**COMPUTER VISION AND IMAGE PROCESSING**

**Percobaan 3: Mengakses Video dan Camera.**



**Oleh:**

**Luthfi Aminulloh 1020181013**

**TEKNIK ELEKTRO**

**PROGRAM PASCASARJANA S2 TERAPAN**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**2019**

# Tujuan Percobaan

1. Mahasiswa dapat menggunakan library OpenCV untuk membuat aplikasi pembacaan- penulisan file dari dan ke video
2. Mahasiswa dapat melakukan pengambilan gambar secara online menggunakan kamera dan menyimpan hasilnya pada sebuah file video.
3. Mahasiswa dapat menambahkan beberapa kreasi sendiri seperti bentuk-bentuk obyek, tulisan, kontrol, dan lain-lain dari video atau kamera.

# Persiapan

1. Praktikum ini dapat dikerjakan dengan pra-syarat bahwa mahasiswa:

* Telah mendapatkan matematika lanjut
* Memiliki konsep dasar sinyal dan sistem
* Memiliki konsep dasar pemrograman dan pengoperasian MS Visual C++

1. Software yang diperlukan:

* Microsoft Visual C++ 2010 express atau lebih tinggi.
* OpenCV Library 2.4.9 atau lebih tinggi.

1. Sarana penunjang praktikum:

* File gambar (.bmp, .jpg)
* File video (.avi)

# Pendahuluan

## Mengambil gambar video

Dalam OpenCV, video dapat dibaca dengan menggunakan umpan dari kamera yang terhubung ke komputer atau dengan membaca file video. Langkah pertama menuju membaca file video adalah membuat objek VideoCapture. Argumennya dapat berupa indeks perangkat atau nama file video yang akan dibaca.

Dalam banyak kasus, hanya satu kamera yang terhubung ke sistem. Jadi, semua yang dapat dilakukan adalah dengan memberikan nilai ‘0’ dan OpenCV menggunakan satu- satunya kamera yang terpasang ke komputer. Ketika lebih dari satu kamera terhubung ke komputer, kita dapat memilih kamera kedua dengan memberikan ‘1’, kamera ketiga dengan meneruskan ‘2’ dan seterusnya. Setelah objek VideoCapture dibuat, kita dapat menangkap frame video demi frame.

|  |
| --- |
|  |
|  |

## Menampilkan Video

Setelah membaca file video, kita bisa menampilkan frame video demi frame. Bingkai video hanyalah gambar dan kita dapat menampilkan setiap bingkai dengan cara yang sama dengan kita menampilkan gambar, yaitu, menggunakan fungsi imshow().

Seperti dalam kasus gambar, kita dapat menggunakan fungsi waitKey() setelah imshow() untuk menjeda setiap frame dalam video. Dalam kasus gambar, kita dapat meneruskan fungsi ‘0’ ke waitKey(), tetapi untuk memutar video, kita perlu memberikan angka yang lebih besar dari ‘0’ ke fungsi waitKey (). Ini karena '0' akan menjeda frame dalam video untuk jumlah waktu yang tak terbatas dan dalam video kita perlu setiap frame untuk ditampilkan hanya untuk beberapa interval waktu yang terbatas, jadi kita harus melewati angka yang lebih besar dari '0' ke fungsi waitKey (). Angka ini sama dengan waktu dalam milidetik yang kita inginkan setiap frame untuk ditampilkan.

Saat membaca frame dari webcam, menggunakan waitKey (1) adalah tepat karena laju frame layar akan dibatasi oleh laju frame webcam bahkan jika kita menetapkan penundaan 1 ms di waitKey. Saat membaca bingkai dari video yang sedang kita proses, mungkin masih tepat untuk mengatur waktu tunda hingga 1 ms sehingga thread tersebut dibebaskan untuk melakukan pemrosesan yang ingin kita lakukan. Dalam kasus yang jarang terjadi, ketika pemutaran perlu pada framerate tertentu, kita mungkin ingin penundaan lebih tinggi dari 1 ms.

## Menulis Video

Setelah selesai dengan menangkap dan memproses frame video demi frame, langkah berikutnya yang kita dapat lakukan adalah untuk menyimpan video. Untuk gambar, itu mudah. Kita hanya perlu menggunakan cv::imwrite(). Tetapi untuk video, kita perlu bekerja keras sedikit lebih keras. Kita perlu membuat objek VideoWriter. Pertama, kita harus menentukan nama file output dengan formatnya (misalnya: output.avi). Kemudian, kita harus menentukan kode FourCC dan jumlah frame per detik (FPS). Terakhir, ukuran frame harus didefinisikan.

*// Define the codec and create VideoWriter object.The output is stored in 'outcpp.avi' file.*

*// Define the fps to be equal to 10. Also frame size is passed.*

*cv::VideoWriter video("out.avi",CV\_FOURCC('M','J','P','G'),10,Size(frame\_width,frame\_height));*

FourCC adalah kode 4-byte yang digunakan untuk menentukan codec video. Daftar kode yang tersedia dapat ditemukan di fourcc.org. Ada banyak kode FOURCC yang tersedia, tetapi dalam posting ini, kami hanya akan bekerja dengan MJPG.

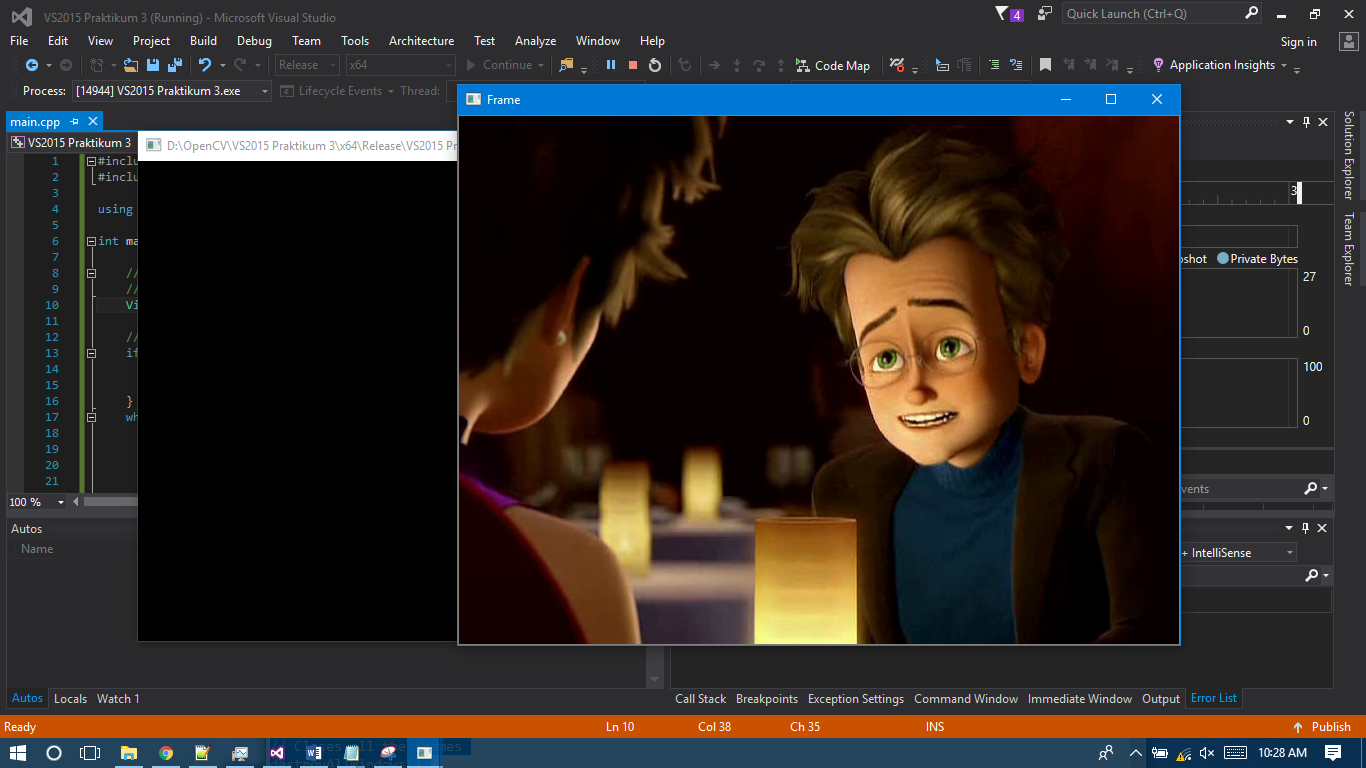
Catatan: Hanya beberapa kode FourCC yang tercantum di atas yang akan berfungsi pada sistem Anda berdasarkan ketersediaan codec di sistem Anda. Kadang-kadang, bahkan ketika codec tertentu tersedia, OpenCV mungkin tidak dapat menggunakannya. MJPG adalah pilihan yang aman.

# Prosedur percobaan

## Membaca Video dari File

|  |
| --- |
| List Program: |
| #include<opencv2\opencv.hpp>  #include <iostream>  using namespace std;  using namespace cv;  int main() {  // Create a VideoCapture object and open the input file  // If the input is the web camera, pass 0 instead of the video file name  VideoCapture cap("../Megamind.avi");  // Check if camera opened successfully  if (!cap.isOpened()) {  cout << "Error opening video stream or file" << endl;  return -1;  }  while (1) {  Mat frame;  // Capture frame-by-frame  cap >> frame;  // If the frame is empty, break immediately  if (frame.empty())  break;  // Display the resulting frame  imshow("Frame", frame);  // Press ESC on keyboard to exit  char c = (char)waitKey(25);  if (c == 27)  break;  }  // When everything done, release the video capture object  cap.release();  // Closes all the frames  destroyAllWindows();  return 0;  } |

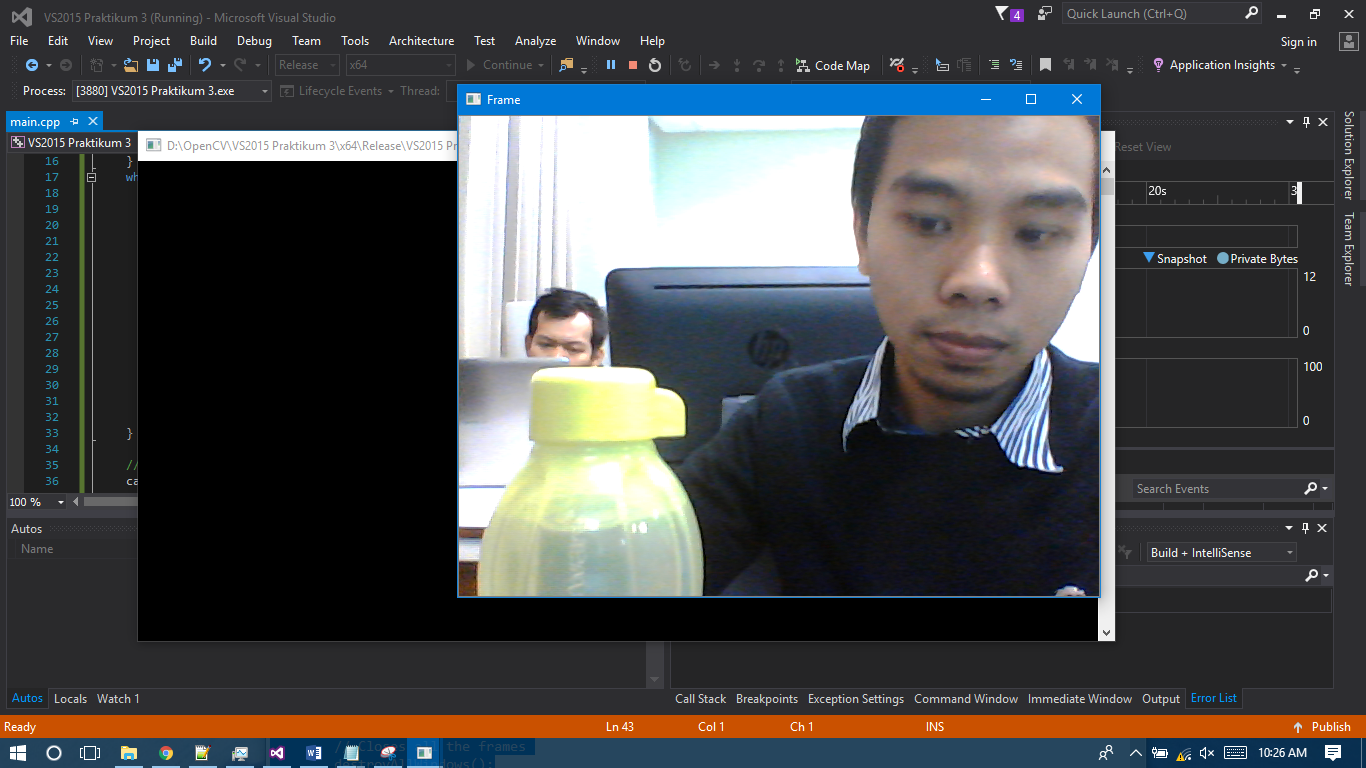
Hasil Percobaan:



## Mengakses kamera secara online

|  |
| --- |
| List Program: |
| #include<iostream>  #include<opencv2\opencv.hpp>  using namespace std;  using namespace cv;  int main() {  // Create a VideoCapture object and use camera to capture the video  VideoCapture cap(0);  // Check if camera opened successfully  if (!cap.isOpened())  {  cout << "Error opening video stream" << endl;  return -1;  }  // Default resolution of the frame is obtained.The default resolution is system dependent.  int frame\_width = cap.get(CV\_CAP\_PROP\_FRAME\_WIDTH);  int frame\_height = cap.get(CV\_CAP\_PROP\_FRAME\_HEIGHT);  // Define the codec and create VideoWriter object.The output is stored in 'outcpp.avi' file.  VideoWriter video("out.avi", CV\_FOURCC('M', 'J', 'P', 'G'), 10, Size(frame\_width, frame\_height));  while (1)  {  Mat frame;  // Capture frame-by-frame  cap >> frame;  // If the frame is empty, break immediately  if (frame.empty())  break;  // Write the frame into the file 'outcpp.avi'  video.write(frame);  // Display the resulting frame  imshow("Frame", frame);  // Press ESC on keyboard to exit  char c = (char)waitKey(1);  if (c == 27)  break;  }  // When everything done, release the video capture and write object  cap.release();  video.release();  // Closes all the windows  destroyAllWindows();  return 0;  } |

Hasil Percobaan:



## Menyimpan hasil capture kamera ke file video

|  |
| --- |
| List Program: |
| #include <iostream>  #include<opencv2\opencv.hpp>  using namespace std;  using namespace cv;  int main() {  // Create a VideoCapture object and use camera to capture the video  VideoCapture cap(0);  // Check if camera opened successfully  if (!cap.isOpened())  {  cout << "Error opening video stream" << endl;  return -1;  }  // Default resolution of the frame is obtained.The default resolution is system dependent.  int frame\_width = cap.get(CV\_CAP\_PROP\_FRAME\_WIDTH);  int frame\_height = cap.get(CV\_CAP\_PROP\_FRAME\_HEIGHT);  // Define the codec and create VideoWriter object.The output is stored in 'outcpp.avi' file.  VideoWriter video("out.avi", CV\_FOURCC('M', 'J', 'P', 'G'), 10, Size(frame\_width, frame\_height));  while (1)  {  Mat frame;  // Capture frame-by-frame  cap >> frame;  // If the frame is empty, break immediately  if (frame.empty())  break;  // Write the frame into the file 'outcpp.avi'  video.write(frame);  // Display the resulting frame  imshow("Frame", frame);  // Press ESC on keyboard to exit  char c = (char)waitKey(1);  if (c == 27)  break;  }  // When everything done, release the video capture and write object  cap.release();  video.release();  // Closes all the windows  destroyAllWindows();  return 0;  } |

Hasil Percobaan:

|  |
| --- |
|  |
|  |

As

|  |
| --- |
|  |
|  |

# Tugas

1. Kombinasikanlah hasil capture dari kamera atau pembacaan file video dengan penambahan beberapa obyek seperti yang telah dibahas pada praktikum 2, misalnya: obyek kotak/lingkaran/elips, tulisan, trackbar, dll. Tuliskan/simpan hasilnya pada sebuah file video!