



# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

---

### MODUL 4

### SINYAL GENERATOR

#### 1. Tujuan Praktikum

1. Memahami pengertian dan konsep dasar gambar digital pada computer vision
2. Mempelajari image processing serta operasi dasar pengolahannya.
3. Mampu merancang dan memvisualisasikan berbagai jenis format image processing menggunakan Python.

#### 2. Alat dan Bahan

1. Laptop/PC
2. Google Colab
3. Library

Untuk pengolahan suara:

- Pydub,
- Numpy,
- Matplotlib,
- OpenCV

Untuk pengolahan gambar:

- PIL (Pillow),
- opencv-python,
- matplotlib

#### 3. Materi

##### 3.1. OpenCv

OpenCV (Open Source Computer Vision Library) merupakan salah satu pustaka visi komputer yang paling populer dan banyak digunakan di dunia. Sejak diluncurkan pada tahun 2000, OpenCV telah menjadi standar de facto dalam pengembangan aplikasi visi komputer, baik untuk tujuan akademis maupun industri. Pustaka ini menyediakan berbagai alat dan fungsi yang memungkinkan pengembang untuk memproses gambar dan video secara efisien, serta mengimplementasikan algoritma visi komputer yang canggih. Dalam modul ini, kami akan membahas konsep-konsep dasar OpenCV yang esensial untuk memahami fungsionalitas intinya, khususnya dalam konteks penggunaan modul Python.



# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

### 3.1.1. Penggunaan OpenCV

#### - Membaca Gambar

Fungsi dasar pertama dalam OpenCV adalah membaca gambar dari sebuah file. Gambar direpresentasikan sebagai matriks multidimensi, di mana setiap elemen matriks merepresentasikan nilai intensitas piksel. Dalam OpenCV, gambar biasanya dibaca dalam format BGR (Blue, Green, Red), bukan RGB, yang merupakan konvensi yang perlu diperhatikan. Fungsi `cv2.imread()` digunakan untuk membaca gambar dari file, dan hasilnya adalah array NumPy yang merepresentasikan gambar tersebut.

```
import cv2
gambar = cv2.imread('path_ke_gambar.jpg')
```

#### - Mengekstraksi Nilai RGB dari Sebuah Piksel

Setiap piksel dalam gambar memiliki nilai intensitas untuk setiap saluran warna (BGR). Untuk mengakses nilai intensitas ini, kita dapat menggunakan indeks array. Misalnya, untuk mengakses nilai BGR dari piksel di koordinat (x, y), kita dapat menggunakan sintaks berikut:

```
(b, g, r) = gambar[y, x]
```

#### - Mengekstrak Wilayah Minat (Region of Interest - ROI)

Wilayah Minat (ROI) adalah bagian tertentu dari gambar yang menjadi fokus analisis. Dalam OpenCV, ROI dapat diekstrak dengan melakukan slicing pada array gambar. Misalnya, untuk mengekstrak ROI dari koordinat (x1, y1) hingga (x2, y2), kita dapat menggunakan sintaks berikut:

```
roi = gambar[y1:y2, x1:x2]
```

ROI sangat berguna dalam aplikasi seperti deteksi objek, pelacakan, dan segmentasi.



# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

### - Mengubah Ukuran Gambar

Mengubah ukuran gambar adalah operasi dasar yang sering dilakukan dalam visi komputer. OpenCV menyediakan fungsi `cv2.resize()` untuk mengubah dimensi gambar. Fungsi ini memungkinkan pengguna untuk menentukan ukuran baru atau faktor skala.

```
gambar_resized = cv2.resize(gambar, (lebar_baru, tinggi_baru))
```

Operasi ini penting untuk memastikan gambar memiliki ukuran yang konsisten sebelum diproses lebih lanjut.

### - Memutar Gambar

Memutar gambar adalah operasi geometris yang sering digunakan dalam pra-pemrosesan gambar. OpenCV menyediakan fungsi `cv2.rotate()` untuk memutar gambar dengan sudut tertentu. Selain itu, kita juga dapat menggunakan matriks transformasi untuk melakukan rotasi yang lebih kompleks.

```
gambar_rotated = cv2.rotate(gambar, cv2.ROTATE_90_CLOCKWISE)
```

### - Menggambar Persegi Panjang

Menggambar bentuk geometris seperti persegi panjang pada gambar adalah operasi yang umum digunakan untuk menandai objek atau wilayah tertentu. OpenCV menyediakan fungsi `cv2.rectangle()` untuk menggambar persegi panjang. Fungsi ini memerlukan koordinat titik awal dan akhir, serta warna dan ketebalan garis.

```
cv2.rectangle(gambar, (x1, y1), (x2, y2), (0, 255, 0), 2)
```



# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

### - Menampilkan Teks

Menambahkan teks pada gambar adalah operasi yang berguna untuk memberikan informasi tambahan, seperti label atau hasil analisis. OpenCV menyediakan fungsi `cv2.putText()` untuk menampilkan teks pada gambar. Fungsi ini memerlukan koordinat titik awal, jenis font, skala, warna, dan ketebalan teks.

```
cv2.putText(gambar, 'Hello, World!', (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
```

Contoh hasil:

Before :



After :



**quiz block :**

**pada gambar after, termasuk penggunaan openCV dalam penggunaan?**

### 3.2. Reading image in OpenCV

Kita akan membahas cara membuka dan memproses gambar menggunakan pustaka OpenCV (Open Source Computer Vision). OpenCV adalah pustaka open-source yang sangat populer untuk visi komputer dan pemrosesan gambar. Pustaka ini mendukung berbagai format file gambar, sehingga memungkinkan pengguna untuk bekerja dengan berbagai jenis data visual. Berikut adalah format file yang didukung oleh OpenCV:



# PRAKTIKUM

## *PENGOLAHAN SINYAL INFORMASI*

- 1) Windows bitmaps – \*.bmp, \*.dib
- 2) JPEG files – \*.jpeg, \*.jpg
- 3) Portable Network Graphics – \*.png
- 4) WebP – \*.webp Sun rasters – \*.sr, \*.ras
- 5) TIFF files – \*.tiff, \*.tif
- 6) Raster dan Vector geospatial data yang didukung oleh GDAL

Untuk menggunakan OpenCV dalam Python, kita perlu menginstal beberapa library prasyarat. Dua library utama yang diperlukan adalah:

- 1) NumPy Library:

NumPy adalah library Python yang digunakan untuk komputasi numerik. Karena gambar diproses dalam bentuk matriks, NumPy digunakan oleh OpenCV di latar belakang untuk memanipulasi data gambar secara efisien.

- 2) OpenCV Python (cv2):

OpenCV menyediakan modul Python yang disebut cv2. Modul ini berisi fungsi-fungsi untuk memanipulasi gambar dan video. Perlu dicatat bahwa versi terbaru OpenCV menggunakan nama cv2, bukan cv.

### 3.3. Instalasi Library Prasyarat

```
pip install opencv-python
pip install numpy
pip install matplotlib
```

syntax ini bisa kalian running saat di google colab.

```
import cv2

# Membaca gambar dari file
gambar = cv2.imread('path_ke_gambar.jpg')

# Memeriksa apakah gambar berhasil dibaca
if gambar is None:
    print("Gambar tidak ditemukan atau format tidak didukung.")
else:
    # Menampilkan gambar dalam sebuah window
    cv2.imshow('Gambar', gambar)
```



## PRAKTIKUM

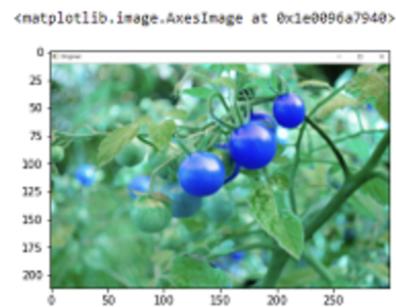
### PENGOLAHAN SINYAL INFORMASI

- `cv2.imread()` : Fungsi ini digunakan untuk membaca gambar dari file. Jika gambar tidak ditemukan atau format tidak didukung, fungsi akan mengembalikan None.
- `cv2.imshow()` : Fungsi ini menampilkan gambar dalam sebuah window. Parameter pertama adalah nama window, dan parameter kedua adalah objek gambar yang telah dibaca.
- `cv2.waitKey()` : Fungsi ini menunggu pengguna menekan tombol apa pun. Parameter 0 berarti program akan menunggu tanpa batas waktu hingga tombol ditekan.
- `cv2.destroyAllWindows()` : Fungsi ini menutup semua window yang terbuka setelah pengguna menekan tombol.

Berikut adalah contoh penggunaan Reading image



Gambar biasa



Menggunakan BGR color format dan matplotlib menggunakan RGB color format

#### 3.4. `cv2.imshow()` method

Library yang menyediakan binding Python untuk menyelesaikan masalah visi komputer. Metode `cv2.imshow()` digunakan untuk menampilkan gambar dalam sebuah jendela. Jendela ini secara otomatis menyesuaikan ukuran gambar yang ditampilkan. adalah library yang menyediakan binding Python untuk menyelesaikan masalah visi komputer. Metode `cv2.imshow()` digunakan untuk menampilkan gambar dalam sebuah jendela. Jendela ini secara otomatis menyesuaikan ukuran gambar yang ditampilkan.



# PRAKTIKUM

## *PENGOLAHAN SINYAL INFORMASI*

Contoh :



notes: Hijau gambar ori, Abu-abu setelah di processing.

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**

### 3.5. `cv2.imwrite()` method

library yang menyediakan binding Python untuk menyelesaikan masalah visi komputer. Metode `cv2.imwrite()` digunakan untuk menyimpan gambar ke perangkat penyimpanan. Metode ini akan menyimpan gambar sesuai format yang ditentukan di direktori kerja saat ini.

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**

### 3.6. Saving an Image

Menyimpan gambar adalah proses mempelajari cara menyimpan gambar dari satu lokasi ke lokasi lain yang diinginkan, seperti ke Google Drive, menggunakan OpenCV dalam Python. Dengan OpenCV, kita juga dapat membuat gambar kosong (blank image) dengan warna apa pun yang diinginkan. Mari kita pelajari lebih dalam dan pahami konsep ini dengan penjelasan lengkap.

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**

### 3.7. Color Spaces in OpenCV

#### 3.7.1. Ruang Warna (Color Spaces)

Ruang warna adalah cara untuk merepresentasikan saluran warna yang ada dalam gambar, yang memberikan corak tertentu pada gambar tersebut. Ada beberapa jenis ruang warna, dan masing-masing memiliki signifikansi tersendiri. Beberapa ruang

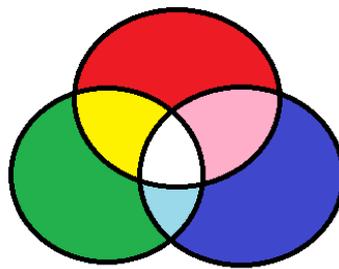


## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

warna yang populer antara lain RGB (Red, Green, Blue), CMYK (Cyan, Magenta, Yellow, Black), dan HSV (Hue, Saturation, Value).

- Ruang Warna BGR: OpenCV secara default menggunakan ruang warna RGB, tetapi sebenarnya menyimpan warna dalam format BGR (Blue, Green, Red). Ini adalah model warna aditif dimana intensitas yang berbeda dari Biru, Hijau, dan Merah menghasilkan berbagai corak warna.

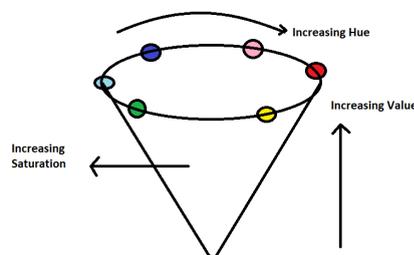


#### 3.7.2. Ruang Warna HSV

Ruang warna HSV menyimpan informasi warna dalam representasi silinder dari titik-titik warna RGB. Ruang ini mencoba menggambarkan warna sebagaimana yang dilihat oleh mata manusia.

1. Hue (Rona): Nilainya berkisar dari 0-179, merepresentasikan jenis warna (misalnya merah, hijau, biru).
2. Saturation (Saturasi): Nilainya berkisar dari 0-255, menunjukkan seberapa murni atau intens warna tersebut.
3. Value (Nilai): Nilainya berkisar dari 0-255, menunjukkan kecerahan atau kegelapan warna.

Ruang warna HSV sering digunakan untuk segmentasi warna karena kemampuannya memisahkan warna berdasarkan rona, saturasi, dan nilai.





## PRAKTIKUM

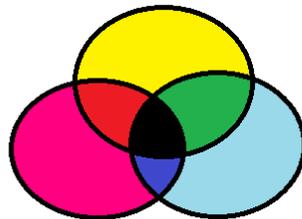
### *PENGOLAHAN SINYAL INFORMASI*

#### 3.7.3. Ruang Warna CMYK

Berbeda dengan RGB, CMYK adalah ruang warna subtraktif. Model CMYK bekerja dengan menutupi sebagian atau seluruh warna pada latar belakang yang lebih terang (biasanya putih). Tinta mengurangi cahaya yang seharusnya dipantulkan. Model ini disebut subtraktif karena tinta "mengurangi" warna merah, hijau, dan biru dari cahaya putih.

1. Cyan: Diperoleh ketika cahaya putih dikurangi merah.
2. Magenta: Diperoleh ketika cahaya putih dikurangi hijau.
3. Yellow: Diperoleh ketika cahaya putih dikurangi biru.
4. Black (K): Ditambahkan untuk meningkatkan kedalaman warna dan detail.

Ruang warna CMYK umumnya digunakan dalam pencetakan, karena lebih sesuai dengan cara tinta bekerja pada media fisik seperti kertas.



#### 3.8. Arithmetic Operations on Images using OpenCV (Set-1 (Addition and Subtraction))

Operasi aritmatika seperti Penjumlahan (Addition), Pengurangan (Subtraction), dan Operasi Bitwise (AND, OR, NOT, XOR) dapat diterapkan pada gambar input. Operasi ini berguna untuk meningkatkan properti gambar input, seperti memperjelas detail, meningkatkan kontras, atau menggabungkan informasi dari beberapa gambar. Operasi aritmatika pada gambar sangat penting untuk menganalisis properti gambar dan dapat digunakan sebagai langkah awal untuk operasi lanjutan seperti thresholding, dilasi, atau segmentasi.



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

#### 3.8.1. Penjumlahan Gambar (Image Addition)

Kita dapat menambahkan dua gambar menggunakan fungsi `cv2.add()`. Fungsi ini secara langsung menjumlahkan nilai piksel dari dua gambar. Namun, penjumlahan langsung seperti ini tidak selalu ideal karena dapat menyebabkan overflow (nilai piksel melebihi batas maksimum).

```
cv2.add(img1, img2)
```

#### 3.8.2. Penjumlahan Tertimbang (Weighted Addition)

Untuk menghindari masalah overflow dan memberikan kontrol lebih terhadap hasil penjumlahan, kita dapat menggunakan fungsi `cv2.addWeighted()`. Fungsi ini memungkinkan kita untuk memberikan bobot (weight) pada setiap gambar sebelum dijumlahkan. Kedua gambar harus memiliki ukuran dan kedalaman (depth) yang sama. Contoh Sintaks:

```
cv2.addWeighted(img1, alpha, img2, beta, gamma)
```

1. `img1`: Gambar pertama.
2. `alpha`: Bobot untuk gambar pertama.
3. `img2`: Gambar kedua.
4. `beta`: Bobot untuk gambar kedua.
5. `gamma`: Nilai skalar yang ditambahkan ke hasil penjumlahan.

#### 3.8.3. Operasi Bitwise

Operasi bitwise seperti AND, OR, NOT, XOR sangat berguna untuk manipulasi piksel pada level biner. Operasi ini sering digunakan dalam aplikasi seperti masking, menggabungkan gambar, atau mengekstrak area tertentu dari gambar. Contoh Sintaks:

1. AND: `cv2.bitwise_and(img1, img2)`
2. OR: `cv2.bitwise_or(img1, img2)`
3. NOT: `cv2.bitwise_not(img1)`
4. XOR: `cv2.bitwise_xor(img1, img2)`

```
# Operasi Bitwise AND  
bitwise_and = cv2.bitwise_and(img1, img2)
```



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

Contoh penggunaan pada image processing:

Input Image1:



Input Image2:



Output:



### 3.9. Arithmetic Operations on Images using OpenCV (Set-2 (Bitwise Operations on Binary Images))

Operasi bitwise adalah teknik penting dalam manipulasi gambar yang digunakan untuk mengekstrak bagian-bagian penting dari gambar atau membuat mask. Operasi ini meliputi AND, OR, XOR, dan NOT. Selain itu, operasi bitwise juga berguna dalam pembuatan gambar baru dan peningkatan properti gambar input.

**Catatan Penting:** Operasi bitwise hanya dapat diterapkan pada gambar-gambar yang memiliki dimensi (ukuran) yang sama.



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

#### 3.9.1. Operasi Bitwise yang Umum digunakan

##### 1. Bitwise AND (`cv2.bitwise_and`)

Operasi AND menggabungkan dua gambar dengan mempertahankan area di mana kedua gambar memiliki nilai piksel yang tidak nol. Operasi ini sering digunakan untuk masking, yaitu mengekstrak area tertentu dari gambar.

```
hasil_and = cv2.bitwise_and(img1, img2)
```

##### 2. Bitwise OR (`cv2.bitwise_or`)

Operasi OR menggabungkan dua gambar dengan mempertahankan area di mana salah satu atau kedua gambar memiliki nilai piksel yang tidak nol. Operasi ini berguna untuk menggabungkan dua gambar atau area gambar.

**Quiz block:**

**Dapatkan kamu membuat kode bitwise versi OR?**

##### 3. Bitwise XOR (`cv2.bitwise_xor`)

Operasi XOR menggabungkan dua gambar dengan mempertahankan area di mana hanya satu gambar yang memiliki nilai piksel yang tidak nol. Operasi ini berguna untuk menemukan perbedaan antara dua gambar.

**Quiz block:**

**Dapatkan kamu membuat kode bitwise versi XOR?**

##### 4. Bitwise NOT (`cv2.bitwise_NOT`)

Operasi NOT membalikkan nilai piksel pada gambar. Operasi ini berguna untuk membuat negatif dari gambar atau membalikkan mask

**Quiz block:**

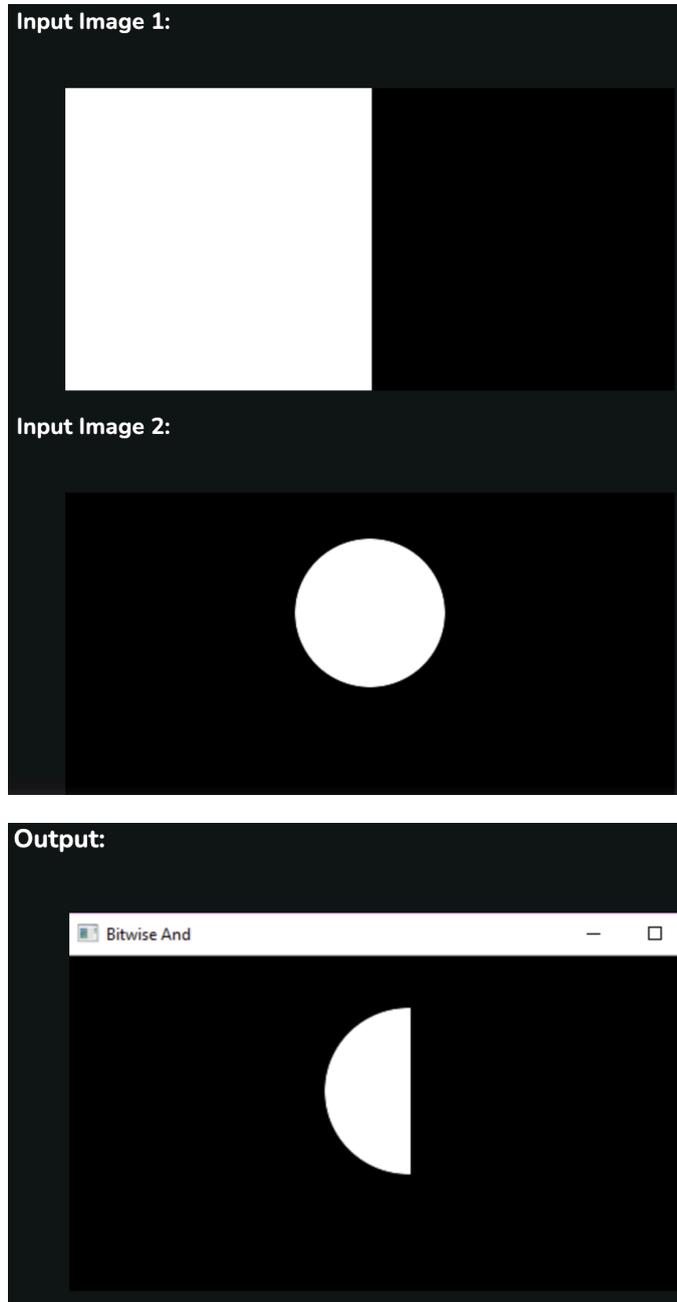
**Dapatkan kamu membuat kode bitwise versi NOT?**



# PRAKTIKUM

## *PENGOLAHAN SINYAL INFORMASI*

Contoh penggunaan pada image processing:





## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

#### 3.10. Mendeteksi Sudut pada Gambar dengan OpenCV

OpenCV (Open Source Computer Vision) adalah pustaka visi komputer yang menyediakan berbagai fungsi untuk melakukan operasi pada gambar atau video. Salah satu operasi yang dapat dilakukan adalah mendeteksi sudut (corner) pada gambar. Untuk mendeteksi sudut, OpenCV menyediakan metode `cv2.goodFeaturesToTrack()` yang menggunakan metode Shi-Tomasi untuk menemukan sudut-sudut terkuat dalam gambar.

##### 1. Metode `cv2.goodFeaturesToTrack()`

Metode ini digunakan untuk mendeteksi sudut-sudut terkuat dalam gambar. Berikut adalah penjelasan parameter yang digunakan:

- a. Gambar Input: Gambar harus dalam format grayscale (hitam-putih). Jika gambar berwarna, konversi terlebih dahulu menggunakan `cv2.cvtColor()`.
- b. Jumlah Sudut (`maxCorners`): Parameter ini menentukan jumlah sudut maksimal yang ingin dideteksi.
- c. Tingkat Kualitas (`qualityLevel`): Nilainya berkisar antara 0 hingga 1. Parameter ini menentukan kualitas minimum sudut yang akan diterima. Semakin kecil nilainya, semakin banyak sudut yang terdeteksi.
- d. Jarak Minimum (`minDistance`): Parameter ini menentukan jarak minimum (dalam piksel) antara dua sudut yang terdeteksi. Jika jarak antara dua sudut lebih kecil dari nilai ini, hanya satu sudut yang akan dipertahankan.
- e. Mask (opsional): Mask adalah gambar biner yang menentukan area di mana sudut akan dideteksi. Jika tidak disediakan, sudut akan dideteksi di seluruh gambar.



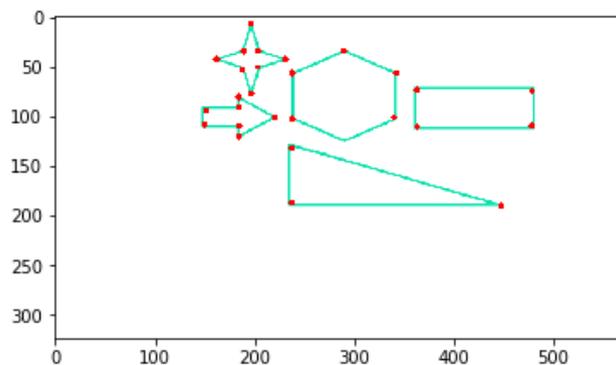
# PRAKTIKUM

## PENGOLAHAN SINYAL INFORMASI

Before :



After :



**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**

### 3.11. Document field detection using Template Matching

Template Matching adalah teknik pengolahan gambar yang digunakan untuk menemukan lokasi bagian kecil (template) dalam gambar yang lebih besar. Teknik ini banyak digunakan dalam proyek deteksi objek, seperti pemeriksaan kualitas produk, pelacakan kendaraan, robotika, dan lainnya. Kita akan mempelajari cara menggunakan template matching untuk mendeteksi bidang-bidang terkait dalam gambar dokumen.

1. Memotong (Crop) Template: Potong bagian-bagian kecil (field) dari gambar dokumen utama dan gunakan sebagai template terpisah.
2. Menentukan Threshold: Tentukan atau sesuaikan threshold untuk setiap bidang yang ingin dideteksi. Threshold ini akan membantu memfilter hasil yang tidak relevan.
3. Menerapkan Template Matching: Gunakan fungsi OpenCV `cv2.matchTemplate()` untuk mencocokkan setiap template dengan gambar dokumen.
4. Menggambar Bounding Box: Gunakan koordinat yang diperoleh dari template matching untuk menggambar kotak pembatas



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

(bounding box) di sekitar bidang yang terdeteksi.

5. **Opsional: Augmentasi Template:** Augmentasi template dan fine-tune threshold untuk meningkatkan akurasi deteksi pada berbagai gambar dokumen.

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**

#### 3.12. **Play a video using OpenCV**

OpenCV (Open Source Computer Vision) adalah pustaka visi komputer yang menyediakan berbagai fungsi untuk melakukan operasi pada gambar atau video. Salah satu operasi yang dapat dilakukan adalah memutar video. Dalam modul ini, kita akan mempelajari cara memutar video menggunakan OpenCV di Python.

Langkah-langkah Memutar Video

1. **Membuat Objek VideoCapture:**

Untuk memutar video, kita perlu membuat objek VideoCapture.

Objek ini dapat menerima dua jenis input:

- a. **Device Index:** Nomor kamera (misalnya, 0 untuk kamera pertama, 1 untuk kamera kedua, dst.).
- b. **Nama File Video:** Path ke file video yang ingin diputar.

2. **Membaca Video Frame per Frame:**

Video dibaca frame per frame menggunakan metode `read()` dari objek VideoCapture.

3. **Menampilkan Video:**

Setiap frame ditampilkan menggunakan fungsi `cv2.imshow()`.

4. **Menghentikan Video:**

Video dapat dihentikan dengan menekan tombol tertentu (misalnya, tombol `q`).

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

#### 3.13. **Create video using multiple images using OpenCV**

OpenCV adalah pustaka populer untuk pengolahan gambar dan video. Dalam modul ini, kita akan mempelajari cara membuat video dari beberapa gambar menggunakan OpenCV. Proses ini melibatkan pembacaan gambar, pengubahan ukuran, dan penggabungan gambar-gambar tersebut menjadi video. Berikut adalah langkah-langkah dan implementasi Python-nya.

1. **Instal Library yang Diperlukan:**
  - a. PIL (Pillow): Untuk membuka dan mengubah ukuran gambar.
  - b. cv2 (OpenCV): Untuk membuat video dari gambar.
2. **Buka dan Ubah Ukuran Gambar:**
  - a. Buka gambar menggunakan PIL dan ubah ukurannya sesuai dengan tinggi dan lebar rata-rata.
3. **Buat Video:**
  - a. Gunakan OpenCV (cv2.VideoWriter) untuk membuat video dari gambar-gambar yang telah diubah ukurannya.
4. **Simpan Video:**
  - a. Video disimpan dalam format .avi atau format lainnya.

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**

#### 3.14. **Extract images from video in Python**

OpenCV adalah pustaka yang kuat untuk pengeditan video, pemindaian gambar, dan pengenalan wajah. Pustaka ini banyak digunakan dalam visi komputer untuk mengekstrak informasi bermakna dari video dan gambar. Dalam modul ini, kita akan membahas cara menyimpan frame dari video sebagai gambar menggunakan OpenCV. Proses ini melibatkan pembacaan video, ekstraksi frame, dan penyimpanan frame sebagai gambar.

Langkah-langkah Menyimpan Gambar dari Video

1. **Impor Library yang Diperlukan:**
  - Library utama yang digunakan adalah cv2 (OpenCV).
2. **Baca Video:**
  - Gunakan cv2.VideoCapture(file\_path) untuk membaca video dari file.
3. **Buat Direktori untuk Menyimpan Gambar:**



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

- Buat folder (misalnya, data) untuk menyimpan gambar yang diekstrak. Jika folder sudah ada, lewati proses pembuatan.
4. Ekstrak dan Simpan Frame:
- Gunakan loop untuk membaca frame per frame dari video. Simpan setiap frame sebagai gambar menggunakan `cv2.imwrite()`.
5. Bersihkan Resource:
- Setelah selesai, lepaskan objek VideoCapture dan tutup semua window OpenCV.

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**

#### **3.15. Displaying the coordinates of the points clicked on the image using Python-OpenCV**

OpenCV memungkinkan kita untuk mengontrol dan mengelola berbagai jenis event mouse, seperti klik kiri, klik kanan, atau pergerakan mouse. Dalam modul ini, kita akan mempelajari cara menampilkan koordinat titik yang diklik pada gambar menggunakan OpenCV. Koordinat ini akan ditampilkan baik di shell (terminal) maupun langsung pada gambar.

Langkah-langkah menampilkan koordinat:

1. Impor Library yang Diperlukan:
  - Library utama yang digunakan adalah `cv2` (OpenCV).
2. Baca dan Tampilkan Gambar:
  - Gunakan `cv2.imread()` untuk membaca gambar dan `cv2.imshow()` untuk menampilkannya.
3. Atur Callback untuk Event Mouse:
  - Gunakan `cv2.setMouseCallback()` untuk mengaitkan fungsi yang akan dipanggil saat event mouse terjadi.
4. Buat Fungsi untuk Menangani Event Mouse:
  - Dalam fungsi ini, periksa apakah event yang terjadi adalah klik kiri (`cv2.EVENT_LBUTTONDOWN`) atau klik kanan (`cv2.EVENT_RBUTTONDOWN`).
  - Tampilkan koordinat titik yang diklik di shell dan pada gambar.



## PRAKTIKUM

### *PENGOLAHAN SINYAL INFORMASI*

5. Tutup Window:
  - Gunakan `cv2.waitKey(0)` untuk menunggu tombol keyboard ditekan dan `cv2.destroyAllWindows()` untuk menutup window.

**Kode contoh akan diberikan saat praktikum. Harap perhatikan!**