dan kota mahasiswa

Tutup menggunakan 2 warna yang berbeda dan bentuk berbeda.



--- SELAMAT BELAJAR ---