

# **DASAR DASAR OPENCV**



---

# **DASAR DASAR OPENCV**

## **Cerdas Mengoprasikan Gambar**

---

**Faisal Najib Abdullah**  
Informatics Research Center



**Kreatif Industri Nusantara**



*'Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.'*

*Imam Syafi'i*

# CONTRIBUTORS

---

FAISAL NAJIB ABDULLAH, Informatics Research Center., Politeknik Pos Indonesia,  
Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1 Pengenalan OpenCV</b>	<b>1</b>
<b>2 Dasar Dasar Code OpenCV</b>	<b>41</b>



# DAFTAR ISI

---

Daftar Gambar	xiii
Daftar Tabel	xix
Foreword	xxv
Kata Pengantar	xxvii
Acknowledgments	xxix
Acronyms	xxxi
List of Symbols	xxxiii
Introduction	xxxv
<i>Faisal Najib Abdullah</i>	

<b>1 Pengenalan OpenCV</b>	<b>1</b>
1.1 OpenCV	1
1.1.1 Definisi OpenCV	1
1.1.2 Sejarah Kecerdasan Opencv	4
1.2 NumPy	5
1.2.1 Definisi NumPy	5
1.2.2 Sejarah NumPy	5

1.2.3	Perkembangan NumPy	6
1.2.4	Contoh Code Numpy	8
1.3	Instalasi OpenCV	10
1.3.1	Instalasi pada Windows	10
1.3.2	Instalasi pada OS X	23
1.3.3	Instalasi pada Ubuntu	34
1.3.4	Instalasi modul Contrib	37
1.3.5	PyCharm	38
<b>2</b>	<b>Dasar Dasar Code OpenCV</b>	<b>41</b>
2.1	Menampilkan gambar	41
2.1.1	Menampilkan Gambar	41
2.1.2	Menampilkan Gambar dan merubah kontras warnanya	43
2.1.3	Menyimpan Gambar menggunakan kode opencv	45
2.2	Menjalankan kamera leptop	47
2.2.1	Menjalankan video kamera leptop	47
2.2.2	Merubah kontras warna pada video	49
2.2.3	Mengetahui ukuran frame yang ditampilkan	51
2.2.4	Menyimpan video	53
2.3	Menggambar Geometric Pada Foto	56
2.3.1	Membuat garis	56
2.3.2	Membuat warna warna pada garis	58
2.3.3	Membuat garis panah	61
2.3.4	Membuat garis kotak	63
2.3.5	Membuat kotak	65
2.3.6	Membuat garis Lingkaran	67
2.3.7	Membuat Text	69
2.4	Frame Numpay	71
2.4.1	Membuat Frame menggunakan Numpay	71
2.5	Ukuran Frame Menggunakan CAP PROP FRAME	73
2.5.1	Berubah Ukuran Frame	73
2.6	Video Text	75
2.6.1	Menampilkan Text pada Frame video	75
2.6.2	Menampilkln waktu pada frame	77
2.7	Event Mouse Klik	79
2.7.1	Menampilkan Event	79
2.7.2	Event Mouse klik kiri	80
2.7.3	Event Mouse klik kiri dan kanan	84

2.7.4	Event Mouse klik kiri dan kanan pada gambar yang dipanggil	86
2.7.5	Event Mouse klik kiri membuat titik dan garis	89
2.7.6	Membuat frame warna sesuai klik	93
2.7.7	Membuat frame warna sesuai klik 2	96
2.7.8	Menampilkan Shape, size, dan dtype	98
2.7.9	Mengcopy gambar di dalam satu frame	100
2.8	Menyatukan gambar dalam satu frame	102
2.8.1	Menyatukan 2 gambar dalam 1 frame	102
2.8.2	Menggabungkan 2 gambar dengan kontras transparan	104
2.9	Trackbar	106
2.9.1	Membuat Trackbar	106
2.9.2	Membuat Trackbar dengan fungsi warna	109
2.9.3	Membuat Trackbar ON dan OFF	112
2.9.4	Trackbar menampilkan text kontras	116
2.10	Object Detection dan Object Tracking menggunakan HSV Color Space	121
2.10.1	Membuat frame kosong	121
2.10.2	Hsv color space	123
2.10.3	Hsv color space menggunakan Trackbar	125
2.10.4	Menentukan kontras warna menggunakan Trackbar pada video	130
2.11	Image Thresholding	133
2.11.1	Membuat kontras warna menggunakan Binary	133
2.11.2	Membuat kontras warna menggunakan Binary Inv	135
2.11.3	Membuat kontras warna menggunakan Trunc	137
2.11.4	Membuat kontras warna menggunakan Tozero	139
2.11.5	Membuat kontras warna menggunakan Tozero Inv	141
2.12	Adative Thresholding	143
2.12.1	Mengubah warna gambar menggunakan Binary	143
2.12.2	Menggunakan Adaptive Thresh Mean	145
2.12.3	Menggunakan Adaptive Thresh Gaussian	147
2.13	Menggunakan Matplotlib pada OpenCV	149
2.13.1	Installasi Matplotlib	149
2.13.2	Menampilkan gambar menggunakan matplotlib	152
2.13.3	Mengubah warna yang ditampilkan matplotlib	154
2.13.4	Menghilangkan koordinat x dan y pada tampilan matplotlib	157

2.13.5	Menggabungkan frame menjadi satu frame	159
2.14	Morphologikan Transformations	161
2.14.1	Menggunakan imread grayscale	161
2.14.2	Menggunakan Thresh Binary Inv	163
2.14.3	Menggunakan Dilate	165
2.14.4	Mengubah ukuran karnal	167
2.14.5	Menggunakan Erode	169
2.14.6	Menggunakan MorphologyEx Open	171
2.14.7	Menggunakan MorphologyEx Close	173
2.14.8	Menggunakan MorphologyEx Gradient	175
2.14.9	Menggunakan MorphologyEx Tophat	177
2.15	Smoothing Images	179
2.15.1	Menggunakan Filter 2D	179
2.15.2	Menggunakan Event Blur	181
2.15.3	Menggunakan Event Gaussian Blur	183
2.15.4	Menggunakan Event Median Blur	185
2.15.5	Menggunakan Event BilateralFilter	187
2.16	Image Gradients	189
2.16.1	Menggunakan Event Laplacian	189
2.16.2	Menggunakan Event Laplacian menggunakan Size	191
2.16.3	Menggunakan Event Sobel X	193
2.16.4	Menggunakan Event Sobel Y	195
2.16.5	Menggunakan Event Bitwise	197
2.16.6	Menggunakan Event Canny	199
2.16.7	Menggabungkan Event Canny	201

# DAFTAR GAMBAR

---

1.1	Download Python	11
1.2	Cari Numpy terbaru pada website ini	12
1.3	Cari SciPy Terbaru pada website ini	13
1.4	Cari CMake Terbaru pada website ini kemudian install	15
1.5	Cari VisualStudio Terbaru pada website ini kemudian install	16
1.6	Download MinGW	17
1.7	Download MinGW	18
1.8	Download OpenNI	19
1.9	Download SensorKinect	19
1.10	Download SensorKinect	20
1.11	Download self-extracting	20
1.12	Download Python	23
1.13	Download MacPorts	25

1.14	Download Kinect	28
1.15	Tutorial pada website tersebut	29
1.16	Website packtpub	32
1.17	FreeBSD	36
1.18	Pycharm	38
1.19	Pycharm Settings	39
1.20	Install library	39
1.21	Install OpenCV	40
2.1	Menampilkan gambar	42
2.2	Merubah kontras warna	44
2.3	Menyimpan gambar	46
2.4	Menggunakan kamera leptop	48
2.5	Kontras warna video	50
2.6	Ukuran Frame video	52
2.7	Setelah dijalankan file tersimpan	54
2.8	File Sebelum dijalankan	55
2.9	Video yang sudah di simpan	55
2.10	MMembuat garis	57
2.11	Membuat garis	59
2.12	Nomor Nomor warna	60
2.13	Membuat garis panah	62
2.14	Membuat garis kotak	64
2.15	Membuat kotak	66
2.16	Membuat garis lingkaran	68
2.17	Membuat Text	70
2.18	Membuat Frame Numpy	72
2.19	Merubah ukuran frame	74
2.20	Menampilkan Text pada Frame video	76

2.21	Menampilkan waktu pada frame	78
2.22	Menampilkan Event	79
2.23	Event Mouse klik kiri	81
2.24	Event Mouse klik kiri	82
2.25	Event Mouse klik kiri	83
2.26	Event Mouse klik kiri dan kanan	85
2.27	Event Mouse klik kiri dan kanan pada gambar yang dipanggil	88
2.28	Event Mouse klik kiri membuat titik dan garis	90
2.29	Event Mouse klik kiri membuat titik dan garis	91
2.30	Event Mouse klik kiri membuat titik dan garis	92
2.31	Membuat frame warna sesuai klik	94
2.32	Membuat frame warna sesuai klik	95
2.33	Membuat frame warna sesuai klik	95
2.34	Membuat frame warna sesuai klik 2	97
2.35	Menampilkan Shape, size, dan dtype	99
2.36	Mengcopy gambar di dalam satu frame	101
2.37	Menyatukan 2 gambar dalam 1 frame	103
2.38	Menggabungkan 2 gambar dengan kontras transparan	105
2.39	Membuat Trackbar	107
2.40	Membuat Trackbar	108
2.41	Membuat Trackbar dengan fungsi warna	110
2.42	Membuat Trackbar dengan fungsi warna	111
2.43	Membuat Trackbar ON dan OFF	114
2.44	Membuat Trackbar ON dan OFF	115
2.45	Trackbar menampilkan text kontras	118
2.46	Trackbar menampilkan text kontras	119
2.47	Trackbar menampilkan text kontras	120
2.48	Membuat frame kosong	122

2.49	Hsv color space	124
2.50	Hsv color space menggunakan Trackbar	127
2.51	Hsv color space menggunakan Trackbar	128
2.52	Hsv color space menggunakan Trackbar	128
2.53	Hsv color space menggunakan Trackbar	129
2.54	Menentukan kontras warna menggunakan Trackbar pada video	132
2.55	Membuat kontras warna menggunakan Binary	134
2.56	Membuat kontras warna menggunakan Binary Inv	136
2.57	Membuat kontras warna menggunakan Trunc	138
2.58	Membuat kontras warna menggunakan Tozero	140
2.59	Membuat kontras warna menggunakan Tozero Inv	142
2.60	Mengubah warna gambar menggunakan Binary	144
2.61	Menggunakan Adaptive Thresh Mean	146
2.62	Menggunakan Adaptive Thresh Gaussian	148
2.63	Installasi matplotlib	149
2.64	Installasi matplotlib	150
2.65	Installasi matplotlib	151
2.66	Menampilkan gambar menggunakan matplotlib	153
2.67	Mengubah warna yang ditampilkan matplotlib	155
2.68	Mengubah warna yang ditampilkan matplotlib	156
2.69	Menghilangkan koordinat x dan y pada tampilan matplotlib	158
2.70	Menggabungkan frame menjadi satu frame	160
2.71	Menggunakan imread grayscale	162
2.72	Menggunakan Thresh Binary Inv	164
2.73	Menggunakan Dilate	166
2.74	Mengubah ukuran karnal	168
2.75	Menggunakan Erode	170
2.76	Menggunakan MorphologyEx Open	172

2.77	Menggunakan MorphologyEx Close	174
2.78	Menggunakan MorphologyEx Gradient	176
2.79	Menggunakan MorphologyEx Tophat	178
2.80	Menggunakan Filter 2D	180
2.81	Menggunakan Event Blur	182
2.82	Menggunakan Event Gaussian Blur	184
2.83	Menggunakan Event Median Blur	186
2.84	Menggunakan Event BilateralFilter	188
2.85	Menggunakan Event Laplacian	190
2.86	Menggunakan Event Laplacian menggunakan Size	192
2.87	Menggunakan Event Sobel X	194
2.88	Menggunakan Event Sobel Y	196
2.89	Menggunakan Event Bitwise	198
2.90	Menggunakan Event Canny	200
2.91	Menggabungkan Event Canny	203



## DAFTAR TABEL

---



# Listings

---

src/MembuatArray.py	8
src/OperasiSederhana.py	8
src/UniversalFunction.py	8
src/LinearAlgebra.py	8
src/Penggabungan.py	9
src/cv2.py	41
src/cv1.py	43
src/cv3.py	45
src/cv4.py	47
src/cv5.py	49
src/cv6.py	51
src/cv7.py	53
src/cv8.py	56
src/cv9.py	58
src/cv10.py	61
src/cv11.py	63
src/cv12.py	65
src/cv13.py	67

src/cv14.py	69
src/cv15.py	71
src/cv16.py	73
src/cv17.py	75
src/cv18.py	77
src/cv19.py	79
src/cv20.py	80
src/cv21.py	84
src/cv22.py	86
src/cv23.py	89
src/cv24.py	93
src/cv25.py	96
src/cv26.py	98
src/cv27.py	100
src/cv28.py	102
src/cv29.py	104
src/cv30.py	106
src/cv31.py	109
src/cv32.py	112
src/cv33.py	116
src/cv34.py	121
src/cv35.py	123
src/cv36.py	125
src/cv37.py	130
src/cv38.py	133
src/cv39.py	135
src/cv40.py	137
src/cv41.py	139
src/cv42.py	141
src/cv43.py	143
src/cv44.py	145
src/cv45.py	147
src/cv46.py	152
src/cv47.py	154
src/cv48.py	157
src/cv49.py	159
src/cv50.py	161
src/cv51.py	163

src/cv52.py	165
src/cv53.py	167
src/cv54.py	169
src/cv55.py	171
src/cv56.py	173
src/cv57.py	175
src/cv58.py	177
src/cv59.py	179
src/cv60.py	181
src/cv61.py	183
src/cv62.py	185
src/cv63.py	187
src/cv64.py	189
src/cv65.py	191
src/cv66.py	193
src/cv67.py	195
src/cv68.py	197
src/cv69.py	199
src/cv70.py	201



# **FOREWORD**

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi pengembang yang ingin mengembangkan kemampuan memprogram di bahasa python menggunakan library opencv.

FAISAL NAJIB ABDULLAH

*Bandung, Jawa Barat*

*Januari, 2020*



## ACKNOWLEDGMENTS

---

Terima kasih atas masukan dari pembimbing agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Belajar.

R. M. A.



# ACRONYMS

---

AI	Artificial Intelligence
ETL	Extract Transform Load
NLP	Natural Language Processing



# SYMBOLS

---

$A$  Amplitude

$\&$  Propositional logic symbol

$a$  Filter Coefficient

$B$  Number of Beats



# INTRODUCTION

---

FAISAL NAJIB ABDULLAH

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. Pemrograman merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi pengantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi program pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren.

$$ABC\mathcal{DEF}\alpha\beta\Gamma\Delta \sum_{def}^{abc} \quad (I.1)$$



# BAB 1

---

## PENGENALAN OPENCV

---

### 1.1 OpenCV

#### 1.1.1 Definisi OpenCV

Computer Vision adalah ilmu pemrograman komputer untuk memproses dan pada akhirnya memahami gambar dan video, atau hanya mengatakan membuat komputer melihatnya. Memecahkan sebagian kecil dari tantangan Visi Komputer tertentu, menciptakan kemungkinan baru yang menarik dalam teknologi, teknik, dan bahkan hiburan. Untuk memajukan penelitian visi dan menyebarkan pengetahuan visi, sangat penting untuk memiliki perpustakaan fungsi pemrograman dengan kode yang dioptimalkan dan portabel, dan mudah-mudahan tersedia secara gratis. Ini adalah tujuan asli tim Intel pada tahun 1999 ketika OpenCV (Open Source Computer Vision Library) secara resmi diluncurkan. Sejak itu, sejumlah programmer telah berkontribusi pada perkembangan perpustakaan terbaru. Perubahan besar terjadi pada tahun 2009 (OpenCV 2) yang mencakup perubahan utama pada antarmuka C++. Rilis perpustakaan terbaru dapat ditemukan di situs web resmi OpenCV. Saat ini perpustakaan memiliki 2500 algoritma yang dioptimalkan. Ini digunakan secara luas di seluruh dunia, memiliki 2,5 juta unduhan dan 40 ribu orang di grup peng-

guna. OpenCV dapat digunakan dalam aplikasi akademik dan komersial juga, di bawah lisensi BSD. Untuk menguasai setiap elemen pustaka OpenCV perlu berdiskusi dengan banyak buku yang tersedia tentang topik OpenCV. Namun demikian, membaca materi yang lebih komprehensif seperti itu harus lebih mudah setelah memahami ide dasar tentang OpenCV dari makalah ini. Bahkan untuk membuatnya lebih nyaman, teks yang disajikan di sini dengan cermat mengikuti salah satu sumber OpenCV terbaru.

OpenCV adalah pustaka Pengolah Gambar yang dibuat oleh Intel, dapat diunduh secara bebas dan tersedia untuk C, C++, Java dan Python, Windows, Linux, Mac OS, iOS dan Android Terbaru Versi-versinya adalah opencv2.4.13 dan opencv-3.1. Ini adalah Sumber terbuka dan juga Mudah digunakan dan diinstal. Ini dirancang untuk efisiensi komputasi fokus yang kuat pada aplikasi waktu nyata Implementasi pertama adalah dalam bahasa pemrograman C; Namun, popularitasnya tumbuh dengan implementasi C++ pada Versi 2.0. Fungsi baru diprogram dengan C++. Namun, saat ini, perpustakaan memiliki antarmuka penuh untuk bahasa pemrograman lain, seperti Java, Python, dan MATLAB / Octave. OpenCV tersedia secara bebas untuk diunduh di <http://opencv.org>. Situs ini menyediakan versi terakhir untuk distribusi (saat ini, 3.0 beta) dan versi yang lebih lama. OpenCV mengharuskan gambar dalam BGR atau Grayscale untuk ditampilkan atau disimpan. Kalau tidak, efek yang tidak diinginkan dapat terjadi.

Pustaka OpenCV (sejak versi 2.2) dibagi menjadi beberapa modul, di mana setiap modul dapat dipahami, secara umum, sebagai didedikasikan untuk satu kelompok masalah penglihatan komputer. Semua kelas dan fungsi didefinisikan dalam ruang nama cv. Oleh karena itu untuk mengaksesnya kita dapat mendahului definisi fungsi utama dengan deklarasi menggunakan namespace cv; atau awali nama kelas dan fungsi OpenCV berdasarkan spesifikasi namespace cv. Objek utama adalah Mat kelas. Seperti dilibatkan oleh nama kelas itu pada dasarnya adalah sebuah matriks yang memegang nilai-nilai piksel dari beberapa gambar dan, di samping itu, sejumlah atribut tentang suatu gambar. Dalam kasus yang paling sederhana, sebuah gambar dapat dibuat sebagai cv :: Mat image ;, membuat gambar dengan ukuran 0 x 0. Mungkin variabel anggota yang paling penting dari objek gambar adalah data di mana anggota image.data sebenarnya adalah penunjuk ke memori yang dialokasikan blok yang berisi data gambar (dalam kasus sepele ini adalah image.data = 0). Atau selama pembuatan objek Mat kita bisa secara eksplisit menentukan ukuran awal dan jenis setiap elemen matriks. Jenis ini menentukan, misalnya, menandatangi nilai gambar piksel 1-byte, atau tiga saluran untuk gambar berwarna , atau bahkan angka titik apung 32-bit atau 64bit.

Setelah objek matematika kelas didefinisikan, fitur yang bagus tentangnya (tidak ada dalam versi awal OpenCV) adalah bahwa alokasi / deallokasi memori dilakukan secara otomatis. Misalnya, memori yang secara otomatis dialokasikan selama gambar dibacakan ke beberapa objek, juga akan secara otomatis dirilis setelah objek yang sesuai keluar dari ruang lingkup. Hal penting lainnya adalah bahwa kelas Mat mengimplementasikan penghitungan referensi dan salinan dangkal. Oleh karena itu, ketika gambar ditugaskan ke yang lain, data gambar itu sendiri tidak disalin dan kedua gambar menunjuk ke blok memori yang sama (ini juga berlaku untuk gambar

yang dilewati / dikembalikan oleh nilai). Namun, karena jumlah referensi didukung, memori yang dialokasikan untuk data gambar (piksel) itu sendiri akan dirilis hanya ketika semua referensi ke gambar dihancurkan.

Dalam versi sebelum OpenCV 2, fungsi dan struktur seperti C digunakan (masih bisa jadi) dan struktur utama adalah IplImage. Meskipun ada cara mudah untuk mengubah struktur IplImage menjadi objek cv :: Mat, sangat disarankan untuk menghindari struktur data yang sudah usang ini.

### 1.1.2 Sejarah Kecerdasan Opencv

Resmi diluncurkan pada tahun 1999, proyek OpenCV pada awalnya merupakan inisiatif Intel Research untuk memajukan aplikasi intensif CPU, bagian dari serangkaian proyek termasuk penelusuran sinar waktu nyata dan dinding layar 3D. Kontributor utama untuk proyek ini termasuk sejumlah pakar optimisasi di Intel Rusia, serta Tim Perpustakaan Kinerja Intel. Pada hari-hari awal OpenCV, tujuan proyek digambaran sebagai:

1. Memajukan penelitian visi dengan menyediakan tidak hanya kode terbuka tetapi juga dioptimalkan untuk infrastruktur visi dasar. Tidak ada lagi menciptakan kembali roda.
2. Menyebarluaskan pengetahuan visi dengan menyediakan infrastruktur umum yang dapat dibangun oleh pengembang, sehingga kode akan lebih mudah dibaca dan dapat ditransfer.
3. Aplikasi komersial berbasis visi mutakhir dengan membuat kode portabel yang dioptimalkan kinerja tersedia secara gratis - dengan lisensi yang tidak memerlukan kode terbuka atau bebas sendiri.

Versi alpha pertama dari OpenCV dirilis ke publik di Konferensi IEEE pada Computer Vision dan Pattern Recognition pada tahun 2000, dan lima beta dirilis antara tahun 2001 dan 2005. Versi 1.0 pertama dirilis pada tahun 2006. Versi 1.1 "prerilis" dirilis pada Oktober 2008.

Rilis utama kedua dari OpenCV adalah pada Oktober 2009. OpenCV 2 mencakup perubahan besar pada antarmuka C++, yang bertujuan untuk lebih mudah, pola yang lebih aman, fungsi baru, dan implementasi yang lebih baik untuk yang sudah ada dalam hal kinerja (terutama pada multi-sistem inti). Rilis resmi sekarang terjadi setiap enam bulan dan pengembangan sekarang dilakukan oleh tim Rusia independen yang didukung oleh perusahaan komersial.

Pada Agustus 2012, dukungan untuk OpenCV diambil alih oleh yayasan nirlaba OpenCV.org, yang mengelola pengembangan dan situs pengguna.

Pada Mei 2016, Intel menandatangani perjanjian untuk mengakuisisi Itseez, pengembang OpenCV terkemuka.

## 1.2 NumPy

### 1.2.1 Definisi NumPy

NumPy adalah perpustakaan untuk bahasa pemrograman Python, menambahkan dukungan untuk array dan matriks multi-dimensi yang besar, bersama dengan koleksi besar fungsi matematika tingkat tinggi untuk beroperasi pada array ini. Nenek moyang NumPy, Numeric, pada awalnya diciptakan oleh Jim Hugunin dengan kontribusi dari beberapa pengembang lainnya. Pada 2005, Travis Oliphant menciptakan NumPy dengan memasukkan fitur-fitur Numarray yang bersaing ke Numeric, dengan modifikasi ekstensif. NumPy adalah perangkat lunak sumber terbuka dan memiliki banyak kontributor.

### 1.2.2 Sejarah NumPy

Bahasa pemrograman Python pada awalnya tidak dirancang untuk komputasi numerik, tetapi menarik perhatian komunitas ilmiah dan teknik sejak awal, sehingga kelompok minat khusus yang disebut matrix-sig didirikan pada tahun 1995 dengan tujuan mendefinisikan paket komputasi array. Di antara anggotanya adalah desainer dan pengelola Python, Guido van Rossum, yang menerapkan ekstensi ke sintaksis Python (khususnya sintaks pengindeksan) untuk membuat komputasi array lebih mudah.

Implementasi dari paket matriks diselesaikan oleh Jim Fulton, kemudian digeneralisasi oleh Jim Hugunin menjadi Numeric, juga beragam disebut ekstensi Numerical Python atau NumPy. Hugunin, seorang mahasiswa pascasarjana di Massachusetts Institute of Technology (MIT), bergabung dengan Corporation for National Research Initiatives (CNRI) untuk bekerja pada JPython pada tahun 1997 meninggalkan Paul Dubois dari Lawrence Livermore National Laboratory (LLNL) ke ambil alih sebagai pengelola. Kontributor awal lainnya termasuk David Ascher, Konrad Hinsen dan Travis Oliphant.

Paket baru bernama Numarray ditulis sebagai pengganti Numeric yang lebih fleksibel. Seperti Numeric, sekarang sudah usang. Numarray memiliki operasi lebih cepat untuk array besar, tetapi lebih lambat daripada Numeric pada array kecil, jadi untuk sementara waktu kedua paket digunakan untuk kasus penggunaan yang berbeda. Versi terakhir Numeric v24.2 dirilis pada 11 November 2005 dan numarray v1.5.2 dirilis pada 24 Agustus 2006.

Ada keinginan untuk memasukkan Numeric ke dalam pustaka standar Python, tetapi Guido van Rossum memutuskan bahwa kodennya tidak dapat dipertahankan dalam keadaannya saat itu.

Pada awal 2005, pengembang NumPy Travis Oliphant ingin menyatukan komunitas sekitar satu paket array dan mem-porting fitur Numarray ke Numeric, merilis hasilnya sebagai NumPy 1.0 pada 2006. Proyek baru ini adalah bagian dari SciPy. Untuk menghindari menginstal paket SciPy besar hanya untuk mendapatkan objek array, paket baru ini dipisahkan dan disebut NumPy. Dukungan untuk Python 3 ditambahkan pada 2011 dengan NumPy versi 1.5.0.

Pada tahun 2011, PyPy memulai pengembangan penerapan API NumPy untuk PyPy. Ini belum sepenuhnya kompatibel dengan NumPy.

### 1.2.3 Perkembangan NumPy

NumPy menargetkan implementasi referensi CPython dari Python, yang merupakan penerjemah bytecode yang tidak mengoptimalkan. Algoritma matematika yang ditulis untuk versi Python ini sering berjalan jauh lebih lambat daripada yang dikompilasi setara. NumPy mengatasi masalah kelambatan sebagian dengan menyediakan array multidimensi dan fungsi dan operator yang beroperasi secara efisien pada array, membutuhkan penulisan ulang beberapa kode, sebagian besar loop internal menggunakan NumPy.

Menggunakan NumPy dalam Python memberikan fungsionalitas yang sebanding dengan MATLAB karena keduanya diinterpretasikan, dan keduanya memungkinkan pengguna untuk menulis program cepat selama sebagian besar operasi bekerja pada array atau matriks, bukan skalar. Sebagai perbandingan, MATLAB menawarkan sejumlah besar kotak alat tambahan, terutama Simulink, sedangkan NumPy secara intrinsik terintegrasi dengan Python, bahasa pemrograman yang lebih modern dan lengkap. Selain itu, paket Python komplementer tersedia; SciPy adalah pustaka yang menambahkan lebih banyak fungsi seperti MATLAB dan Matplotlib adalah paket merencanakan yang menyediakan fungsi merencanakan seperti MATLAB. Secara internal, baik MATLAB dan NumPy mengandalkan BLAS dan LAPACK untuk perhitungan aljabar linier yang efisien.

Binding Python dari perpustakaan visi komputer yang banyak digunakan OpenCV memanfaatkan array NumPy untuk menyimpan dan beroperasi pada data. Karena gambar dengan banyak saluran hanya direpresentasikan sebagai array tiga dimensi, pengindeksan, pemotongan atau penutupan dengan array lainnya adalah cara yang sangat efisien untuk mengakses piksel tertentu dari suatu gambar. Array NumPy sebagai struktur data universal dalam OpenCV untuk gambar, poin fitur yang diekstraksi, filter kernel dan banyak lagi yang sangat menyederhanakan alur kerja pemrograman dan debugging.

#### Struktur data ndarray

Fungsionalitas inti dari NumPy adalah ndarray, untuk susunan n-dimensional, struktur data. Array-array ini adalah pandangan yang melintas pada memori. Berbeda dengan struktur data daftar built-in Python (yang, meskipun namanya, adalah array dinamis), array ini diketik secara homogen: semua elemen dari array tunggal harus dari tipe yang sama.

Array tersebut juga dapat dilihat ke buffer memori yang dialokasikan oleh ekstensi C atau C plus plus, Cython, dan Fortran ke juru bahasa CPython tanpa perlu menyalin data, memberikan tingkat kompatibilitas dengan perpustakaan numerik yang ada. Fungsi ini dieksplorasi oleh paket SciPy, yang membungkus sejumlah perpustakaan seperti itu (terutama BLAS dan LAPACK). NumPy memiliki dukungan bawaan untuk ndarrays yang dipetakan di memori.

#### Keterbatasan

Memasukkan atau menambahkan entri ke sebuah array tidak sepele mungkin dengan

daftar Python. Rutin np.pad untuk memperluas array sebenarnya membuat array baru dengan nilai bentuk dan padding yang diinginkan, menyalin array yang diberikan ke yang baru dan mengembalikannya. Operasi NumPy np.concatenate tidak benar-benar menghubungkan dua array tetapi mengembalikan yang baru, diisi dengan entri dari kedua array yang diberikan secara berurutan. Membentuk kembali dimensi array dengan np.reshape hanya mungkin selama jumlah elemen dalam array tidak berubah. Keadaan ini berasal dari fakta bahwa array NumPy harus dilihat pada buffer memori yang berdekatan. Paket pengganti yang disebut Blaze berupaya mengatasi batasan ini.

Algoritma yang tidak dapat diekspresikan sebagai operasi vektor biasanya akan berjalan lambat karena mereka harus diimplementasikan dalam Python murni, sedangkan vektorisasi dapat meningkatkan kompleksitas memori dari beberapa operasi dari konstan ke linier, karena array sementara harus dibuat yang sama besarnya dengan input. Kompilasi kode numerik Runtime telah diterapkan oleh beberapa kelompok untuk menghindari masalah ini; solusi open source yang beroperasi dengan NumPy termasuk scipy.weave, numexpr dan Numba. Cython dan Pythran adalah alternatif kompilasi statis untuk ini.

## 1.2.4 Contoh Code Numpy

Membuat Array menggunakan Numpy Python

```

1 import numpy as np
2 x = np.array([1, 2, 3])
3 x
4 # Hasil dari x
5 array([1, 2, 3])
6
7 y = np.arange(10) # seperti rentang Python, tetapi mengembalikan
8      array
9 y
10 # Hasil dari y
11 array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

```

Operasi Sederhana Numpay

```

1 a = np.array([1, 2, 3, 6])
2 b = np.linspace(0, 2, 4) # buat array dengan empat titik yang
   berjarak sama mulai dari 0 dan berakhir dengan 2.
3 c = a - b
4 c
5 # Hasil dari c
6 array([-1.0, -0.5, 0.0, 1.5])
7
8 a**2
9 # Hasil dari a**2
10 array([1, 4, 9, 36])

```

Fungsi Universal Numpay

```

1 a = np.linspace(-np.pi, np.pi, 100)
2 b = np.sin(a)
3 c = np.cos(a)

```

Aljabar Linear bagaimana memuat matriks menggunakan code numpay dengan mudah.

```

1 from numpy.random import rand
2 from numpy.linalg import solve, inv
3 a = np.array([[1, 2, 3], [3, 4, 6.7], [5, 9.0, 5]])
4 a.transpose()
5 # Hasil
6 array([[ 1. ,  3. ,  5. ],
7        [ 2. ,  4. ,  9. ],
8        [ 3. ,  6.7,  5. ]])
9
10 inv(a)
11 # Hasil
12 array([[-2.27683616,  0.96045198,  0.07909605],
13        [ 1.04519774, -0.56497175,  0.1299435 ],
14        [ 0.39548023,  0.05649718, -0.11299435]])
15
16 b = np.array([3, 2, 1])
17 solve(a, b) # pecahkan persamaan ax = b
18 # Hasil
19 array([-4.83050847,  2.13559322,  1.18644068])

```

```
20
21 c = rand(3, 3) * 20 # buat matriks 3x3 acak dalam [0,1] diskalakan
22 oleh 20
23
24 # Hasil
25 array ([[ 3.98732789, 2.47702609, 4.71167924],
26 [ 9.24410671, 5.5240412 , 10.6468792 ],
27 [ 10.38136661, 8.44968437, 15.17639591]])
28
29 np.dot(a, c) # perkalian matriks
30 # Hasil
31 array ([[ 53.61964114, 38.8741616 , 71.53462537],
32 [ 118.4935668 , 86.14012835, 158.40440712],
33 [ 155.04043289, 104.3499231 , 195.26228855]])
34
35 a @ c # Dimulai dengan Python 3.5 dan NumPy 1.10
36 # Hasil
37 array ([[ 53.61964114, 38.8741616 , 71.53462537],
38 [ 118.4935668 , 86.14012835, 158.40440712],
39 [ 155.04043289, 104.3499231 , 195.26228855]])
```

Contoh penggabungan Numpay dengan Opencv.

```
1 import numpy as np
2 import cv2
3 r = np.reshape(np.arange(256*256)%256,(256,256)) # 256x256 piksel
4         array dengan gradien horizontal dari 0 hingga 255 untuk saluran
5         warna merah
6 g = np.zeros_like(r) # array dengan ukuran dan tipe yang sama dengan
7         r tetapi diisi dengan 0s untuk saluran warna hijau
8 b = r.T # Transposisi r akan memberikan gradien vertikal untuk
9         saluran warna biru
10 cv2.imwrite('gradients.png', np.dstack([b,g,r])) # Gambar OpenCV
11         ditafsirkan sebagai BGR, array kedalaman-ditumpuk akan ditulis ke
12         file PNG 8bit RGB yang disebut 'gradients.png'
13 # Hasil
14 True
```

## 1.3 Instalasi OpenCV

### 1.3.1 Instalasi pada Windows

Windows tidak datang dengan Python yang sudah diinstal. Namun, wizard instalasi tersedia untuk Python, NumPy, SciPy, dan OpenCV yang telah dikompilasi. Atau, kita dapat membangun dari sumber. Sistem build OpenCV menggunakan CMake untuk konfigurasi dan Visual Studio atau MinGW untuk kompilasi.

Jika kami ingin dukungan untuk kamera kedalaman, termasuk Kinect, pertama-tama kita harus menginstal OpenNI dan SensorKinect, yang tersedia sebagai binari yang dikompilasi dengan wizard penginstalan. Kemudian, kita harus membangun OpenCV dari sumber.

#### Catatan

Versi OpenCV yang dikompilasi sebelumnya tidak menawarkan dukungan untuk kamera yang dalam. Pada Windows, OpenCV 2 menawarkan dukungan yang lebih baik untuk Python 32-bit daripada Python 64-bit. namun, dengan mayoritas komputer yang dijual hari ini menggunakan sistem 64-bit, instruksi kami akan merujuk pada 64-bit. Semua installer memiliki versi 32-bit yang tersedia dari situs yang sama dengan 64-bit.

1. Beberapa langkah berikut merujuk pada pengeditan variabel PATH sistem. Tugas ini dapat dilakukan di jendela Variabel Lingkungan di Panel Kontrol.
2. Pada Windows Vista / Windows 7 / Windows 8, klik menu Start dan luncurkan Panel Kontrol. Sekarang, navigasikan ke System and Security, Sistem, Pengaturan sistem lanjutan. Klik pada tombol Variabel Lingkungan....
3. Pada Windows XP, klik pada menu Mulai dan navigasikan ke Control Panel — Sistem. Pilih tab Advanced. Klik pada tombol Variabel Lingkungan....
4. Sekarang, di bawah System variable, pilih Path dan klik tombol Edit....
5. Buat perubahan sesuai petunjuk.
6. Untuk menerapkan perubahan, klik pada semua tombol OK (sampai kita kembali ke jendela utama Control Panel).
7. Kemudian, keluar dan masuk kembali (atau reboot).

## Menggunakan penginstal biner (tidak ada dukungan untuk kamera kedalam-an)

Anda dapat memilih untuk menginstal Python dan pustaka terkait secara terpisah jika Anda mau; Namun, ada distribusi Python yang datang dengan installer yang akan mengatur seluruh tumpukan SciPy (yang mencakup Python dan NumPy), yang membuatnya sangat sepele untuk mengatur lingkungan pengembangan.

Salah satu distribusi tersebut adalah Anaconda Python (dapat diunduh di ). Setelah installer diunduh, jalankan dan ingat untuk menambahkan path ke instalasi Anaconda ke variabel PATH Anda mengikuti prosedur sebelumnya.

Berikut adalah langkah-langkah untuk mengatur Python7, NumPy, SciPy, dan OpenCV:

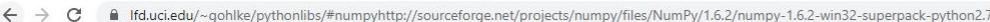
1. Unduh dan instal 64 bit Python 3.7.1 dari

<https://www.python.org/ftp/python/3.7.1/python-3.7.1-amd64.exe>

.. /		
<a href="#">amd64/</a>		
<a href="#">amd64rc1/</a>	20-Oct-2018 15:28	-
<a href="#">amd64rc2/</a>	26-Sep-2018 22:44	-
<a href="#">win32/</a>	13-Oct-2018 16:41	-
<a href="#">win32rc1/</a>	20-Oct-2018 15:28	-
<a href="#">win32rc2/</a>	26-Sep-2018 22:45	-
<a href="#">Python-3.7.1.tar.xz</a>	13-Oct-2018 16:42	-
<a href="#">Python-3.7.1.tar.xz.asc</a>	20-Oct-2018 06:25	16960060
<a href="#">Python-3.7.1.tgz</a>	20-Oct-2018 06:26	833
<a href="#">Python-3.7.1.tgz.asc</a>	20-Oct-2018 06:25	22802018
<a href="#">Python-3.7.1.lrc1.tar.xz</a>	20-Oct-2018 06:26	833
<a href="#">Python-3.7.1.lrc1.tar.xz.asc</a>	26-Sep-2018 20:17	16968876
<a href="#">Python-3.7.1.lrc1.tar.xz.asc</a>	26-Sep-2018 20:17	833
<a href="#">Python-3.7.1.lrc1.tgz</a>	26-Sep-2018 20:16	22800318
<a href="#">Python-3.7.1.lrc1.tgz.asc</a>	26-Sep-2018 20:17	833
<a href="#">Python-3.7.1.lrc2.tar.xz</a>	13-Oct-2018 07:24	16974832
<a href="#">Python-3.7.1.lrc2.tar.xz.asc</a>	13-Oct-2018 07:24	833
<a href="#">Python-3.7.1.lrc2.tgz</a>	13-Oct-2018 07:23	22805187
<a href="#">Python-3.7.1.lrc2.tgz.asc</a>	13-Oct-2018 07:24	833
<a href="#">python-3.7.1-amd64-webinstall.exe</a>	20-Oct-2018 15:27	1333960
<a href="#">python-3.7.1-amd64-webinstall.exe.asc</a>	20-Oct-2018 15:27	836
<a href="#">python-3.7.1-amd64.exe</a>	20-Oct-2018 15:27	26260496
<a href="#">python-3.7.1-amd64.exe.asc</a>	20-Oct-2018 15:27	836
<a href="#">python-3.7.1-embed-amd64.zip</a>	20-Oct-2018 15:27	6879900
<a href="#">python-3.7.1-embed-amd64.zip.asc</a>	20-Oct-2018 15:27	836
<a href="#">python-3.7.1-embed-win32.zip</a>	20-Oct-2018 15:28	6377805
<a href="#">python-3.7.1-embed-win32.zip.asc</a>	20-Oct-2018 15:28	836
<a href="#">python-3.7.1-macosx10.6.pkg</a>	20-Oct-2018 07:55	34360623
<a href="#">python-3.7.1-macosx10.6.pkg.asc</a>	20-Oct-2018 07:55	833
<a href="#">python-3.7.1-macosx10.9.pkg</a>	20-Oct-2018 07:17	27725111

Gambar 1.1 Download Python

## 2. Unduh dan instal NumPy 1.6.2.

 lfd.uci.edu/~gohlke/pythonlibs/#numpy

# Unofficial Windows Binaries for Python Extension Packages

by Christoph Gohlke, Laboratory for Fluorescence Dynamics, University of California, Irvine.

Updated on 23 January 2020 at 07:40 UTC.

This page provides 32- and 64-bit Windows binaries of many scientific open-source extension packages for the official [CPython distribution](#) of the [Python](#) available for the [PyPy](#) distribution.

The files are unofficial (meaning: informal, unrecognized, personal, unsupported, no warranty, no liability, provided "as is") and made available for testing. Most binaries are built from source code found on [PyPI](#) or in the projects public revision control systems. Source code changes, if any, have been submitted by packages.

Refer to the documentation of the individual packages for license restrictions and dependencies.

If downloads fail, reload this page, enable JavaScript, disable download managers, disable proxies, clear cache, use Firefox, reduce number and frequency manually as needed.

Use [pip](#) version 19.2 or newer to [install the downloaded .whl files](#). This page is not a pip package index.

Many binaries depend on [numpy-1.16+mk1](#) and the current [Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019](#) for Python 3, or [Package x64](#), [x86](#), and [SP1](#) for Python 2.7.

Install [numpy+mk1](#) before other packages that depend on it.

The binaries are compatible with the most recent official CPython distributions on Windows >=6.0. Chances are they do not work with custom Python distributions OSGeo4W, ABAQUS, Cygwin, Pythonxy, Canopy, EPD, Anaconda, WinPython etc. Many binaries are not compatible with Windows XP or Wine.

The packages are ZIP or 7z files, which allows for manual or scripted installation or repackaging of the content.

**The files are provided "as is" without warranty or support of any kind. The entire risk as to the quality and performance is with you.**

The opinions or statements expressed on this page should not be taken as a position or endorsement of the Laboratory for Fluorescence Dynamics or the University of California, Irvine.

---

**Index by date:** spectrum jupyter statsmodels cupy sqlalchemy ruamel.yaml scipy cython numpy pillow-simd pymatgen glfw oiffile oiflib  
akima vdsr nsf uciwebauth transformations sdffile netcdf4 molmass lfdfiles imres fmkr fesfiles chebvfit cmanfile dnacurve or

**Gambar 1.2** Cari Numpy terbaru pada wbsite ini

3. Unduh dan pasang SciPy 11.0 dari

```
http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipyhttp://sour
```

(ini sama dengan NumPy dan ini adalah pemasang komunitas).

## Unofficial Windows Binaries for Python Extension Packages

by **Christoph Gohlke, Laboratory for Fluorescence Dynamics, University of California, Irvine.**

Updated on 23 January 2020 at 07:40 UTC.

This page provides 32- and 64-bit Windows binaries of many scientific open-source extension packages for the official [CPython distribution](#) of the [Python](#) available for the [PyPy](#) distribution.

The files are unofficial (meaning: informal, unrecognized, personal, unsupported, no warranty, no liability, provided "as is") and made available for testing.

Most binaries are built from source code found on [PyPI](#) or in the projects public revision control systems. Source code changes, if any, have been submitted by packages.

Refer to the documentation of the individual packages for license restrictions and dependencies.

If downloads fail, reload this page, enable JavaScript, disable download managers, disable proxies, clear cache, use Firefox, reduce number and frequency manually as needed.

Use pip version 19.2 or newer to [install the downloaded whl files](#). This page is not a pip package index.

Many binaries depend on [numpy-1.16+mkl](#) and the current [Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019](#) for Python 3, or Package [x64](#), [x86](#), and [SP1](#) for Python 2.7.

Install [numpy+mkl](#) before other packages that depend on it.

The binaries are compatible with the most recent official CPython distributions on Windows >=6.0. Chances are they do not work with custom Python distributions OSGeo4W, ABAQUS, Cygwin, Pythonxy, Canopy, EPD, Anaconda, WinPython etc. Many binaries are not compatible with Windows XP or Wine.

The packages are ZIP or 7z files, which allows for manual or scripted installation or repackaging of the content.

**The files are provided "as is" without warranty or support of any kind. The entire risk as to the quality and performance is with you.**

The opinions or statements expressed on this page should not be taken as a position or endorsement of the Laboratory for Fluorescence Dynamics or the University of California, Irvine.

---

**Index by date:** spectrum jupyter statsmodels cupy sqlalchemy ruamel.yaml scipy cython numpy pillow-simd pymatgen glfw oiffile orjson akima vidsrc nsf uciwebauth transformations sdffile netmiffile molmass lsfdfiles imres fmkr fcsfiles chehvfit cmanfile dnacurve orfs

### Gambar 1.3 Cari SciPy Terbaru pada website ini

4. Unduh ZIP self-extracting dari OpenCV 3.0.0 dari

```
https://github.com/Itseez/opencv
```

. Jalankan ZIP ini, dan ketika diminta, masukkan folder tujuan.

5. Salin

```
<unzip_destination>\opencv\build\python\2.7\cv2.pyd
```

ke

```
C:\Python2.7\Lib\site-packages
```

(dengan asumsi bahwa kami telah menginstal Python 2.7 ke lokasi default). Jika Anda menginstal Python 2.7 dengan Anaconda, gunakan folder instalasi

Anaconda daripada instalasi Python default. Sekarang, instalasi Python baru dapat menemukan OpenCV.

6. Langkah terakhir diperlukan jika kita ingin skrip Python dijalankan menggunakan instalasi Python baru secara default. Edit variabel PATH sistem dan tambahkan C:/Python2.7 (dengan asumsi kami telah menginstal Python 2.7 ke lokasi default) atau folder instalasi Anaconda Anda. Hapus jalur Python sebelumnya, seperti C:/Python2.6. Logout dan login kembali (sebagai alternatif, reboot).

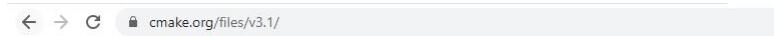
## Menggunakan CMake dan kompiler

Windows tidak dilengkapi dengan kompiler atau CMake. Kita perlu menginstalnya. Jika kami ingin dukungan untuk kamera kedalaman, termasuk Kinect, kami juga perlu menginstal OpenNI dan SensorKinect.

Mari kita asumsikan bahwa kita telah menginstal Python 2.7, NumPy, dan SciPy 32-bit baik dari binari (seperti dijelaskan sebelumnya) atau dari sumber. Sekarang, kita dapat melanjutkan dengan menginstal kompiler dan CMake, menginstal op-sional OpenNI dan SensorKinect, dan kemudian membangun OpenCV dari sumber:

1. Unduh dan instal CMake 3.1.2 dari

<http://www.cmake.org/files/v3.1/cmake3.1.2-win32-x86.exe>



### Index of /files/v3.1

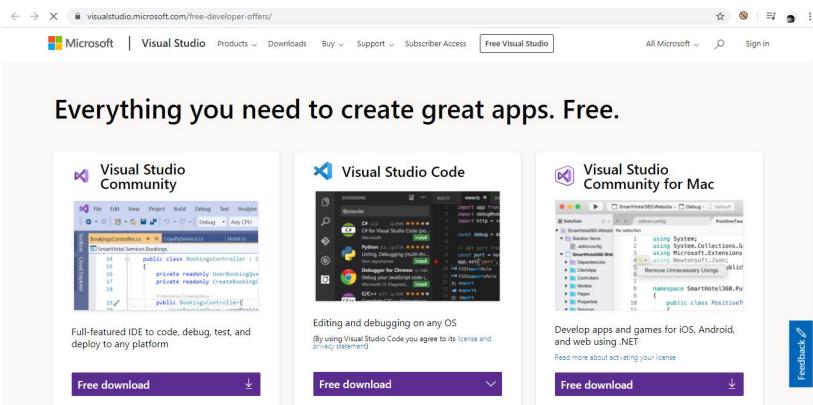
Name	Last modified	Size	Description
Parent Directory		-	
<a href="#">cmake-3.1.0-1-src.tar.bz2</a>	2014-12-17 13:10	4.5M	
<a href="#">cmake-3.1.0-1.tar.bz2</a>	2014-12-17 13:10	9.0M	
<a href="#">cmake-3.1.0-Darwin-universal.dmg</a>	2014-12-17 13:10	45M	
<a href="#">cmake-3.1.0-Darwin-universal.tar.Z</a>	2014-12-17 13:10	63M	
<a href="#">cmake-3.1.0-Darwin-universal.tar.gz</a>	2014-12-17 13:10	44M	
<a href="#">cmake-3.1.0-Darwin64-universal.dmg</a>	2014-12-17 13:10	28M	
<a href="#">cmake-3.1.0-Darwin64-universal.tar.Z</a>	2014-12-17 13:10	40M	
<a href="#">cmake-3.1.0-Darwin64-universal.tar.gz</a>	2014-12-17 13:10	28M	
<a href="#">cmake-3.1.0-Darwin64.dmg</a>	2014-12-19 11:38	29M	
<a href="#">cmake-3.1.0-Darwin64.tar.Z</a>	2014-12-19 11:38	41M	
<a href="#">cmake-3.1.0-Darwin64.tar.gz</a>	2014-12-19 11:38	29M	
<a href="#">cmake-3.1.0-Linux-i386.sh</a>	2014-12-17 13:10	24M	
<a href="#">cmake-3.1.0-Linux-i386.tar.Z</a>	2014-12-17 13:10	33M	
<a href="#">cmake-3.1.0-Linux-i386.tar.gz</a>	2014-12-17 13:10	24M	
<a href="#">cmake-3.1.0-Linux-x86_64.sh</a>	2014-12-17 13:10	25M	
<a href="#">cmake-3.1.0-Linux-x86_64.tar.Z</a>	2014-12-17 13:10	35M	
<a href="#">cmake-3.1.0-Linux-x86_64.tar.gz</a>	2014-12-17 13:10	25M	
<a href="#">cmake-3.1.0-SHA-256.txt</a>	2016-04-13 12:48	1.9K	
<a href="#">cmake-3.1.0-SHA-256.txt.asc</a>	2016-04-13 12:48	819	

**Gambar 1.4** Cari CMake Terbaru pada website ini kemudian install

Saat menjalankan penginstal, pilih Tambahkan CMake ke PATH sistem untuk semua pengguna atau Tambahkan CMake ke PATH sistem untuk pengguna saat ini. Jangan khawatir tentang fakta bahwa CMake versi 64-bit tidak tersedia, CMake hanyalah alat konfigurasi dan tidak melakukan kompilasi apa pun. Sebaliknya, pada Windows, itu menciptakan file proyek yang dapat dibuka dengan Visual Studio.

2. Unduh dan instal Microsoft Visual Studio 2013 (edisi Desktop jika Anda bekerja pada Windows 7) dari

<https://www.visualstudio.com/products/free-developer-offers-vs-community/>

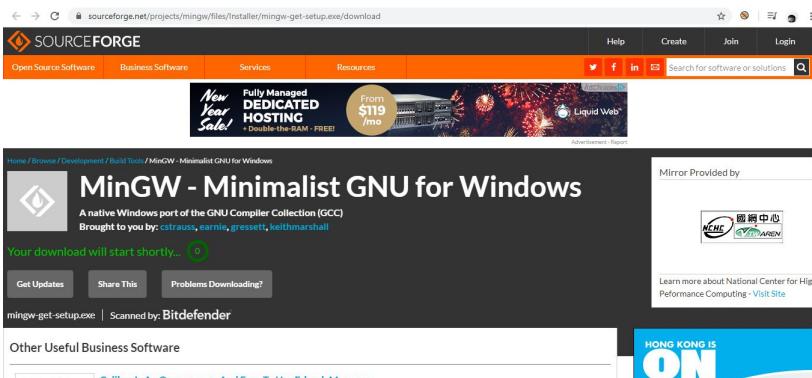


**Gambar 1.5** Cari VisualStudio Terbaru pada website ini kemudian install

Perhatikan bahwa Anda harus masuk dengan akun Microsoft Anda dan jika Anda tidak memiliki, Anda dapat membuatnya di tempat. Instal perangkat lunak dan reboot setelah instalasi selesai.

Untuk MinGW, dapatkan penginstalnya dari

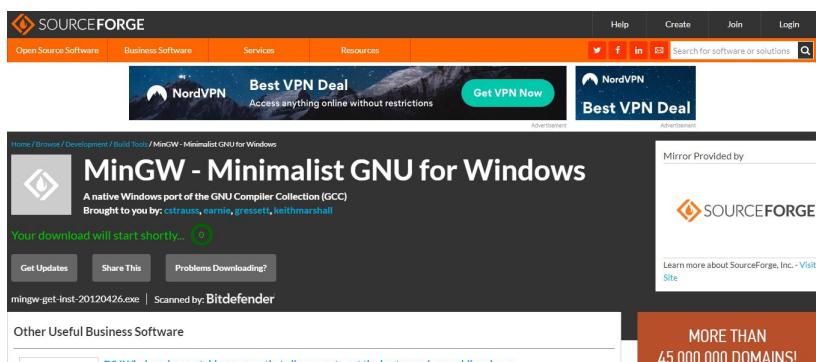
<http://sourceforge.net/projects/mingw/files/Installer/mingw->



Gambar 1.6 Download MinGW

dan

<http://sourceforge.net/projects/mingw/files/OldFiles/mingw->



**Gambar 1.7** Download MinGW

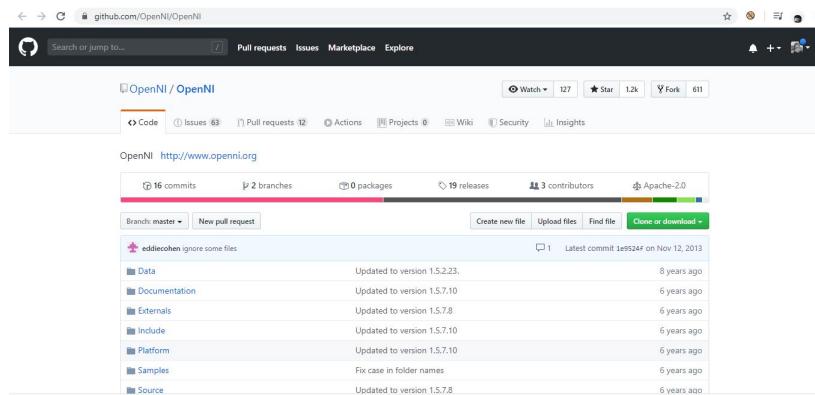
Saat menjalankan penginstal, pastikan jalur tujuan tidak mengandung spasi dan bahwa kompiler C ++ opsional disertakan. Edit variabel PATH sistem dan tambahkan;

C :\MinGW\bin

(dengan asumsi MinGW diinstal ke lokasi default). Mulai ulang sistem.

3. Secara opsional, unduh dan instal OpenNI 1.5.4.0 dari tautan yang disediakan di beranda GitHub di OpenNI di

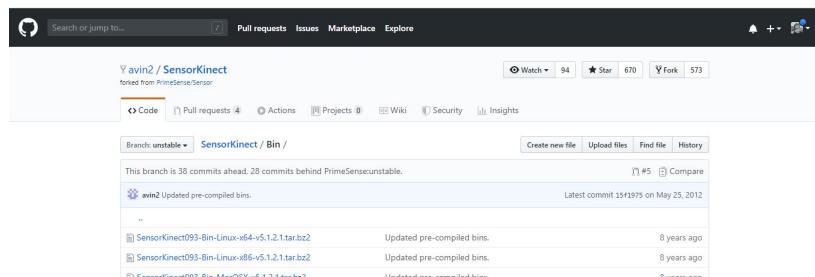
<https://github.com/OpenNI/OpenNI>



**Gambar 1.8** Download OpenNI

4. Anda dapat mengunduh dan menginstal SensorKinect 0.93 dari

<https://github.com/avin2/SensorKinect/blob/unstable/Bin/Sens>

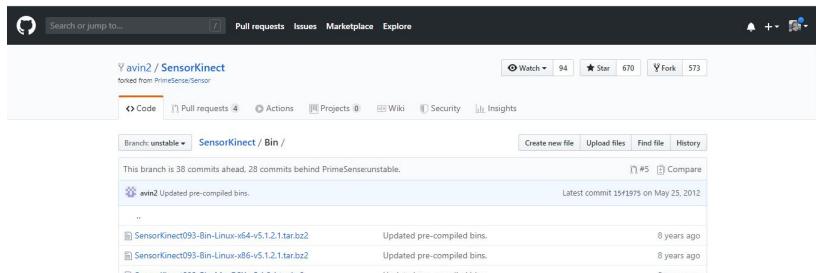


**Gambar 1.9** Download SensorKinect

Atau, untuk Python 64-bit, unduh di

<https://github.com/avin2/SensorKinect/blob/unstable/Bin/SensorKinect-64bit.zip>

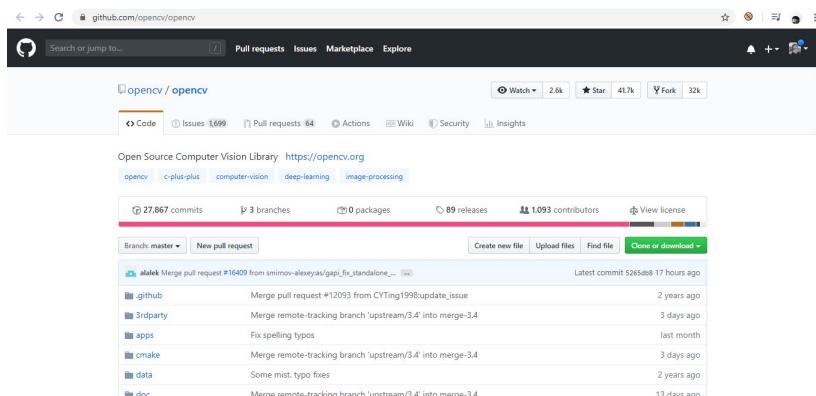
(64-bit). Perhatikan bahwa repositori ini tidak aktif selama lebih dari tiga tahun.



**Gambar 1.10** Download SensorKinect

## 5. Unduh ZIP self-extracting dari OpenCV 3.0.0 dari

<https://github.com/Itseez/opencv>



**Gambar 1.11** Download self-extracting

Jalankan ZIP yang mengekstraksi sendiri, dan ketika diminta, masukkan folder tujuan apa pun, yang akan kita sebut sebagai

<unzip\_destination>. Subfolder, <unzip\_destination>\opencv

kemudian dibuat.

6. Buka Command Prompt dan buat folder lain tempat build kita akan menggunakan ini perintah:

mkdir <build\_folder> Ubah direktori folder build: cd <build>

7. Sekarang, kami siap mengkonfigurasi bangunan kami. Untuk memahami semua opsi, kita dapat membaca kode di

<unzip\_destination>\opencv\CMakeLists.txt

Namun, untuk tujuan buku ini, kita hanya perlu menggunakan opsi yang akan memberi kita rilis dengan binding Python, dan secara opsional, kedalaman dukungan kamera melalui OpenNI dan SensorKinect.

8. Buka CMake (cmake-gui) dan tentukan lokasi kode sumber OpenCV dan folder tempat Anda ingin membangun pustaka. Klik pada Konfigurasi. Pilih proyek yang akan dihasilkan. Dalam hal ini, pilih Visual Studio 12 (yang sesuai dengan Visual Studio 2013). Setelah CMake selesai mengkonfigurasi proyek, itu akan menampilkan daftar opsi build. Jika Anda melihat latar belakang merah, itu berarti bahwa proyek Anda mungkin perlu dikonfigurasi ulang: CMake mungkin melaporkan bahwa ia gagal menemukan beberapa dependensi. Banyak dependensi OpenCV adalah opsional, jadi jangan terlalu khawatir. **Catatan** Jika build gagal diselesaikan atau Anda mengalami masalah di kemudian hari, coba instal dependensi yang hilang (sering kali tersedia sebagai binari prebuilt), dan kemudian bangun kembali OpenCV dari langkah ini. Anda memiliki opsi untuk memilih / membatalkan pilihan opsi bangunan (sesuai dengan perpustakaan yang telah Anda instal pada mesin Anda) dan klik Konfigurasi lagi, sampai Anda mendapatkan latar belakang yang jelas (putih).
9. Di akhir proses ini, Anda dapat mengeklik Hasilkan, yang akan membuat file OpenCV.sln di folder yang Anda pilih untuk membangun. Anda kemudian dapat menavigasi ke

<build\_folder>/OpenCV.sln

dan membuka file dengan Visual Studio 2013, dan melanjutkan dengan membangun proyek,

ALL\_BUILD

- . Anda perlu membangun versi Debug dan Rilis dari OpenCV, jadi lanjutkan dan bangun perpustakaan dalam mode Debug, lalu pilih Lepaskan dan bangun kembali (F7 adalah kunci untuk meluncurkan bangunan).
10. Pada tahap ini, Anda akan memiliki folder bin di direktori build OpenCV, yang akan berisi semua file .dll yang dihasilkan yang memungkinkan Anda untuk memasukkan OpenCV dalam proyek Anda. Atau, untuk MinGW, jalankan perintah berikut:

```
cmake -D: CMAKE_BUILD_TYPE = RELEASE -D: WITH_OPENNI = ON -D:  
"MinGWMakefiles" <unzip_destination>\opencv
```

Jika OpenNI tidak diinstal, abaikan -D: WITH\_OPENNI = ON. (D:  
<openni\_install\_destination>\Lib -D: OPENNI\_INCLUDE\_DIR =  
<openni\_install\_destination>\Sertakan -  
D: OPENNI\_PRIME\_SENSOR\_MODULE\_BIN\_DIR =  
<sensorkinect\_install\_destination>\Sensor\Bin.

Atau, untuk MinGW, jalankan perintah ini:

```
mingw32-make
```

## 11. Salin

```
<build_folder> \ lib \ Release \ cv2.pyd (dari Visual Studio)  
<build_folder> \ lib \ cv2.pyd (dari build MinGW) hingga  
<python_installation_folder> \ paket-situs.
```

## 12. Terakhir, edit variabel PATH sistem dan tambahkan

```
<build_folder>/bin/Release
```

(untuk build Visual Studio) atau

```
<build_folder>/bin
```

(untuk build MinGW). Mulai ulang sistem anda.

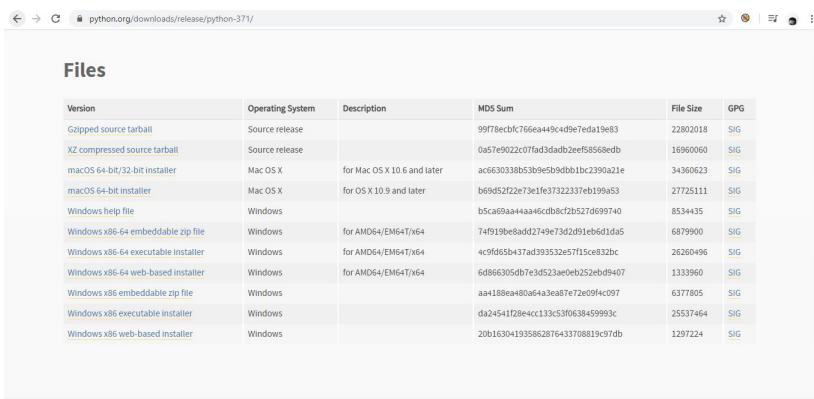
### 1.3.2 Instalasi pada OS X

Beberapa versi Mac digunakan dengan versi Python 2.7 yang sudah diinstal sebelumnya yang disesuaikan oleh Apple untuk kebutuhan internal sistem. Namun, ini telah berubah dan versi standar OS X dikirimkan dengan instalasi standar Python. Di python.org, Anda juga dapat menemukan biner universal yang kompatibel dengan sistem Intel baru dan PowerPC lama.

#### Catatan

Anda dapat memperoleh penginstal ini di

<https://www.python.org/downloads/release/python-371/>



The screenshot shows a table titled "Files" listing various Python 3.7.1 download options. The columns are: Version, Operating System, Description, MD5 Sum, File Size, and GPG. The table includes rows for Gzipped source tarball, XZ compressed source tarball, macOS 64-bit/32-bit installer, macOS 64-bit installer, Windows help file, Windows x86-64 embeddable zip file, Windows x86-64 executable installer, Windows x86-64 web-based installer, Windows x86 embeddable zip file, Windows x86 executable installer, and Windows x86 web-based installer.

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		99f78ebfb766ea449c4db9e7eda19e83	22802018	SIG
XZ compressed source tarball	Source release		0a87e6902207fad3adbd2ee58598edb	16960960	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	ac66303385b53b9e5b0db1bc2390a21e	34360623	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	b69d52f2ze73fe3732337eb199a53	27725111	SIG
Windows help file	Windows		b5ca9aa4aa44e4cd8cf2b52d99740	8534435	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	74f919be8add2749e73d2d91eb61da5	6879900	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	4c9fd65b437a3d9352e2ef71f5ce3232	26260496	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	6d866305db7e3d523ae0eb252ebd9407	1333960	SIG
Windows x86 embeddable zip file	Windows		aa189ea80b064a3e87e72609fc097	6377805	SIG
Windows x86 executable installer	Windows		da24541f28aeccc133c53f0638459993c	25537464	SIG
Windows x86 web-based installer	Windows		20b16304193586287643370819c97db	1297224	SIG

**Gambar 1.12** Download Python

(lihat PPC Mac OS X 32-bit atau tautan Intel® Mac OS X 64-bit). Menginstal Python dari file .dmg yang diunduh hanya akan menimpa instalasi sistem Python Anda saat ini.

Untuk Mac, ada beberapa pendekatan yang mungkin untuk mendapatkan standar Python 2.7, NumPy, SciPy, dan OpenCV. Semua pendekatan pada akhirnya membutuhkan OpenCV untuk dikompilasi dari sumber menggunakan Alat Pengembang Xcode. Namun, tergantung pada pendekatannya, tugas ini otomatis bagi kami dalam berbagai cara oleh alat pihak ketiga. Kami akan melihat pendekatan semacam ini menggunakan MacPorts atau Homebrew. Alat-alat ini berpotensi melakukan segala yang dapat dilakukan CMake, ditambah lagi membantu kami mengatasi dependensi dan memisahkan pustaka pengembangan kami dari pustaka sistem.

**Tip**

Saya merekomendasikan MacPorts, terutama jika Anda ingin mengkompilasi OpenCV dengan dukungan kamera mendalam melalui OpenNI dan SensorKinect. Tambalan dan skrip yang relevan, termasuk beberapa yang saya kelola, siap pakai untuk MacPorts. Sebaliknya, Homebrew saat ini tidak memberikan solusi yang sudah jadi untuk mengkompilasi OpenCV dengan dukungan kamera yang dalam. Sebelum melanjutkan, pastikan bahwa Alat Pengembang Xcode disiapkan dengan benar:

Unduh dan instal Xcode dari Mac App Store atau

<https://developer.apple.com/xcode/downloads/>

Selama instalasi, jika ada opsi untuk menginstal Command Line Tools, pilihlah.

Buka Xcode dan terima perjanjian lisensi.

Langkah terakhir diperlukan jika penginstal tidak memberi kami opsi untuk menginstal Alat Baris Perintah. Arahkan ke Xcode, Preferensi, Unduh, dan klik tombol Instal di sebelah Command Line Tools. Tunggu instalasi untuk menyelesaikan dan keluar dari Xcode.

Atau, Anda dapat menginstal alat baris perintah Xcode dengan menjalankan perintah berikut (di terminal):

```
$ xcode-select {install}
```

Sekarang, kami memiliki kompiler yang diperlukan untuk pendekatan apa pun.

Menggunakan MacPorts dengan paket yang sudah jadi

Kita bisa menggunakan manajer paket MacPorts untuk membantu kita mengatur Python 2.7, NumPy, dan OpenCV. MacPorts menyediakan perintah terminal yang mengotomatiskan proses mengunduh, mengkompilasi, dan menginstal berbagai perangkat lunak sumber terbuka (OSS). MacPorts juga menginstal dependensi sesuai kebutuhan. Untuk setiap perangkat lunak, dependensi dan resep bangunan didefinisikan dalam file konfigurasi yang disebut Portfile. Repozitori MacPorts adalah kumpulan dari Portfiles.

Mulai dari sistem di mana Xcode dan alat-alat command-line-nya sudah diatur, langkah-langkah berikut akan memberi kita instalasi OpenCV melalui MacPorts:

1. Unduh dan instal MacPorts dari

[http://www.macports.org/install.php.](http://www.macports.org/install.php)

The screenshot shows the MacPorts website with the URL [macports.org/install.php](http://macports.org/install.php) in the address bar. The page title is "Mac Ports". A "Download" button is visible in the top right. The main navigation menu includes Home, **Installing MacPorts**, Available Ports, Documentation, Support & Development, Contact Us, and News. Below the menu, there's a sidebar with links to Available Downloads, MacPorts FAQ, Report a Bug, Bug reporting Guidelines, Git Repository, MacPorts Team, and Becoming a Member. The main content area is titled "Quickstart" and lists three steps: 1. Install Xcode and the Xcode Command Line Tools, 2. Agree to Xcode license in Terminal: `sudo xcodebuild -license`, and 3. Install MacPorts for your version of the Mac operating system, with options for macOS Catalina v10.15, macOS Mojave v10.14, macOS High Sierra v10.13, macOS Sierra v10.12, and Older OS? See here.

### Installing MacPorts

MacPorts version 2.6.2 is available in various formats for download and installation (note, if you are upgrading to a new major release of macOS, see the [migration info page](#)):

### Gambar 1.13 Download MacPorts

2. Jika Anda ingin dukungan untuk kamera kedalaman Kinect, Anda perlu memberi tahu MacPorts tempat untuk mengunduh Portfile khusus yang telah saya tulis. Untuk melakukannya, edit

/opt/local/etc/macports/sources.conf (dengan asumsi bahwa MacPorts diinstal ke lokasi default). Tepat di atas garis,

`rsync: //rsync.macports.org/release/ports/ [default]`, tambahkan baris berikut:

`http://nummist.com/opencv/ports.tar.gz`

Simpan file. Sekarang, MacPorts tahu bahwa ia harus mencari Portfile di repositori online saya terlebih dahulu, dan kemudian repositori online default.

3. Buka terminal dan jalankan perintah berikut untuk memperbarui MacPorts:

```
$ sudo port selfupdate
```

Saat diminta, masukkan kata sandi Anda.

4. Sekarang (jika kita menggunakan repositori saya), jalankan perintah berikut untuk menginstal OpenCV dengan binding Python 2.7 dan dukungan untuk kamера kedalaman, termasuk Kinect:

```
$ sudo port instal opencv + python27 + openni_sensorkinect
```

Atau (dengan atau tanpa repositori saya), jalankan perintah berikut untuk menginstal OpenCV dengan binding Python 2.7 dan dukungan untuk kamera kedalaman, tidak termasuk Kinect:

```
$ sudo port instal opencv + python27 + openni
```

### Catatan

Ketergantungan, termasuk Python 2.7, NumPy, OpenNI, dan (dalam contoh pertama) SensorKinect, diinstal secara otomatis juga. Dengan menambahkan python27 ke perintah, kami menentukan bahwa kami ingin varian OpenCV (membangun konfigurasi) dengan Python 2.7 binding. Demikian pula, menambahkan openni\_sensorkinect menentukan varian dengan dukungan seluas mungkin untuk kamera kedalaman melalui OpenNI dan SensorKinect. Anda dapat menghilangkan + openni\_sensorkinect jika Anda tidak bermaksud menggunakan kamera kedalaman, atau Anda dapat menggantinya dengan + openni jika Anda bermaksud menggunakan kamera kedalaman yang kompatibel dengan OpenNI tetapi tidak dengan Kinect. Untuk melihat daftar lengkap varian yang tersedia sebelum menginstal, kita dapat memasukkan perintah berikut:

```
$ port varian opencv
```

Bergantung pada kebutuhan penyesuaian kami, kami dapat menambahkan varian lain ke perintah pemasangan. Untuk fleksibilitas yang lebih besar, kita dapat menulis varian kita sendiri (seperti yang dijelaskan di bagian selanjutnya).

5. Juga, jalankan perintah berikut untuk menginstal SciPy:

```
$ sudo port install py27-scipy
```

6. Eksekusi instalasi Python bernama python2.7. Jika kita ingin menautkan python default yang dapat dieksekusi ke python2.7, mari kita juga jalankan perintah ini:

```
$ sudo port install python_select  
$ sudo port pilih python python27
```

## Menggunakan MacPorts dengan paket kustom Anda sendiri

Dengan beberapa langkah tambahan, kita dapat mengubah cara MacPorts mengkomplisi OpenCV atau perangkat lunak lainnya. Seperti disebutkan sebelumnya, resep build MacPorts didefinisikan dalam file konfigurasi yang disebut Portfiles. Dengan membuat atau mengedit Portfile, kita dapat mengakses alat bantu yang sangat dapat dikonfigurasi, seperti CMake, sementara juga memanfaatkan fitur MacPorts, seperti resolusi ketergantungan.

Mari kita asumsikan bahwa kita sudah menginstal MacPorts. Sekarang, kita dapat mengkonfigurasi MacPorts ke gunakan Portfile khusus yang kita tulis:

1. Buat folder di suatu tempat untuk menampung Portfiles khusus kami. Kami akan merujuk ke folder ini sebagai

```
<local_repository>
```

2. Edit file /opt/local/etc/macports/sources.conf (dengan anggapan bahwa MacPorts diinstal ke lokasi default). Tepat di atas

```
rsync: //rsync.macports.org/release/ports/ default baris, t  
file: // <local_repository>
```

Misalnya, jika <local\_repository> adalah / Users / Joe / Portfiles

Catat tiga tebasan dan simpan file. Sekarang, MacPorts tahu bahwa ia harus mencari Portfiles di

```
<local_repository>
```

terlebih dahulu, dan kemudian, repositori online default-nya.

3. Buka terminal dan perbarui MacPorts untuk memastikan bahwa kami memiliki Portfile terbaru dari repositori default:

```
$ sudo port selfupdate
```

4. Mari kita salin opencv Portfile repositori default sebagai contoh. Kami juga harus menyalin struktur direktori, yang menentukan bagaimana paket dikategorikan oleh MacPorts:

```
$ mkdir <local_repository> / graphics /  
$ cp  
/opt/local/var/macports/sources/rsync.macports.org/release/p  
cs / opencv <local_repository> / grafis
```

Sebagai alternatif, untuk contoh yang menyertakan dukungan Kinect, kita dapat mengunduh repositori online saya dari <http://nummist.com/opencv/ports.tar.gz>, unzip, dan salin seluruh folder grafiknya ke



**Gambar 1.14** Download Kinect

<local\_repository>

:

```
$ cp <unzip_destination> / graphics <local_repository>
```

## 5. Edit

<local\_repository>/graphics/opencv/Portfile.

Perhatikan bahwa file ini menentukan flag konfigurasi, dependensi, dan varian CMake. Untuk detail tentang pengeditan Portfile, buka

<http://guide.macports.org/#development>.

The screenshot shows the MacPorts Guide website. At the top, there's a header with the MacPorts logo and navigation links. Below the header, the main content area has a sidebar on the left containing a table of contents for various sections like 'Introduction', 'Using MacPorts', and 'Portfile Development'. The main content area features a large section titled 'Chapter 1. Introduction' with several bullet points explaining the benefits of using MacPorts. At the bottom of this section, there's a note about MacPorts being developed on macOS but working on other Unix-like systems. To the right of the main content, there's a sidebar with three boxes: 'Terminal I/O and file text.', '\$ Commands to be typed into a terminal window.', 'Command output to a terminal window.', and 'File text.'

**Gambar 1.15** Tutorial pada website tersebut

Untuk melihat flag konfigurasi CMake mana yang relevan dengan OpenCV, kita perlu melihat kode sumbernya. Unduh arsip kode sumber dari [6. Sekarang, kita perlu membuat file indeks di repositori lokal kita sehingga MacPorts dapat menemukan Portfile baru:](https://github.com/unzip ke lokasi mana pun, dan baca <unzip_destination> /OpenCV-3.0.0/CM Setelah melakukan pengeditan ke Portfile, simpanlah.</a></p>
</div>
<div data-bbox=)

```
$ cd <local_repository>
$ portindex
```

7. Mulai sekarang, kita dapat memperlakukan file pembuka kustom kita sama seperti paket MacPorts lainnya. Sebagai contoh, kita dapat menginstalnya sebagai berikut:

```
$ sudo port instal opencv + python27 + openni_sensorkinect
```

Perhatikan bahwa Portfile repositori lokal kami lebih diutamakan daripada Portfile repositori default karena urutan urutannya. `/opt/local/etc/macports/sources.conf`.

## Menggunakan Homebrew dengan paket yang sudah jadi (tidak ada dukungan untuk kamera kedalaman)

Homebrew adalah manajer paket lain yang dapat membantu kami. Biasanya, MacPorts dan Homebrew tidak boleh diinstal pada mesin yang sama.

Mulai dari sistem di mana Xcode dan alat-alat command-line-nya sudah diatur, langkah-langkah berikut akan memberi kita instalasi OpenCV melalui Homebrew:

1. Buka terminal dan jalankan perintah berikut untuk menginstal Homebrew:

```
$ ruby -e "$(curl -fsSkLraw.githubusercontent.com/mxcl/homebrew/go)"
```

2. Tidak seperti MacPorts, Homebrew tidak secara otomatis menempatkan executable-nya di PATH. Untuk melakukannya, buat atau edit file

```
~/.profile
```

dan tambahkan baris ini di bagian atas kode:

```
ekspor PATH = /usr/local/bin:/usr/local/sbin:$
```

Simpan file dan jalankan perintah ini untuk menyegarkan PATH:

```
$ source ~/.profile
```

Perhatikan bahwa executable yang diinstal oleh Homebrew sekarang diutamakan daripada executable yang diinstal oleh sistem.

3. Untuk laporan diagnostik mandiri Homebrew, jalankan perintah berikut:

```
$ brew doctor
```

Ikuti saran pemecahan masalah yang diberikannya.

4. Sekarang, perbarui Homebrew:

```
$ brew update
```

5. Jalankan perintah berikut untuk menginstal Python 2.7:

```
$ brew install python
```

6. Sekarang, kita dapat menginstal NumPy. Pilihan paket pustaka Python Homebrew terbatas, jadi kami menggunakan alat manajemen paket terpisah yang disebut pip, yang dilengkapi dengan Homebrew Python:

```
$ pip install numpy
```

7. SciPy berisi beberapa kode Fortran, jadi kita memerlukan kompiler yang sesuai. Kita dapat menggunakan Homebrew untuk menginstal kompiler gfortran:

```
$ brew install gfortran
```

Sekarang, kita dapat menginstal SciPy:

```
$ pip install scipy
```

8. Untuk menginstal OpenCV pada sistem 64-bit (semua perangkat keras Mac baru sejak akhir 2006), jalankan perintah berikut:

**Tip****Mengunduh kode contoh**

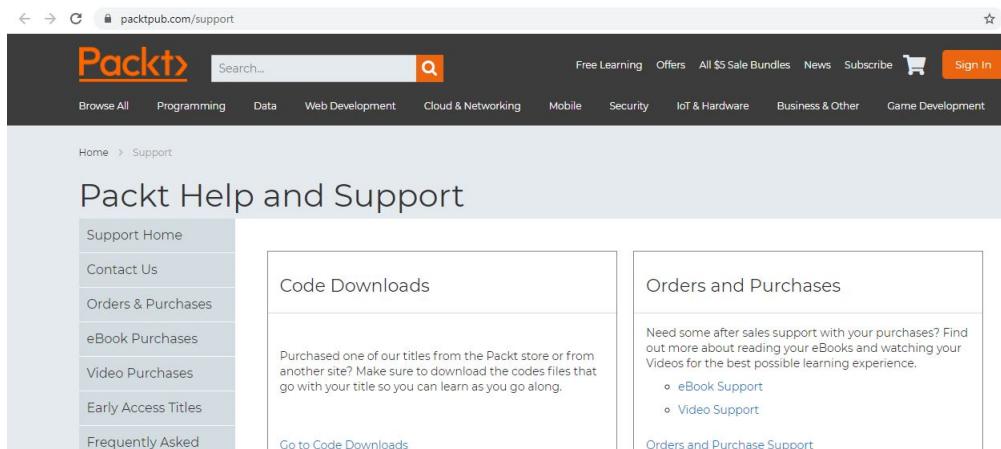
Anda dapat mengunduh file kode contoh untuk semua buku Penerbitan Packt yang telah Anda beli dari akun Anda di

<http://www.packtpub.com>.

Jika Anda membeli buku ini di tempat lain, Anda dapat mengunjungi

<http://www.packtpub.com/support>

dan mendaftar agar file-file tersebut diemail langsung kepada Anda.



**Gambar 1.16** Website packtpub

### Menggunakan Homebrew dengan paket kustom Anda sendiri

Homebrew memudahkan untuk mengedit definisi paket yang ada:

```
$ brew edit opencv
```

Definisi paket sebenarnya adalah skrip dalam bahasa pemrograman Ruby. Kiat untuk mengeditnya dapat ditemukan di halaman Wiki Homebrew di

<https://github.com/mxcl/homebrew/wiki/Formula-Cookbook>.

Sebuah skrip dapat menentukan flag konfigurasi Make atau CMake, antara lain.

Untuk melihat flag konfigurasi CMake mana yang relevan dengan OpenCV, kita perlu melihat kode sumbernya. Unduh arsip kode sumber dari

<https://github.com/Itseez/opencv/archive/3.0.0.zip>

, unzip ke lokasi mana pun, dan baca

`<unzip_destination>/OpenCV-2.4.3/CMakeLists.txt.`

Setelah mengedit skrip Ruby, simpan. Paket khusus dapat diperlakukan seperti biasa. Misalnya, dapat diinstal sebagai berikut:

```
$ brew install opencv
```

### 1.3.3 Instalasi pada Ubuntu

Pertama dan terpenting, berikut adalah catatan singkat tentang versi Ubuntu dari sistem operasi: Ubuntu memiliki siklus rilis 6 bulan di mana setiap rilis adalah versi minor .04 atau .10 dari versi utama (14 pada saat penulisan). Namun, setiap dua tahun, Ubuntu merilis versi yang diklasifikasikan sebagai dukungan jangka panjang (LTS) yang akan memberi Anda dukungan lima tahun oleh Canonical (perusahaan di belakang Ubuntu). Jika Anda bekerja di lingkungan perusahaan, disarankan untuk menginstal salah satu versi LTS. Yang terbaru yang tersedia adalah 14.04.

Ubuntu hadir dengan Python 2.7 yang sudah diinstal. Repozitori Ubuntu standar berisi paket OpenCV 2.4.9 tanpa dukungan untuk kamera yang dalam. Pada saat penulisan ini, OpenCV 3 belum tersedia melalui repositori Ubuntu, jadi kita harus membuatnya dari sumber. Untungnya, sebagian besar sistem Unix-like dan Linux datang dengan semua perangkat lunak yang diperlukan untuk membangun proyek dari awal yang sudah diinstal. Ketika dibangun dari sumber, OpenCV dapat mendukung kamera kedalaman melalui OpenNI dan SensorKinect, yang tersedia sebagai binari yang dikompilasi dengan skrip instalasi.

#### **Menggunakan repositori Ubuntu (tidak ada dukungan untuk kamera kedalaman)**

Kita dapat menginstal Python dan semua dependensi yang diperlukan menggunakan manajer paket apt, dengan menjalankan perintah berikut:

```
> sudo apt-get install build-essential  
> sudo apt-get install cmake git libgtk2.0-dev pkg-config  
libavformat-dev libswscale-dev  
> sudo apt-get install python-dev python-numpy libtbb2 li
```

Secara setara, kita bisa menggunakan Ubuntu Software Center, yang merupakan tampilan grafis apt package manager.

**Membangun OpenCV dari sumber** Sekarang kita telah menginstal seluruh tumpukan Python dan cmake, kita dapat membangun OpenCV. Pertama, kita perlu mengunduh kode sumbernya <https://github.com/Itseez/opencv> strak arsip dan pindahkan ke folder yang tidak di-zip di terminal. Kemudian, jalankan perintah berikut:

```
> mkdir build  
> cd build  
> cmake -D CMAKE_BUILD_TYPE =Release -D CMAKE_INSTALL_PREFIX=  
> make  
> make install
```

Setelah instalasi berakhir, Anda mungkin ingin melihat contoh Python OpenCV di <opencv\_folder>/opencv/samples/python dan <script\_folder>. Instalasi pada sistem mirip Unix lainnya Pendekatan untuk Ubuntu (seperti yang dijelaskan sebelumnya) kemungkinan akan bekerja pada distribusi Linux apa pun yang berasal dari Ubuntu 14.04 LTS atau Ubuntu 14.10 sebagai berikut: Kubuntu 14.04 LTS atau Kubuntu 14.10 Xubuntu 14.04 LTS atau Xubuntu 14.10 Linux Mint 17

Pada Debian Linux dan turunannya, manajer paket apt berfungsi sama seperti di Ubuntu, meskipun paket yang tersedia mungkin berbeda.

Di Gentoo Linux dan turunannya, manajer paket Portage mirip dengan MacPorts (seperti dijelaskan sebelumnya), meskipun paket yang tersedia mungkin berbeda.

Pada turunan FreeBSD, proses instalasi sekali lagi mirip dengan MacPorts; sebenarnya, MacPorts berasal dari sistem instalasi port yang diadopsi pada FreeBSD. Bacalah Buku Pegangan FreeBSD yang luar biasa di <https://www.freebsd.org/doc/handbook/> untuk ikhtisar proses instalasi perangkat lunak.

The screenshot shows the FreeBSD Handbook homepage. At the top, there's a header with the title "FreeBSD Handbook". Below the header, there's a "Copyright" notice and a "Last modified" timestamp. A "Abstract" section follows, containing a brief introduction to the handbook. Navigation links like "Table of Contents" and "Preface" are also visible.

### Gambar 1.17 FreeBSD

Pada sistem mirip Unix lainnya, manajer paket dan paket yang tersedia mungkin berbeda. Konsultasikan dokumentasi manajer paket Anda dan cari paket dengan opencv di namanya. Ingatlah bahwa OpenCV dan binding Python-nya dapat dipecah menjadi beberapa paket.

Juga, cari semua catatan instalasi yang diterbitkan oleh penyedia sistem, pengelola repositori, atau komunitas. Karena OpenCV menggunakan driver kamera dan codec media, membuat semua fungsinya berfungsi dapat menjadi rumit pada sistem dengan dukungan multimedia yang buruk. Dalam beberapa keadaan, paket sistem mungkin perlu dikonfigurasi ulang atau diinstal ulang untuk kompatibilitas.

Jika paket tersedia untuk OpenCV, periksa nomor versinya. OpenCV 3 atau lebih tinggi direkomendasikan untuk tujuan buku ini. Juga, periksa apakah paket menawarkan binding Python dan dukungan kamera mendalam melalui OpenNI dan SensorKinect. Terakhir, periksa apakah ada orang di komunitas pengembang yang melaporkan keberhasilan atau kegagalan dalam menggunakan paket.

Jika, sebaliknya, kami ingin melakukan pembuatan kustom OpenCV dari sumber, mungkin bermanfaat untuk merujuk ke skrip instalasi untuk Ubuntu (seperti yang dibahas sebelumnya) dan menyesuaikannya dengan manajer paket dan paket yang ada di sistem lain.

### 1.3.4 Instalasi modul Contrib

Berbeda dengan OpenCV 2.4, beberapa modul terdapat dalam repositori yang disebut `opencv_contrib`, yang tersedia di [https://github.com/Itseez/opencv\\_contrib](https://github.com/Itseez/opencv_contrib). Saya sangat merekomendasikan menginstal modul ini karena mengandung fungsionalitas tambahan yang tidak termasuk dalam OpenCV, seperti modul pengenalan wajah. Setelah diunduh (baik melalui zip atau git, saya sarankan git agar Anda dapat tetap up to date dengan perintah `git pull` sederhana), Anda dapat menjalankan kembali perintah `cmake` Anda untuk memasukkan pembangunan OpenCV dengan modul `opencv_contrib` sebagai berikut:

```
cmake -DOPENCV_EXTRA_MODULES_PATH = <opencv_contrib> / modules  
<opencv_source_directory>
```

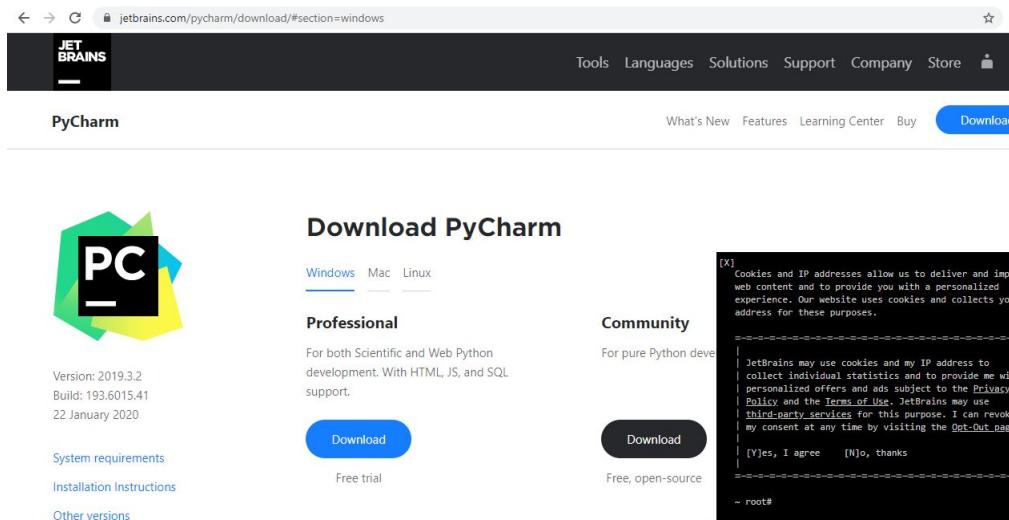
Jadi, jika Anda telah mengikuti prosedur standar dan membuat direktori build di folder unduhan OpenCV Anda, Anda harus menjalankan perintah berikut:

```
mkdir build && cd build  
cmake -D CMAKE_BUILD_TYPE = Lepaskan -DOPENCV_EXTRA_MODULES_PATH  
<opencv_contrib> / modules -D CMAKE_INSTALL_PREFIX = / us  
make
```

### 1.3.5 PyCharm

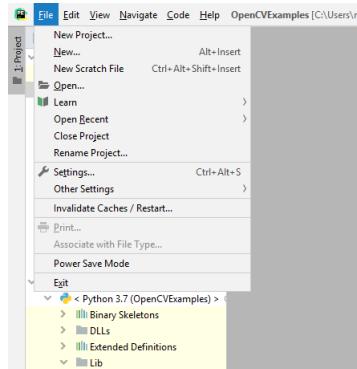
Pycharm merupakan tools untuk menjalankan program python didalamnya sudah terdapat berbagai macam library dari python itu sendiri, kita hanya perlu mencari library yang kita butuhkan kemudian klik install, maka kita tidak perlu melakukan hal hal yang telah di contohkan untuk menginstall opencv pada windows atau yang lainnya. Yang perlu kita lakukan yang pertama kalinya adalah kita mendownload aplikasi PyCharm.

<https://www.jetbrains.com/pycharm/download/#section=windows>



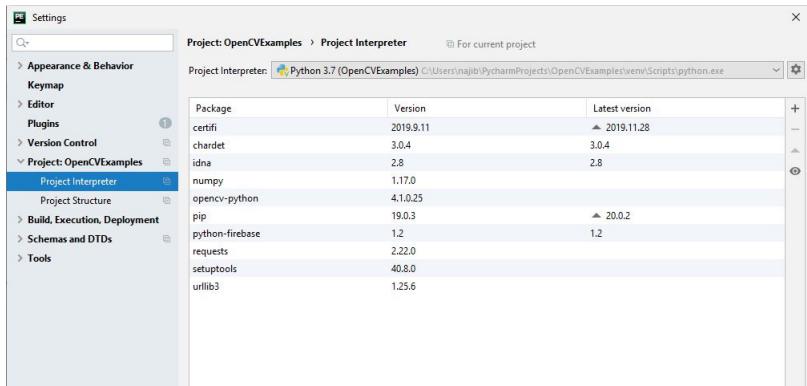
**Gambar 1.18** Pycharm

Kemudian lakukan installasi, setelah istallasi selesai selanjutnya kita buka aplikasi kemudian pilih file kemudian settings.



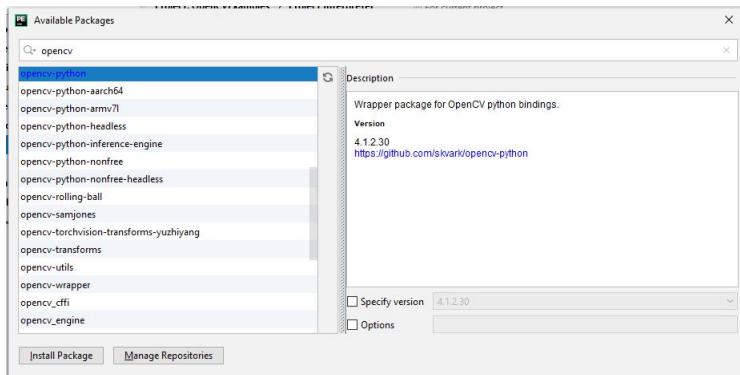
Gambar 1.19 Pycharm Settings

Kemudian pilih project interpreter, lalu klik tambah pada pojok kanan, maka tampilannya akan seperti ini:



Gambar 1.20 Install library

Setelah masuk pada tampilan ini kita cari library apa yang kita butuhkan untuk menjalankan project yang akan kita bangun, karna bahasan kita pada saat ini yaitu OPENCV maka yang kita cari adalah OpenCV, Setelah menemukannya kita langsung saja klik install kira kira membutuhkan waktu lumayan lama, jika internet kita stabil kurang lebih 20-30 menit waktu yang dibutuhkan untuk menginstall opencv ini.



**Gambar 1.21** Install OpenCV

## BAB 2

---

# DASAR DASAR CODE OPENCV

---

### 2.1 Menampilkan gambar

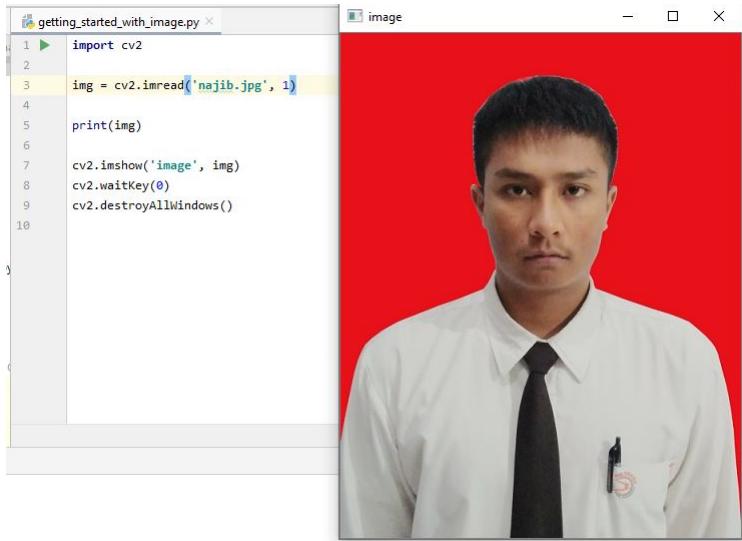
#### 2.1.1 Menampilkan Gambar

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 print(img)
6
7 cv2.imshow('image', img)
8
9 cv2.waitKey(0)
10
11 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file

dan nomor untuk gradiasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.

- iii. lakukan print untuk menampilkan gambar
- iv. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- v. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- vi. destroyAllWindows digunakan untuk menutup frame.



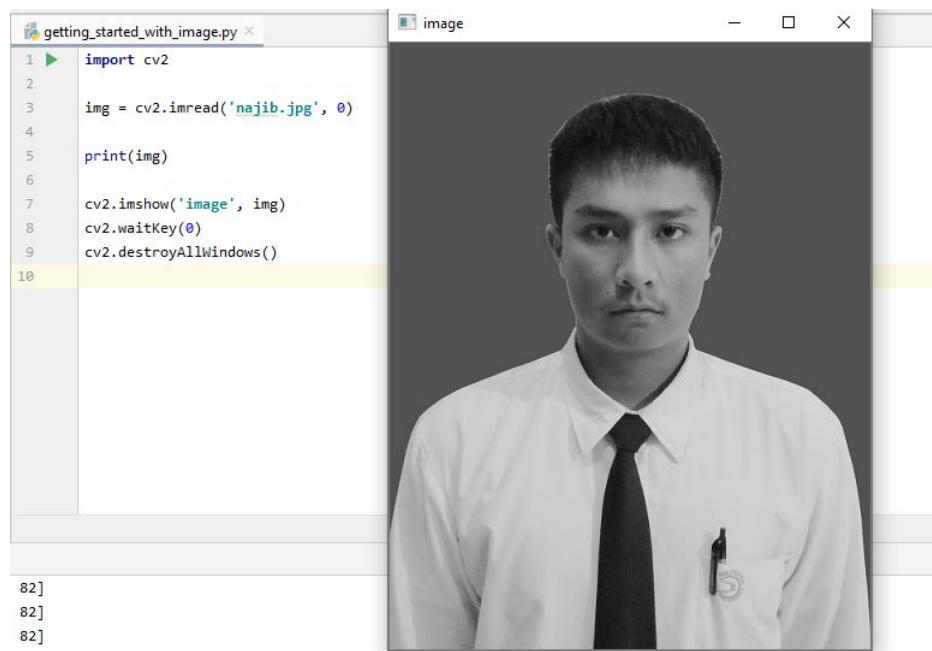
**Gambar 2.1** Menampilkan gambar

Hasil yang ditampilkan sama seperti foto aslinya karna tidak ada dari foto yang di rubah sama sekali, kodingan ini hanya bertujuan untuk menampilkan gambar saja.

## 2.1.2 Menampilkan Gambar dan merubah kontras warnanya

```
1 import cv2  
2  
3 img = cv2.imread('najib.jpg', 0)  
4  
5 print(img)  
6  
7 cv2.imshow('image', img)  
8  
9 cv2.waitKey(0)  
10  
11 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradasi warnanya, pada bagian ini menggunakan angka 0 merubah gambar menjadi hitam putih.
- iii. lakukan print untuk menampilkan gambar
- iv. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- v. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- vi. destroyAllWindows digunakan untuk menutup frame.



The image shows a Python code editor window on the left and a preview window on the right. The code in the editor is:

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 0)
4
5 print(img)
6
7 cv2.imshow('image', img)
8 cv2.waitKey(0)
9 cv2.destroyAllWindows()
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82]
82]
82]
```

The preview window shows a black and white portrait of a young man wearing a white shirt and a dark tie. He has short dark hair and is looking directly at the camera. The background is a solid dark color.

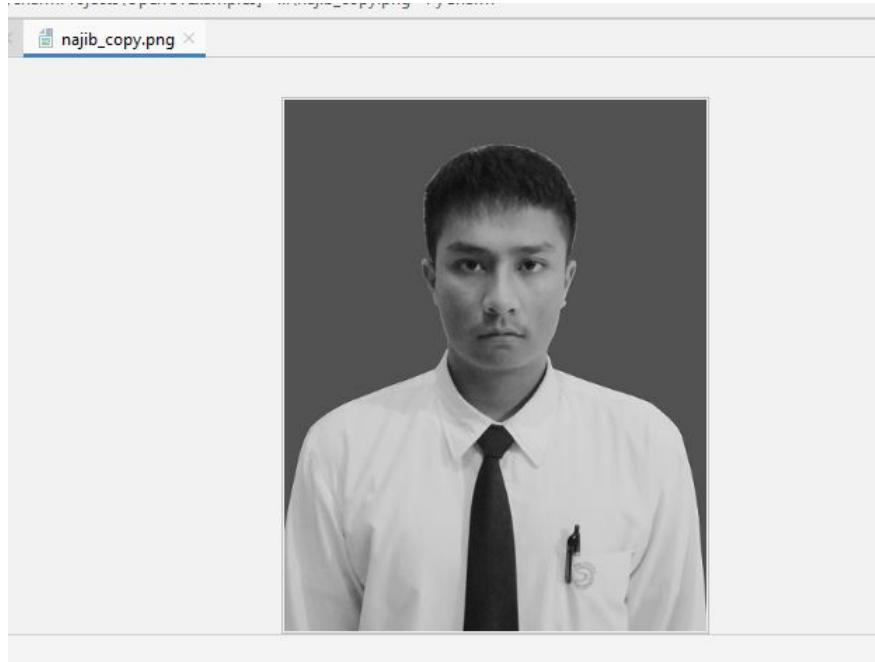
**Gambar 2.2** Merubah kontras warna

Pada bagian kodingan ini foto di rubah kontras warnanya menjadi hitam putih, pada kodingan sebelumnya yang di rubah hanya satu huruf saja untuk menjadikan foto ini menjadi seperti ini. yaitu pada bagian imread nya menjadi 0.

### 2.1.3 Menyimpan Gambar menggunakan kode opencv

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 0)
4
5 print(img)
6
7 cv2.imshow('image', img)
8
9 cv2.waitKey(0)
10
11 cv2.destroyAllWindows()
12
13 cv2.imwrite('najib_copy.png', img)
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradasi warnanya, pada bagian ini menggunakan angka 0 merubah gambar menjadi hitam putih.
- iii. lakukan print untuk menampilkan gambar
- iv. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- v. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- vi. destroyAllWindows digunakan untuk menutup frame.
- vii. terakhir gambar disimpan menggunakan kode imwrite, pertama tuliskan nama gambar yang akan disimpan beserta format gambarnya, kemudian kode img untuk menyatakan yang disimpan tersebut adalah gambar.



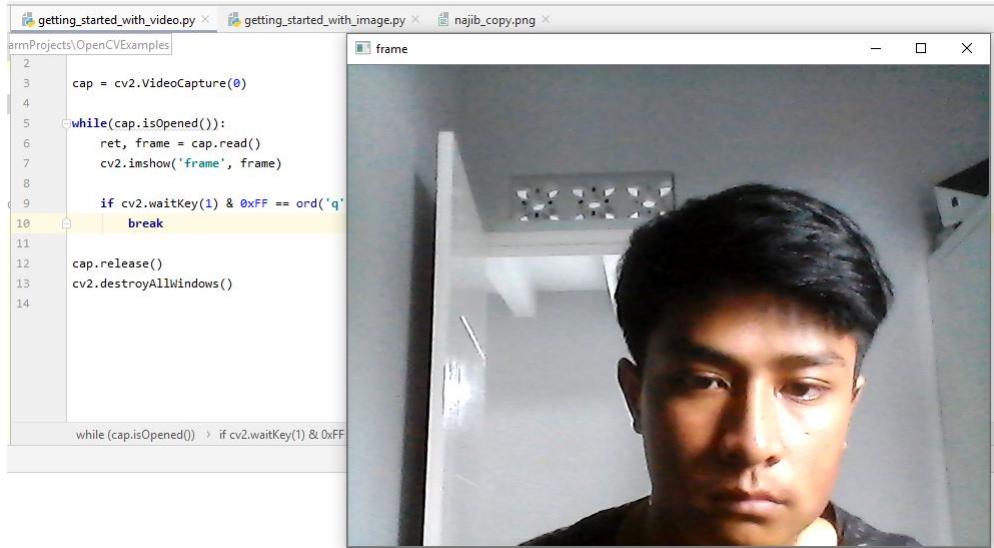
**Gambar 2.3** Menyimpan gambar

Gambar berhasil disimpan dengan nama najibcopy.png, gambar disimpan sesuai yang telah di edit kontras warnanya menjadi hitam putih sesuai cede yang kita jalankan.

## 2.2 Menjalankan kamera lepton

```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 while(cap.isOpened()):
6
7     ret, frame = cap.read()
8
9     cv2.imshow('frame', frame)
10
11    if cv2.waitKey(1) & 0xFF == ord('q'):
12        break
13
14 cap.release()
15
16 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian buat variable baru dengan nama cap kemudian panggil VideoCapture(0) yang artinya menjalankan kamera lepton.
- iii. membuat while yaitu perulangan membuka frame
- iv. kemudian didalam perulangan tersebut terdapat frame yang membaca atau merekam video.
- v. kemudian buat frame dengan nama frame
- vi. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- vii. release untuk menutup videocapture
- viii. destroyAllWindows digunakan untuk menutup frame.



The image shows a screenshot of a Windows desktop environment. On the left, there is a code editor window titled 'getting\_started\_with\_video.py' which contains the following Python code:

```
1  armProjects\OpenCVExamples
2  3      cap = cv2.VideoCapture(0)
4
5  while(cap.isOpened()):
6      ret, frame = cap.read()
7      cv2.imshow('frame', frame)
8
9      if cv2.waitKey(1) & 0xFF == ord('q'):
10         break
11
12     cap.release()
13     cv2.destroyAllWindows()
14
```

On the right, there is a window titled 'frame' displaying a live video feed of a person's face. The video feed is captured from a laptop camera.

**Gambar 2.4** Menggunakan kamera leptop

Pada videocapture ini hanya merekam menggunakan kamera leptop saja belum masuk ke pengolahan gambar.

## 2.2.2 Merubah kontras warna pada video

```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 while(cap.isOpened()):
6
7     ret, frame = cap.read()
8
9     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
10
11    cv2.imshow('frame', gray)
12
13 if cv2.waitKey(1) & 0xFF == ord('q'):
14     break
15
16 cap.release()
17
18 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian buat variable baru dengan nama cap kemudian panggil VideoCapture(0) yang artinya menjalankan kamera laptop.
- iii. membuat while yaitu perulangan membuka frame
- iv. kemudian didalam perulangan tersebut terdapat frame yang membaca atau merekam video.
- v. buat variable dengan nama gray karna kita mau berubah kontras warnanya menjadi hitam putih, kemudian panggil cvtColor didalam frame dengan warna abu abu.
- vi. kemudian buat frame dengan nama frame
- vii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- viii. release untuk menutup videocapture
- ix. destroyAllWindows digunakan untuk menutup frame.



```
getting_started_with_video.py x getting_started_with_image.py x nabil_copy.png x
armProjects\OpenCVExamples [frame]
2
3
4
5     cap = cv2.VideoCapture(0)
6
7     while(cap.isOpened()):
8         ret, frame = cap.read()
9
10        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
11        cv2.imshow('frame', gray)
12
13        if cv2.waitKey(1) & 0xFF == ord('q'):
14            break
15
16
17        cap.release()
18        cv2.destroyAllWindows()
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
```

**Gambar 2.5** Kontras warna video

Pada videocapture ini video sudah di rubah kontras warnanya menjadi abu abu, kita bisa rubah sesuai yang kita inginkan.

### 2.2.3 Mengetahui ukuran frame yang ditampilkan

```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 while(cap.isOpened()):
6
7     ret, frame = cap.read()
8
9     print(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
10
11    print(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
12
13 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
14
15 cv2.imshow('frame', gray)
16
17 if cv2.waitKey(1) & 0xFF == ord('q'):
18     break
19
20 cap.release()
21
22 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian buat variable baru dengan nama cap kemudian panggil VideoCapture(0) yang artinya menjalankan kamera laptop.
- iii. membuat while yaitu perulangan membuka frame
- iv. kemudian didalam perulangan tersebut terdapat frame yang membaca atau merekam video.
- v. kita cukup print mengambil dari videocapture dan panggil CAP PROP FRAME WIDTH untuk mengetahui ukuran lebarnya dan CAP PROP FRAME HEIGHT untuk ukuran tingginya
- vi. buat variable dengan nama gray karna kita mau berubah kontras warnanya menjadi hitam putih, kemudian panggil cvtColor didalam frame dengan warna abu abu.
- vii. kemudian buat frame dengan nama frame
- viii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- ix. release untuk menutup videocapture
- x. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.6** Ukuran Frame video

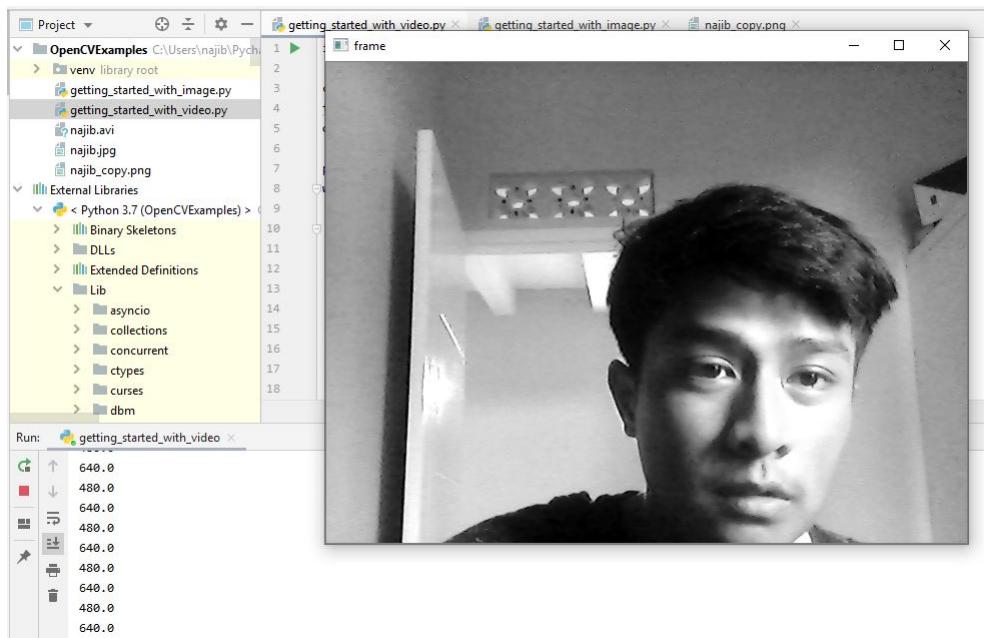
Maka akan ditampilkan secara berulang karna berada pada while dan pada bagian videocapture juga jika tidak di lakukan perulangan maka sekali muncul akan langsung keluar secara otomatis.

## 2.2.4 Menyimpan video

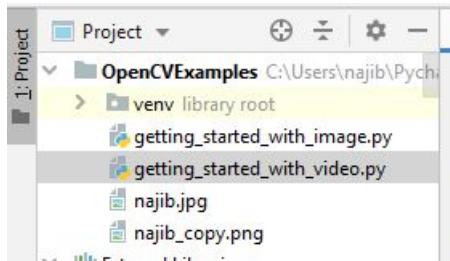
```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 fourcc = cv2.VideoWriter_fourcc(*'XVID')
6
7 out = cv2.VideoWriter('najib.avi', fourcc, 20.0, (640,480))
8
9 print(cap.isOpened())
10
11 while(cap.isOpened()):
12
13     ret, frame = cap.read()
14
15     if ret == True:
16
17         print(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
18
19         print(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
20
21         out.write(frame)
22
23         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
24
25         cv2.imshow('frame', gray)
26
27         if cv2.waitKey(1) & 0xFF == ord('q'):
28             break
29     else:
30         break
31
32 cap.release()
33
34 out.release()
35
36 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian buat variable baru dengan nama cap kemudian panggil VideoCapture(0) yang artinya menjalankan kamera leptop.
- iii. membuat variabel fourcc untuk merekam video yang dijalankan.
- iv. membuat variable out untuk menyimpan video dengan nama najib.avi dan frame berukuran 640,480.
- v. melakukan print apakah true atau false kamera leptop terbuka.
- vi. membuat while yaitu perulangan membuka frame
- vii. kemudian didalam perulangan tersebut terdapat frame yang membaca atau merekam video.
- viii. jika kamera true merekam maka akan melakukan perintah.
- ix. kita cukup print mengambil dari videocapture dan panggil CAP PROP FRAME WIDTH untuk mengetahui ukuran lebarnya dan

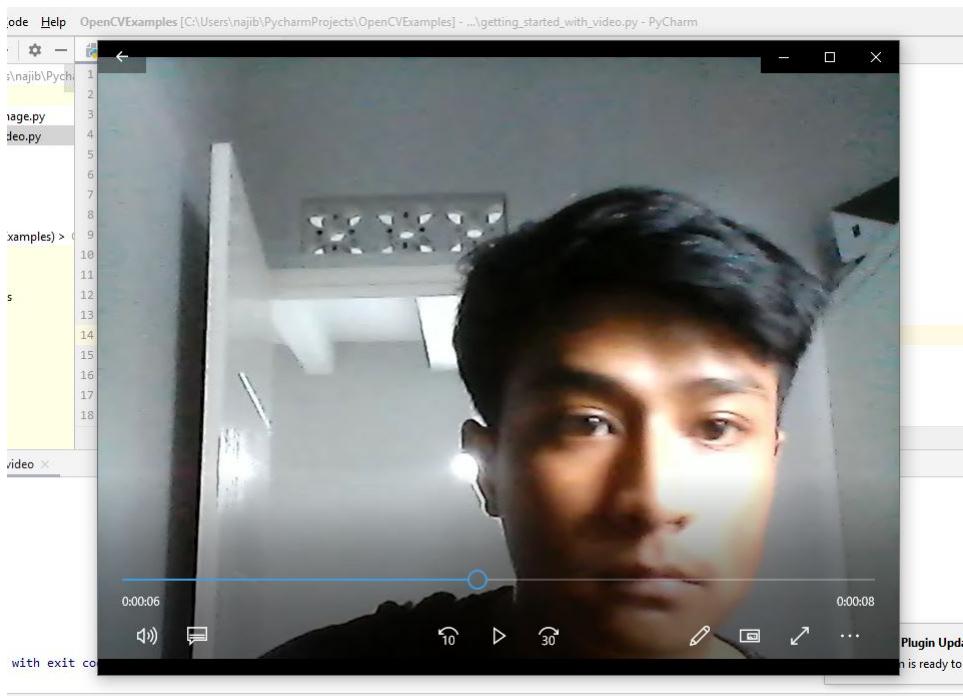
- x. kita cukup print mengambil dari videocapture dan panggil CAP PROP FRAME HEIGHT untuk ukuran tingginya
- xi. buat variable dengan nama gray karna kita mau berubah kontras warnanya menjadi hitam putih, kemudian panggil cvtColor didalam frame dengan warna abu abu.
- xii. kemudian buat frame dengan nama frame
- xiii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xiv. release untuk menutup videocapture
- xv. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.7** Setelah dijalankan file tersimpan



**Gambar 2.8** File Sebelum dijalankan



**Gambar 2.9** Video yang sudah di simpan

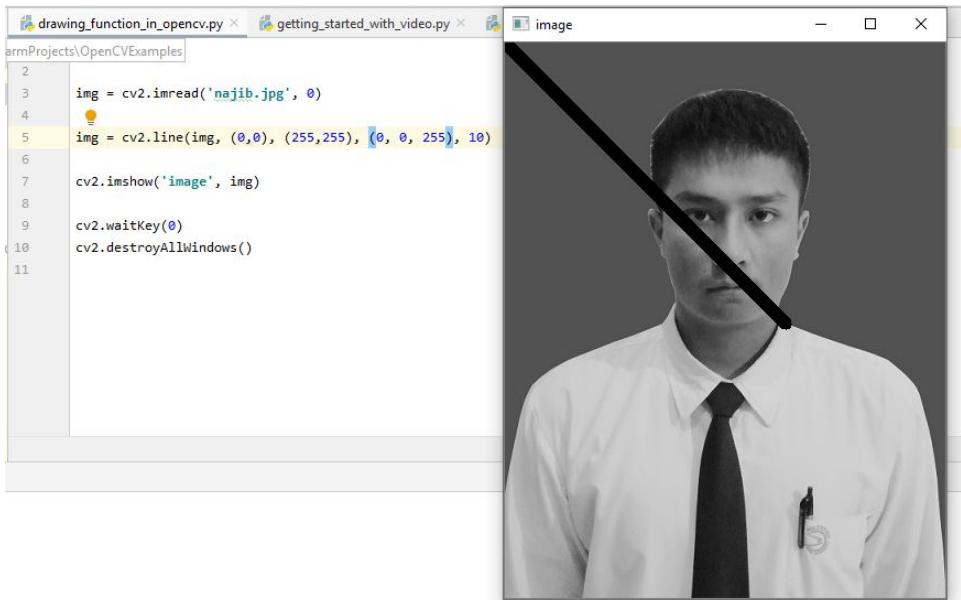
Video tersimpan langsung ke folder yang dituju, video dapat disesuaikan formatnya sesuai yang kita mau.

## 2.3 Menggambar Geometric Pada Foto

### 2.3.1 Membuat garis

```
1 import cv2  
2  
3 img = cv2.imread('najib.jpg', 0)  
4  
5 img = cv2.line(img, (0,0), (255,255), (0, 0, 255), 10)  
6  
7 cv2.imshow('image', img)  
8  
9 cv2.waitKey(0)  
10  
11 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradiasi warnanya, pada bagian ini menggunakan angka 0 yang artinya gambar berubah menjadi hitam putih.
- iii. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- iv. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- v. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- vi. destroyAllWindows digunakan untuk menutup frame.



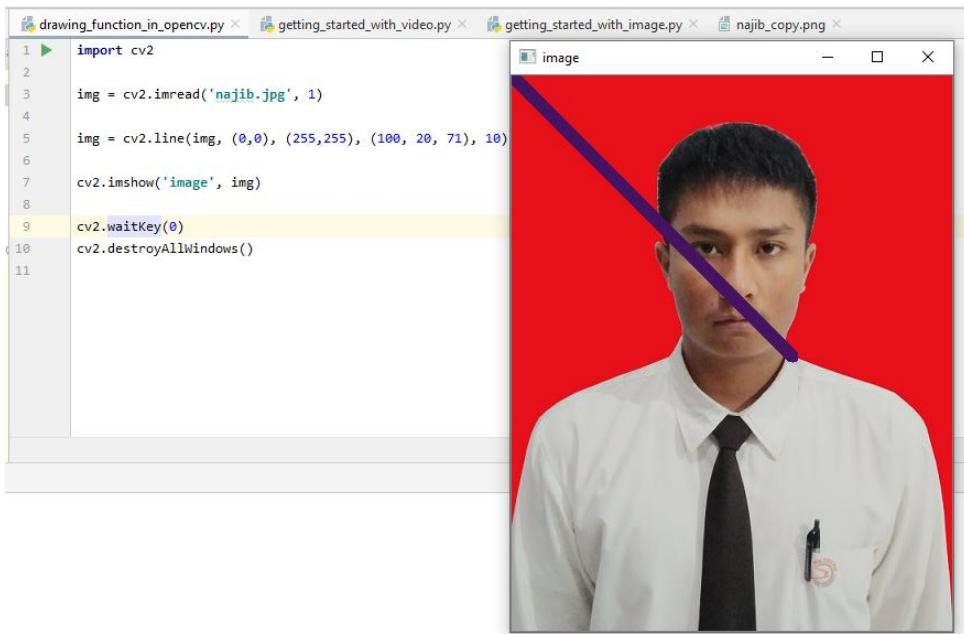
**Gambar 2.10** Membuat garis

garis bisa kita taruh dimana saja sesuai yang diinginkan kenapa garisnya menjulur dari pojok kiri atas ke tengah karna titik 0,0 berada di pojok kiri atas sedangkan titik 255,255 berada di tengah tengah gambar, gambar ini pun menjadi hitam putih karna di awal pada imread nya diberikan angka 0 yang membuat gambar berubah menjadi hitam putih.

### 2.3.2 Membuat warna warna pada garis

```
1 import cv2  
2  
3 img = cv2.imread('najib.jpg', 1)  
4  
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)  
6  
7 cv2.imshow('image', img)  
8  
9 cv2.waitKey(0)  
10  
11 cv2.destroyAllWindows()
```

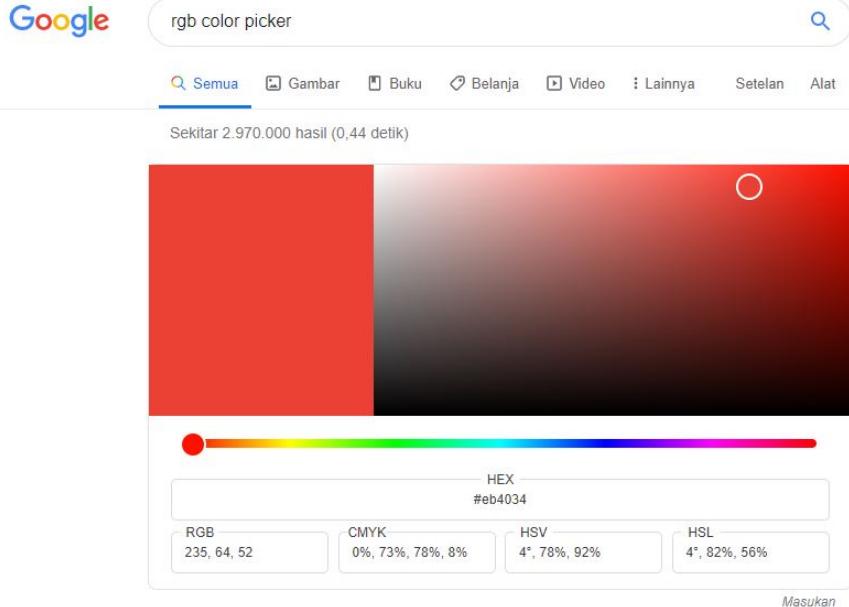
- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.
- iii. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- iv. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- v. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- vi. destroyAllWindows digunakan untuk menutup frame.



```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
6
7 cv2.imshow('image', img)
8
9 cv2.waitKey(0)
10 cv2.destroyAllWindows()
```

**Gambar 2.11** Membuat garis

garis bisa kita taruh dimana saja sesuai yang diinginkan kenapa garisnya menjulur dari pojok kiri atas ke tengah karna titik 0,0 berada di pojok kiri atas sedangkan titik 255,255 berada di tengah tengah gambar, gambar menjadi seperti aslinya karna pada imreadnya 1.



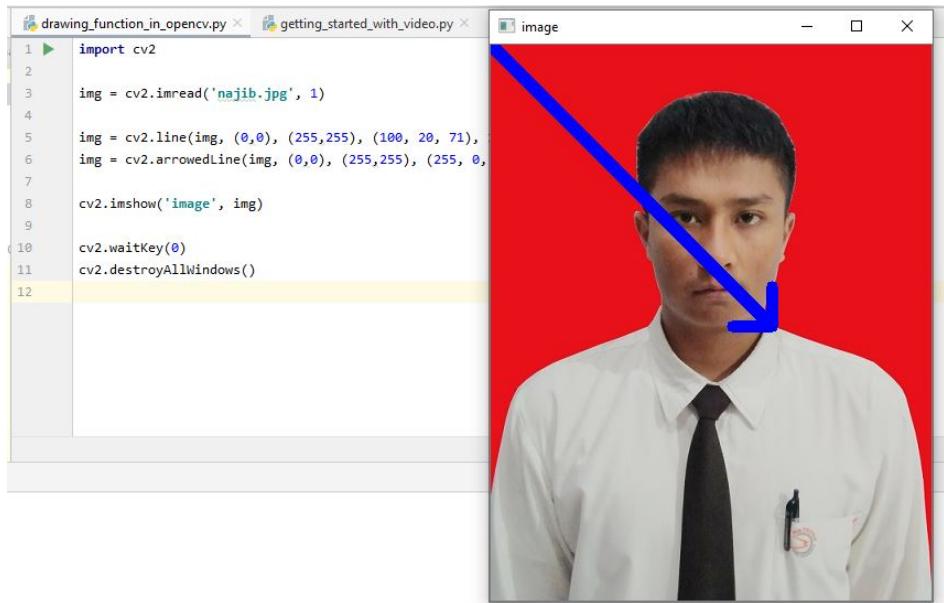
**Gambar 2.12** Nomor Nomor warna

jika kita ingin warna yang sesuai dengan keinginan kita, kita bisa langsung search di google seperti pada gambar maka akan ada nomor nomornya untuk setiap warna.

### 2.3.3 Membuat garis panah

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
6
7 img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0, 0), 10)
8
9 cv2.imshow('image', img)
10
11 cv2.waitKey(0)
12
13 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.
- iii. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- iv. kemudian buat garis panah menggunakan cv2.arrowedLine, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- v. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- vi. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- vii. destroyAllWindows digunakan untuk menutup frame.



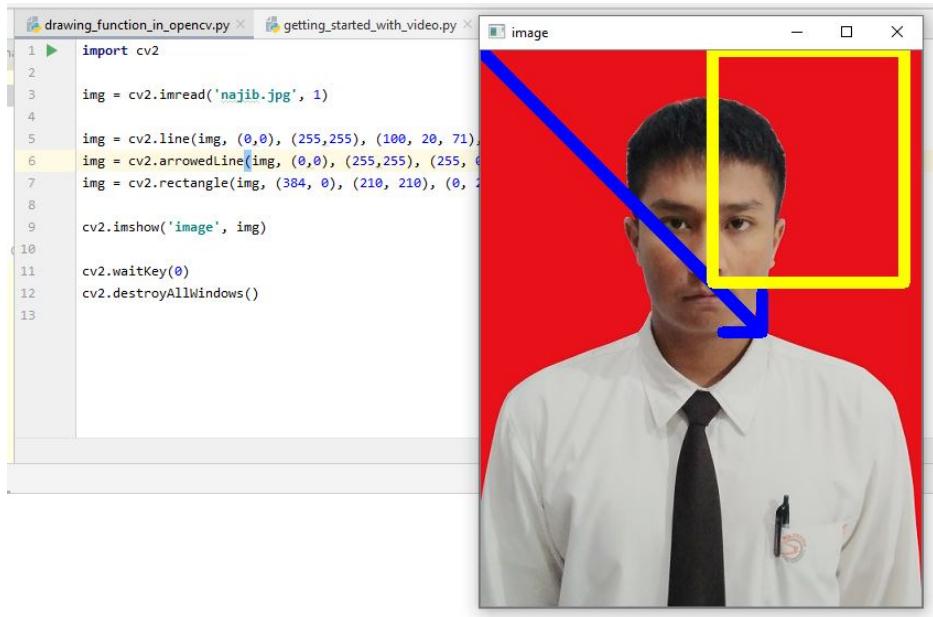
**Gambar 2.13** Membuat garis panah

Garis panah yang dibuat menumpuk dengan garis yang awal jadi yang di lihat seperti garis yang sebelumnya hilang padahal garis tersebut tertumpuk.

### 2.3.4 Membuat garis kotak

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
6
7 img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0, 0), 10)
8
9 img = cv2.rectangle(img, (384, 0), (210, 210), (0, 0, 255),
10   10)
11
12 cv2.imshow('image', img)
13
14 cv2.waitKey(0)
15
16 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradiasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.
- iii. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- iv. kemudian buat garis panah menggunakan cv2.arrowedLine, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- v. kemudian buat garis kotak menggunakan cv2.rectangle, 384,0 merupakan dimana titik awal garis tersebut dan 210,210 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- vi. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- vii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- viii. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.14** Membuat garis kotak

Garis kotak ini bisa kita atur mau warna yang bagaimana, ukuran yang bagaimana, dan posisi yang bagaimana sesuai yang di inginkan dan sesuai kebutuhannya.

### 2.3.5 Membuat kotak

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
6
7 img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0, 0), 10)
8
9 img = cv2.rectangle(img, (384, 0), (210, 210), (0, 255, 255),
-1)
10
11 cv2.imshow('image', img)
12
13 cv2.waitKey(0)
14
15 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradiasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.
- iii. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- iv. kemudian buat garis panah menggunakan cv2.arrowedLine, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- v. kemudian buat garis kotak menggunakan cv2.rectangle, 384,0 merupakan dimana titik awal garis tersebut dan 210,210 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah -1 yang membuat kotak tersisi full karna jika plus yang membesar adalah bagian luarnya jika minus yang membesar adalah bagian dalamnya jika minus maka akan full.
- vi. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- vii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- viii. destroyAllWindows digunakan untuk menutup frame.

The image shows a Python code editor with two tabs: 'drawing\_function\_in\_opencv.py' and 'getting\_started\_with\_video.py'. The code in 'drawing\_function\_in\_opencv.py' uses OpenCV's drawing functions to draw a blue line, an arrowed line, and a rectangle on a man's face. The resulting image window titled 'image' displays a man's face with these drawn shapes. A large blue arrow points from the code to the image.

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71)
6 img = cv2.arrowedLine(img, (0,0), (255,255), (255,
7 img = cv2.rectangle(img, (384, 0), (210, 210), (0,
8
9 cv2.imshow('image', img)
10
11 cv2.waitKey(0)
12 cv2.destroyAllWindows()
```

**Gambar 2.15** Membuat kotak

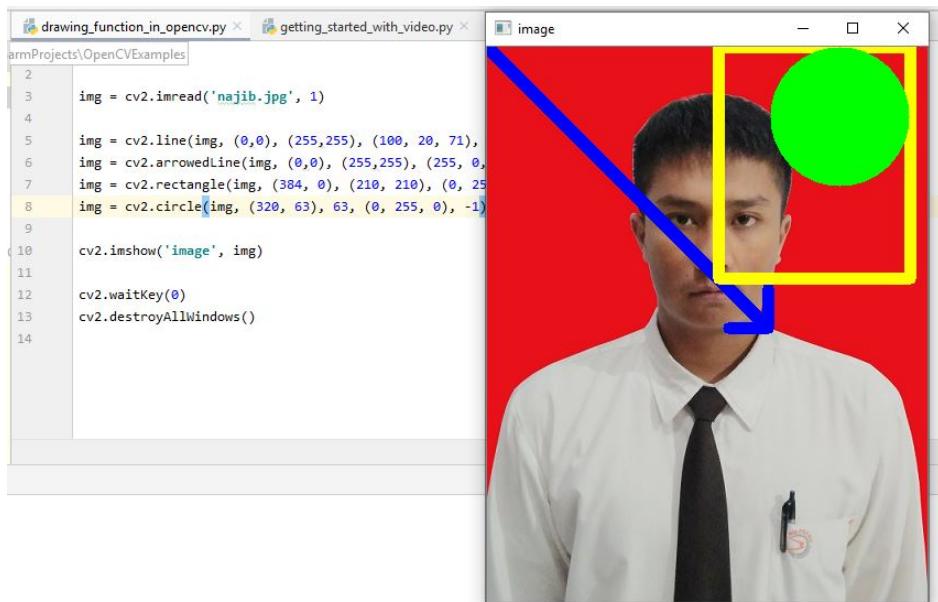
Garis kotak ini bisa kita atur mau warna yang bagaimana, ukuran yang bagaimana, dan posisi yang bagaimana sesuai yang di inginkan dan sesuai kebutuhannya.

### 2.3.6 Membuat garis Lingkaran

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
6
7 img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0, 0), 10)
8
9 img = cv2.rectangle(img, (384, 0), (210, 210), (0, 255, 255),
10   10)
11
12 img = cv2.circle(img, (320, 63), 63, (0, 255, 0), -1)
13
14 cv2.imshow('image', img)
15
16 cv2.waitKey(0)
17 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradiasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.
- iii. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- iv. kemudian buat garis panah menggunakan cv2.arrowedLine, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- v. kemudian buat garis kotak menggunakan cv2.rectangle, 384,0 merupakan dimana titik awal garis tersebut dan 210,210 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- vi. kemudian buat garis kotak menggunakan cv2.circle, 320,63 merupakan dimana titik awal garis tersebut dan 63 merupakan titik tengah, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah -1 yang membuat kotak terisi full karna jika plus yang membesar adalah bagian luarnya jika minus yang membesar adalah bagian dalamnya jika minus maka akan full.
- vii. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.

- viii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- ix. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.16** Membuat garis lingkaran

Garis Lingkaran ini bisa kita atur mau warna yang bagaimana, ukuran yang bagaimana, dan posisi yang bagaimana sesuai yang di inginkan dan sesuai kebutuhannya.

### 2.3.7 Membuat Text

```
1 import cv2
2
3 img = cv2.imread('najib.jpg', 1)
4
5 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
6
7 img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0, 0), 10)
8
9 img = cv2.rectangle(img, (384, 0), (210, 210), (0, 255, 255),
10   10)
11
12 img = cv2.circle(img, (320, 63), 63, (0, 255, 0), -1)
13
14 font = cv2.FONT_HERSHEY_SIMPLEX
15
16 img = cv2.putText(img, 'OpenCv', (10, 400), font, 3, (0, 255,
17   0), 10, cv2.LINE_AA)
18
19 cv2.imshow('image', img)
20
21 cv2.waitKey(0)
22
23 cv2.destroyAllWindows()
```

- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradiasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.
- iii. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- iv. kemudian buat garis panah menggunakan cv2.arrowedLine, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- v. kemudian buat garis kotak menggunakan cv2.rectangle, 384,0 merupakan dimana titik awal garis tersebut dan 210,210 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- vi. kemudian buat garis kotak menggunakan cv2.circle, 320,63 merupakan dimana titik awal garis tersebut dan 63 merupakan titik tengah, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah -1 yang membuat kotak terisi full karna jika

plus yang membesar adalah bagian luarnya jika minus yang membesar adalah bagian dalamnya jika minus maka akan full.

- vii. tentukan fontnya terlebih dahulu
- viii. kemudian gunakan putText atur sesuai seperti pada gambar.
- ix. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- x. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xi. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows a PyCharm interface with several open tabs. The active tab contains the following Python code:

```
1  import cv2
2
3  img = cv2.imread('najib.jpg', 1)
4
5  img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
6  img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0, 0))
7  img = cv2.rectangle(img, (384, 0), (210, 210), (0, 255, 255), 3)
8  img = cv2.circle(img, (320, 63), 63, (0, 255, 0), -1)
9  font = cv2.FONT_HERSHEY_SIMPLEX
10 img = cv2.putText(img, 'OpenCv', (10, 400), font, 3, (0, 255, 0))
11
12 cv2.imshow('image', img)
13
14 cv2.waitKey(0)
15 cv2.destroyAllWindows()
16
```

To the right of the code editor is a window titled "image" displaying a portrait of a man. The background is red. Overlaid on the image are several drawn elements: a blue arrow pointing downwards from the top left, a yellow rectangle around the man's head, a green circle inside the yellow rectangle, and the word "OpenCv" written in large green letters at the bottom left.

Gambar 2.17 Membuat Text

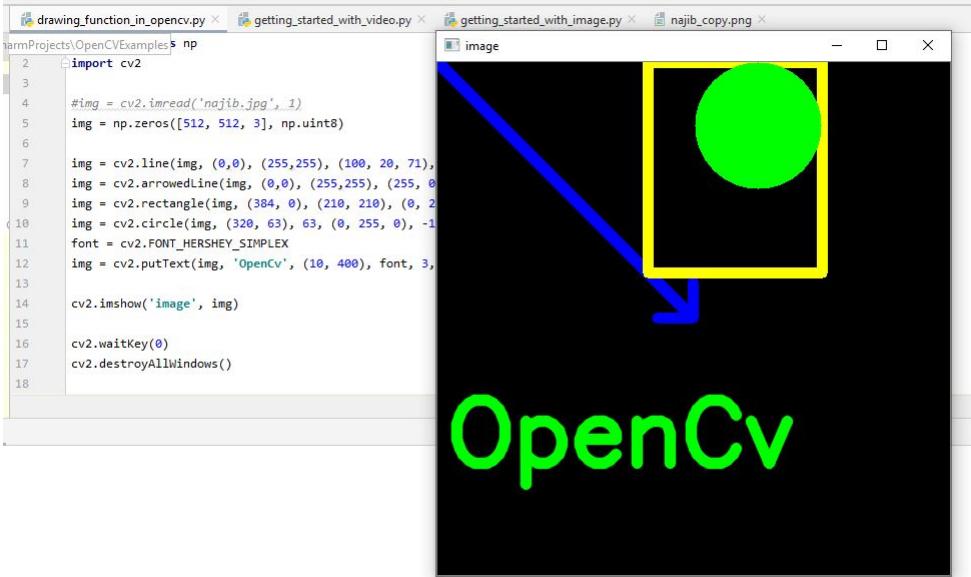
Text ini bisa kita buat sesuai kata kata yang di inginkan dan kata kata yang sesuai pada gambar, mau warna yang bagaimana, ukuran yang bagaimana, dan posisi yang bagaimana sesuai yang di inginkan dan sesuai kebutuhannya.

## 2.4 Frame Numpay

```
1 import numpy as np
2
3 import cv2
4
5 #img = cv2.imread('najib.jpg', 1)
6 img = np.zeros([512, 512, 3], np.uint8)
7
8 img = cv2.line(img, (0,0), (255,255), (100, 20, 71), 10)
9
10 img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0, 0), 10)
11
12 img = cv2.rectangle(img, (384, 0), (210, 210), (0, 255, 255),
13                      10)
14
15 font = cv2.FONT_HERSHEY_SIMPLEX
16
17 img = cv2.putText(img, 'OpenCV', (10, 400), font, 3, (0, 255,
18                      0), 10, cv2.LINE_AA)
19
20 cv2.imshow('image', img)
21
22 cv2.waitKey(0)
23
24 cv2.destroyAllWindows()
```

- i. lakukan import numpy as np
- ii. lakukan Import library open cv yaitu cv2
- iii. kemudian buat frame dari numpy yaitu zeros, kemudian ukuran fram dan warna dari fram, yang di buat adalah hitam.
- iv. kemudian buat garis menggunakan cv2.line, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- v. kemudian buat garis panah menggunakan cv2.arrowedLine, 0,0 merupakan dimana titik awal garis tersebut dan 255,255 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.
- vi. kemudian buat garis kotak menggunakan cv2.rectangle, 384,0 merupakan dimana titik awal garis tersebut dan 210,210 merupakan titik akhir dari garis tersebut, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah ketebalan dari garis yang dibuat.

- vii. kemudian buat garis kotak menggunakan cv2.circle, 320,63 merupakan dimana titik awal garis tersebut dan 63 merupakan titik tengah, kemudian selanjutnya adalah warna dari garis tersebut, dan yang terakhir adalah -1 yang membuat kotak terisi full karna jika plus yang membesar adalah bagian luarnya jika minus maka akan full.
- viii. tentukan fontnya terlebih dahulu
- ix. kemudian gunakan putText atur sesuai seperti pada gambar.
- x. kemudian buat frame untuk menampilkan gambar menggunakan imshow dengan nama frame image.
- xi. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xii. destroyAllWindows digunakan untuk menutup frame.



```

1  #!/usr/bin/python
2  import cv2
3
4  #img = cv2.imread('najib.jpg', 1)
5  img = np.zeros([512, 512, 3], np.uint8)
6
7  img = cv2.line(img, (0,0), (255,255), (100, 20, 71),
8  img = cv2.arrowedLine(img, (0,0), (255,255), (255, 0,
9  img = cv2.rectangle(img, (384, 0), (210, 210), (0, 2
10 img = cv2.circle(img, (320, 63), 63, (0, 255, 0), -1
11 font = cv2.FONT_HERSHEY_SIMPLEX
12 img = cv2.putText(img, 'OpenCv', (10, 400), font, 3,
13
14 cv2.imshow('image', img)
15
16 cv2.waitKey(0)
17 cv2.destroyAllWindows()
18

```

**Gambar 2.18** Membuat Frame Numpy

Frame ini dibuat menggunakan matriks yang ada pada library numpy, kita tidak perlu lagi menghitung berapa matriksnya kita cukup gunakan zeros dan ukuran yang di butuhkan.

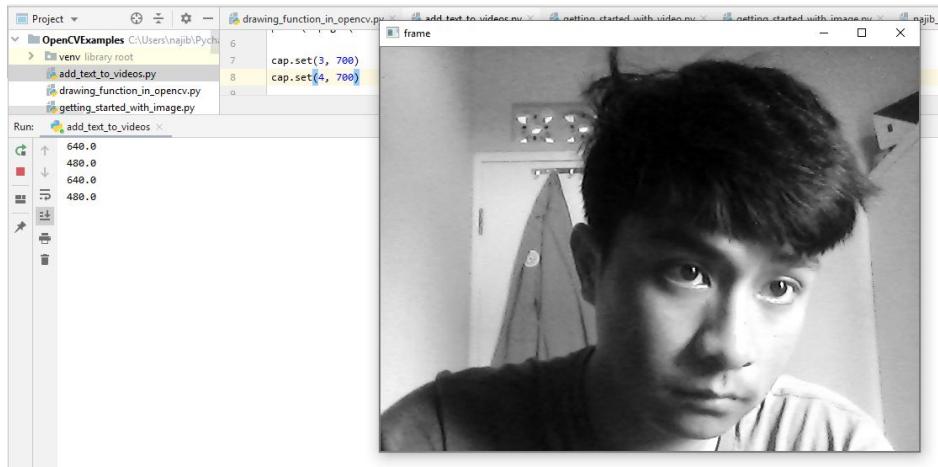
## 2.5 Ukuran Frame Menggunakan CAP PROP FRAME

### 2.5.1 Berubah Ukuran Frame

```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 print(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
6
7 print(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
8
9 cap.set(3, 700)
10
11 cap.set(4, 700)
12
13 print(cap.get(3))
14
15 print(cap.get(4))
16
17 while(cap.isOpened()):
18
19     ret, frame = cap.read()
20
21     if ret == True:
22
23         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
24
25         cv2.imshow('frame', gray)
26
27         if cv2.waitKey(1) & 0xFF == ord('q'):
28             break
29         else:
30             break
31
32 cap.release()
33
34 cv2.destroyAllWindows()
```

- i. import cv2
- ii. membuat variable cap untuk menghubungkan kamera leptop
- iii. print ukuran frame lebar dan tinggi menggunakan CAP PROP FRAME WIDTH dan CAP PROP FRAME HEIGHT
- iv. membuat ukuran frame baru, jika ukurannya tidak sesuai maka akan otomatis mengikuti ukutan sebelumnya.
- v. print untuk menampilkan ukuran frame saat ini
- vi. membuat while yaitu perulangan membuka frame
- vii. kemudian didalam perulangan tersebut terdapat frame yang membaca atau merekam video.
- viii. jika kamera true merekam maka akan melakukan perintah.

- ix. buat variable dengan nama gray karna kita mau berubah kontras warnanya menjadi hitam putih, kemudian panggil cvtColor didalam frame dengan warna abu abu.
- x. kemudian buat frame dengan nama frame
- xi. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xii. release untuk menutup videocapture
- xiii. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.19** Merubah ukuran frame

Pada gambar menunjukan yang di print tersebut tidak sesuai dengan yang di inginkan, karna jika kita merubah tidak sesuai dengan frame yang tertera maka akan mengikuti frame yang sebelumnya di gunakan, jadi jika ingin merubah harus sesuai.

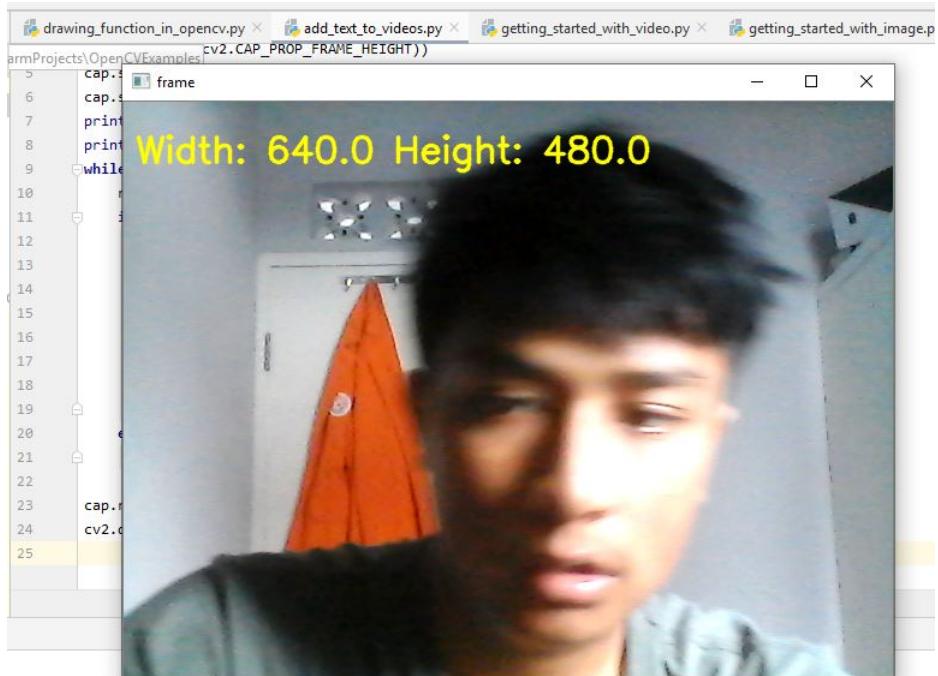
## 2.6 Video Text

### 2.6.1 Menampilkan Text pada Frame video

```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 print(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
6
7 print(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
8
9 cap.set(3, 700)
10
11 cap.set(4, 700)
12
13 print(cap.get(3))
14
15 print(cap.get(4))
16
17 while(cap.isOpened()):
18
19     ret, frame = cap.read()
20
21     if ret == True:
22
23         font = cv2.FONT_HERSHEY_SIMPLEX
24
25         text = 'Width: ' + str(cap.get(3)) + ' Height: ' + str(
26             cap.get(4))
27
28         frame = cv2.putText(frame, text, (10, 50), font, 1,
29             (0, 255, 255), 2, cv2.LINE_AA)
30
31         cv2.imshow('frame', frame)
32
33         if cv2.waitKey(1) & 0xFF == ord('q'):
34             break
35     else:
36         break
37
38 cap.release()
39
40 cv2.destroyAllWindows()
```

- i. import cv2
- ii. membuat variable cap untuk menghubungkan kamera leptop
- iii. print ukuran frame lebar dan tinggi menggunakan CAP PROP FRAME WIDTH dan CAP PROP FRAME HEIGHT
- iv. membuat ukuran frame baru, jika ukurannya tidak sesuai maka akan otomatis mengikuti ukutan sebelumnya.
- v. print untuk menampilkan ukuran frame saat ini
- vi. membuat while yaitu perulangan membuka frame

- vii. kemudian didalam perulangan tersebut terdapat frame yang membaca atau merekam video.
- viii. jika kamera true merekam maka akan melakukan perintah.
- ix. membuat font untuk tulisan pada gambar
- x. menampilkan tulisan ukuran frame
- xi. menampilkan tulisan pada frame
- xii. kemudian buat frame dengan nama frame
- xiii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xiv. release untuk menutup videocapture
- xv. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.20** Menampilkan Text pada Frame video

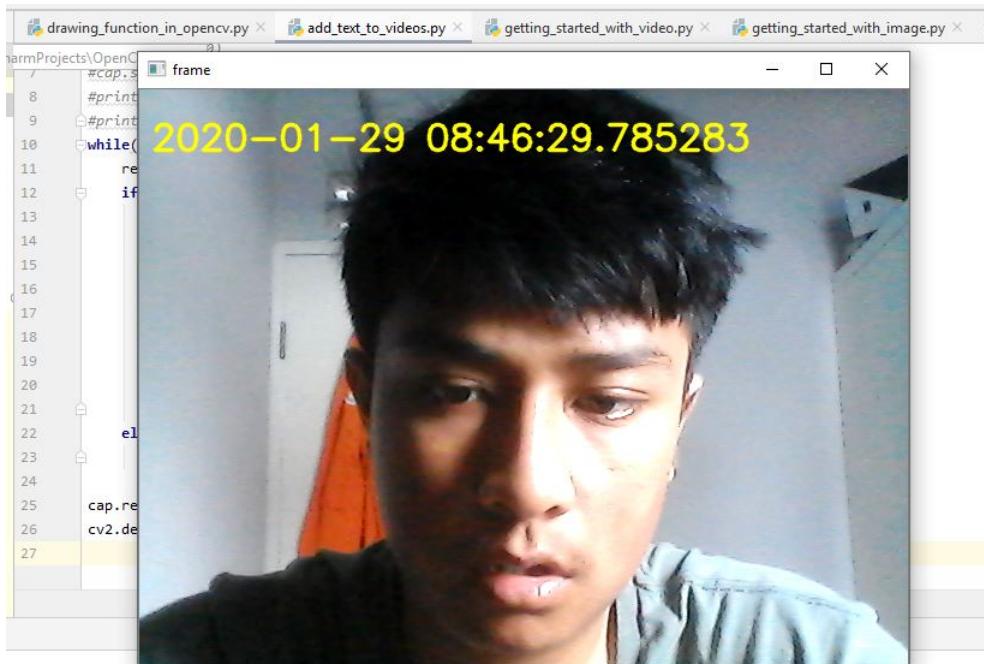
Pada gambar di tampilkan text dengan width ukuran lebarnya yang di panggil menggunakan print yang sebelumnya dan height tingginya, jadi code ini berguna untuk menampilkan ukuran frame yang sedang di gunakan.

## 2.6.2 Menampilkan waktu pada frame

```
1 import cv2
2
3 import datetime
4
5 cap = cv2.VideoCapture(0)
6
7 print(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
8
9 print(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
10
11 #cap.set(3, 700)
12 #cap.set(4, 700)
13 #print(cap.get(3))
14 #print(cap.get(4))
15
16 while(cap.isOpened()):
17
18     ret, frame = cap.read()
19
20     if ret == True:
21
22         font = cv2.FONT_HERSHEY_SIMPLEX
23
24         text = 'Width: ' + str(cap.get(3)) + ' Height: ' + str(
25             cap.get(4))
26
27         datet = str(datetime.datetime.now())
28
29         frame = cv2.putText(frame, datet, (10, 50), font, 1,
30             (0, 255, 255), 2, cv2.LINE_AA)
31
32         cv2.imshow('frame', frame)
33
34         if cv2.waitKey(1) & 0xFF == ord('q'):
35             break
36
37 else:
38     break
39
40 cap.release()
41
42 cv2.destroyAllWindows()
```

- i. import cv2
- ii. import date time
- iii. membuat variable cap untuk menghubungkan kamera leptop
- iv. print ukuran frame lebar dan tinggi menggunakan CAP PROP FRAME WIDTH dan CAP PROP FRAME HEIGHT
- v. membuat while yaitu perulangan membuka frame
- vi. kemudian didalam perulangan tersebut terdapat frame yang membaca atau merekam video.
- vii. jika kamera true merekam maka akan melakukan perintah.

- viii. membuat font untuk tulisan pada gambar
- ix. menampilkan tulisan ukuran frame
- x. menampilkan tulisan tanggal dan waktu pada frame
- xi. kemudian buat frame dengan nama frame
- xii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xiii. release untuk menutup videocapture
- xiv. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.21** Menampilkan waktu pada frame

Pada gambar di tampilkan text berwarna kuning yang menampilkan tanggal, bulan, tahun, jam hingga detik yang di ambil menggunakan library date.

## 2.7 Event Mouse Klik

### 2.7.1 Menampilkan Event

```
1 import numpy as np
2
3 import cv2
4
5 events = [i for i in dir(cv2) if 'EVENT' in i]
6
7 print(events)
```

- i. Import numpy
- ii. import cv2
- iii. menampilkan even event yang dapat digunakan untuk mouse klik
- iv. menampilkannya dengan print



The screenshot shows the PyCharm IDE interface. On the left, the project tree displays a folder named 'OpenCVExamples' containing several Python files. In the center, a code editor window titled 'Mouse\_even\_opencv\_python.py' shows the following code:

```
1 # Import numpy as np
2 # Import cv2
3
4 events = [i for i in dir(cv2) if 'EVENT' in i]
5 print(events)
```

Below the code editor, the 'Run' tool window is open, showing the command 'Mouse\_even\_opencv\_python'. The output pane at the bottom displays the executed code and its result:

```
[ 'EVENT_FLAG_ALTKEY', 'EVENT_FLAG_CTRLKEY', 'EVENT_FLAG_LBUTTON', 'EVENT_FLAG_RBUTTON', 'EVENT_FLAG_SHIFTKEY', 'EVENT_LBUTTONDOWN', 'EVENT_LBUTTONDOWN', 'EVENT_MBUTTONDOWN', 'EVENT_MBUTTONUP', 'EVENT_MOUSEWHEEL', 'EVENT_MOUSEMOVE', 'EVENT_RBUTTONDOWN', 'EVENT_RBUTTONDOWN', 'EVENT_RBUTTONUP']
```

At the very bottom of the output pane, it says 'Process finished with exit code 0'.

Gambar 2.22 Menampilkan Event

## 2.7.2 Event Mouse klik kiri

```
1 import numpy as np
2
3 import cv2
4
5 #events = [i for i in dir(cv2) if 'EVENT' in i]
6 #print(events)
7
8 def click_event(event, x, y, flags, param):
9
10     if event == cv2.EVENT_LBUTTONDOWN:
11
12         print(x, ' ', ' ', y)
13
14         font = cv2.FONT_HERSHEY_SIMPLEX
15
16         strXY = str(x) + ' ', ' ' + str(y)
17
18         cv2.putText(img, strXY, (x, y), font, .5, (255, 255,
19         0), 2)
20
21         cv2.imshow('image', img)
22
23 img = np.zeros((512, 512, 3), np.uint8)
24
25 cv2.imshow('image', img)
26
27 cv2.setMouseCallback('image', click_event)
28
29
30 cv2.destroyAllWindows()
```

- i. Import numpy
- ii. import cv2
- iii. buat def dengan nama click event
- iv. jika mouse mengklik kiri maka akan melakukan sesuatu
- v. pada frame akan menampilkan posisi pada frame yang di klik
- vi. membuat frame dengan ukuran 512 512 dengan warna hitam
- vii. menampilkan frame dengan nama image
- viii. memanggil fungsi klik pada mouse
- ix. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- x. destroyAllWindows digunakan untuk menutup frame.

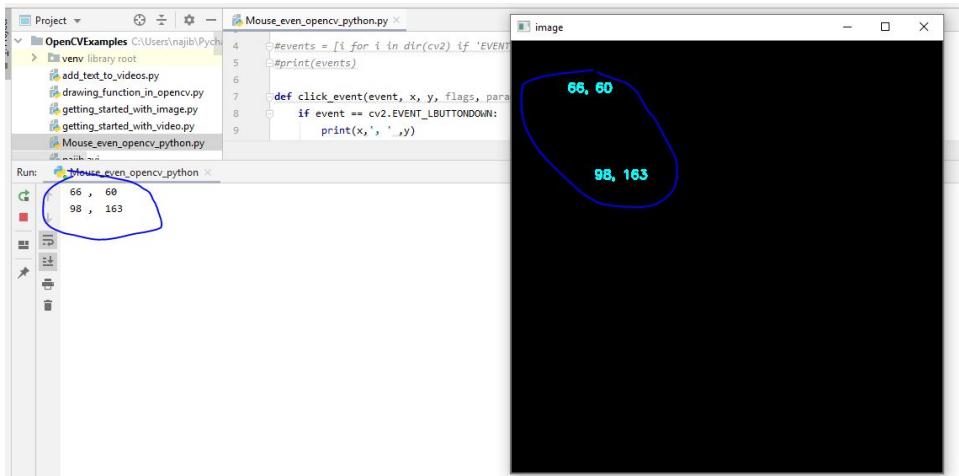
The screenshot shows a Python code editor and a separate image viewer window. The code in the editor is as follows:

```
7 def click_event(event, x, y, flags, param):
8     if event == cv2.EVENT_LBUTTONDOWN:
9         print(x, ' ', y)
10        font = cv2.FONT_HERSHEY_SIMPLEX
11        strXY = str(x) + ' ' + str(y)
12        cv2.putText(img, strXY, (x, y), font, .5,
13                    cv2.imshow('image', img)
14
15 img = np.zeros((512, 512, 3), np.uint8)
16 cv2.imshow('image', img)
17
18 cv2.setMouseCallback('image', click_event)
19
20 cv2.waitKey(0)
21 cv2.destroyAllWindows()
22
```

The image viewer window is titled "image" and is currently empty.

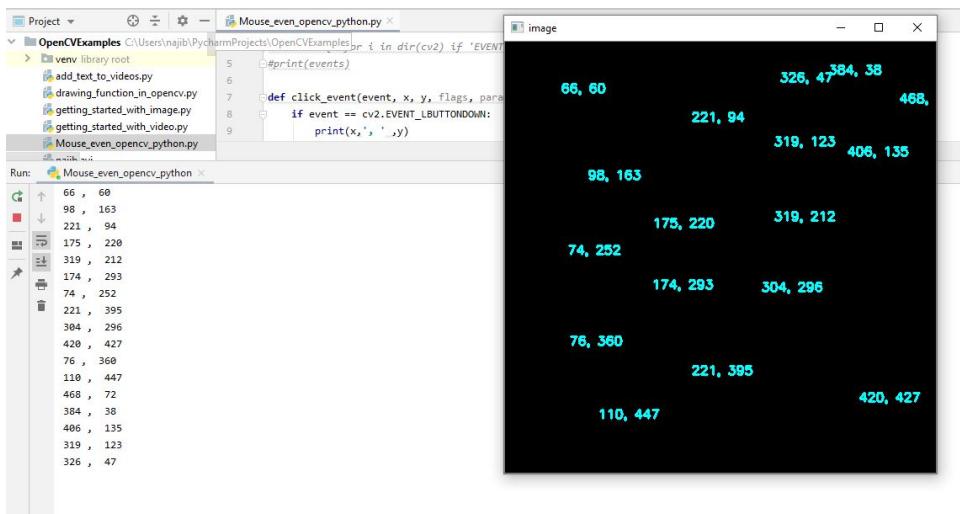
**Gambar 2.23** Event Mouse klik kiri

Tampilan sebelum mouse kiri mengklik pada frame di bagian mana pun, gambar frame hitam karan frame tidak mengambil gambar dari mana pun



**Gambar 2.24** Event Mouse klik kiri

Ketika frame di klik kiri pada bagian frame maka akan menampilkan seperti pada gambar, frame akan menampilkan lokasi x dan y dimana mouse meng-klik, dan akan di print pada serial monitoe.



Gambar 2.25 Event Mouse klik kiri

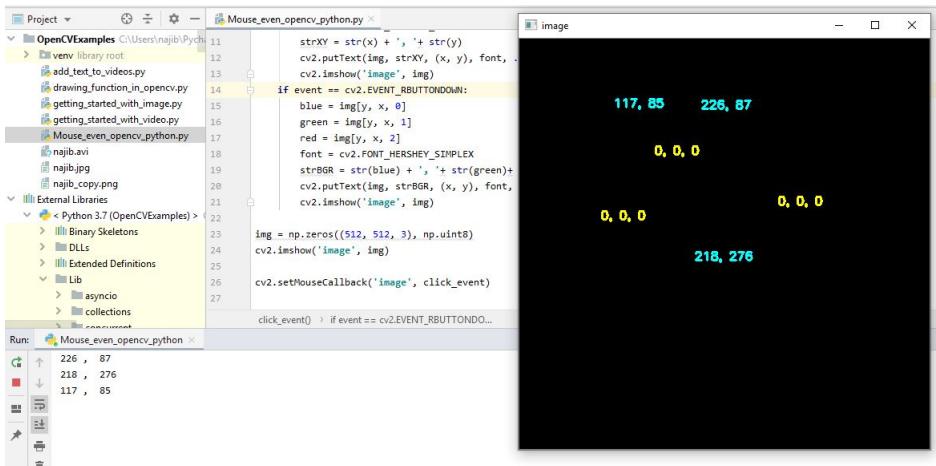
Jika terus di klik maka frame akan terus menampilkan lokasi x dan y yang di minta oleh mouse, jika kita mengklik pada lokasi yang telah di klik pun frame akan tetap terus menampilkan sesuai permintaan mouse.

### 2.7.3 Event Mouse klik kiri dan kanan

```
1 import numpy as np
2
3 import cv2
4
5 #events = [i for i in dir(cv2) if 'EVENT' in i]
6 #print(events)
7
8 def click_event(event, x, y, flags, param):
9
10     if event == cv2.EVENT_LBUTTONDOWN:
11
12         print(x, ' ', ' ',y)
13
14         font = cv2.FONT_HERSHEY_SIMPLEX
15
16         strXY = str(x) + ' , ' + str(y)
17
18         cv2.putText(img, strXY, (x, y), font, .5, (255, 255,
19         0), 2)
20
21         cv2.imshow('image', img)
22
23     if event == cv2.EVENT_RBUTTONDOWN:
24
25         blue = img[y, x, 0]
26
27         green = img[y, x, 1]
28
29         red = img[y, x, 2]
30
31         font = cv2.FONT_HERSHEY_SIMPLEX
32
33         strBGR = str(blue) + ' , ' + str(green)+ ' , ' + str(red)
34
35         cv2.putText(img, strBGR, (x, y), font, .5, (0, 255,
36         255), 2)
37
38         cv2.imshow('image', img)
39
40 cv2.imshow('image', img)
41
42 cv2.setMouseCallback('image', click_event)
43
44 cv2.waitKey(0)
45
46 cv2.destroyAllWindows()
```

- i. Import numpy
- ii. import cv2
- iii. buat def dengan nama click event
- iv. jika mouse mengklik kiri maka akan melakukan sesuatu

- v. pada frame akan menampilkan posisi pada frame yang di klik
- vi. dan jika mouse mengklik kanan
- vii. maka frame akan menampilkan nomor warna yang ada pada frame yang di klik tersebut, terdapat 3 nomor karna menggunakan konsep bgr yaitu blue green dan red.
- viii. membuat frame dengan ukuran 512 512 dengan warna hitam
- ix. menampilkan frame dengan nama image
- x. memanggil fungsi klik pada mouse
- xi. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xii. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.26** Event Mouse klik kiri dan kanan

Pada gambar menampilkan koordinat x dan y untuk yang berwarna biru yang sudah di jelaskan pada section sebelumnya, sedangkan yang berwarna kuning berguna untuk menampilkan nomor warna pada frame karan pada frame berwarna hitam dan nomor wara untuk hitam adalah 0,0,0 maka frame menampilkan seperti pada gambar.

## 2.7.4 Event Mouse klik kiri dan kanan pada gambar yang dipanggil

```

1 import numpy as np
2
3 import cv2
4
5 #events = [i for i in dir(cv2) if 'EVENT' in i]
6 #print(events)
7
8 def click_event(event, x, y, flags, param):
9
10    if event == cv2.EVENT_LBUTTONDOWN:
11
12        print(x, ' ', ' ', y)
13
14        font = cv2.FONT_HERSHEY_SIMPLEX
15
16        strXY = str(x) + ' , ' + str(y)
17
18        cv2.putText(img, strXY, (x, y), font, .5, (255, 255,
19        0), 2)
20
21        cv2.imshow('image', img)
22
23    if event == cv2.EVENT_RBUTTONDOWN:
24
25        blue = img[y, x, 0]
26
27        green = img[y, x, 1]
28
29        red = img[y, x, 2]
30
31        font = cv2.FONT_HERSHEY_SIMPLEX
32
33        strBGR = str(blue) + ' , ' + str(green) + ' , ' + str(red)
34
35        cv2.putText(img, strBGR, (x, y), font, .5, (0, 255,
36        255), 2)
37
38        cv2.imshow('image', img)
39
40 #img = np.zeros((512, 512, 3), np.uint8)
41 img = cv2.imread('najib.jpg')
42
43 cv2.imshow('image', img)
44
45 cv2.setMouseCallback('image', click_event)
46
47 cv2.waitKey(0)
48
49 cv2.destroyAllWindows()

```

- i. Import numpy
- ii. import cv2

- iii. buat def dengan nama click event
- iv. jika mouse mengklik kiri maka akan melakukan sesuatu
- v. pada frame akan menampilkan posisi pada frame yang di klik
- vi. dan jika mouse mengklik kanan
- vii. maka frame akan menampilkan nomor warna yang ada pada frame yang di klik tersebut, terdapat 3 nomor karna menggunakan konsep bgr yaitu blue green dan red.
- viii. memanggil gambar untuk di taruh pada frame yang telah di buat
- ix. menampilkan frame dengan nama image
- x. memanggil fungsi klik pada mouse
- xi. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xii. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows the PyCharm IDE interface. On the left, the Project panel displays a file structure under 'OpenCVExamples' containing various Python scripts and image files. The central editor window shows the code for 'Mouse\_even\_opencv\_python.py'. The code uses OpenCV's mouse callback function to print coordinates and text to an image frame. The rightmost window shows the resulting image where mouse events are highlighted with colored text overlays.

```

15     blue = img[y, x, 0]
16     green = img[y, x, 1]
17     red = img[y, x, 2]
18
19     font = cv2.FONT_HERSHEY_SIMPLEX
20     strBGR = str(blue) + ', ' + str(green)
21     cv2.putText(img, strBGR, (x, y), font, 1)
22     cv2.imshow('image', img)
23
24     #img = np.zeros((512, 512, 3), np.uint8)
25     img = cv2.imread('najib.jpg')
26     cv2.imshow('image', img)
27
28     cv2.setMouseCallback('image', click_event)
29
30     cv2.waitKey(0)
31     cv2.destroyAllWindows()

```

Run: Mouse\_even\_opencv\_python

Output:

- 311 , 113
- 283 , 157
- 201 , 148
- 127 , 148
- 183 , 39
- 78 , 132
- 27 , 135
- 57 , 356
- 230 , 415

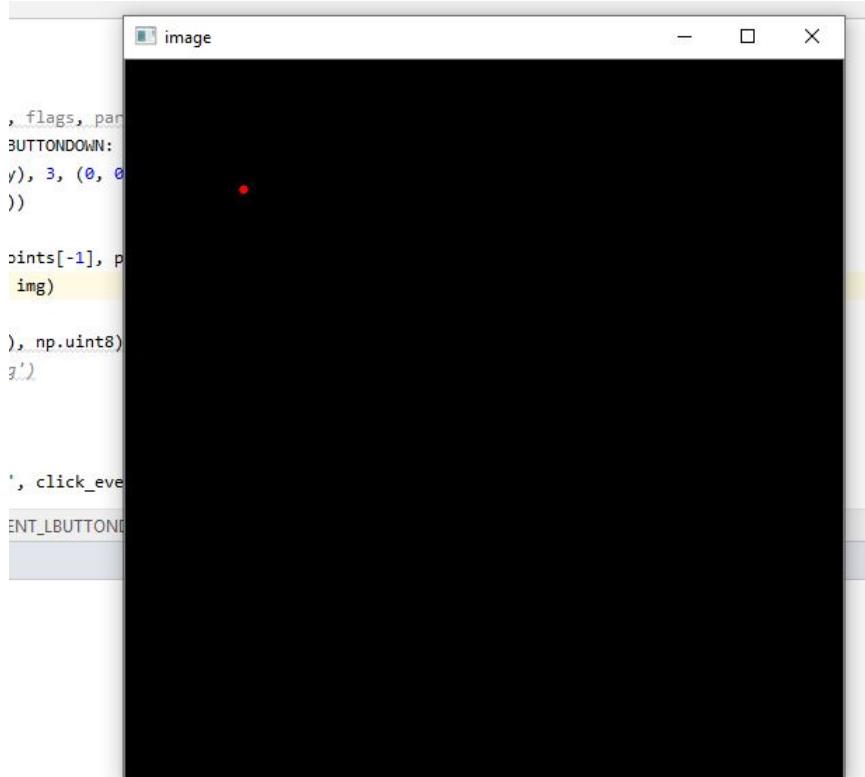
**Gambar 2.27** Event Mouse klik kiri dan kanan pada gambar yang dipanggil

Pada gambar menampilkan koordinat x dan y untuk yang berwarna biru yang sudah di jelaskan pada section sebelumnya, sedangkan yang berwarna kuning berguna untuk menampilkan nomor warna pada frame, frame akan menampilkan nomor warna sesuai warna yang ada pada mouse yang di klik.

## 2.7.5 Event Mouse klik kiri membuat titik dan garis

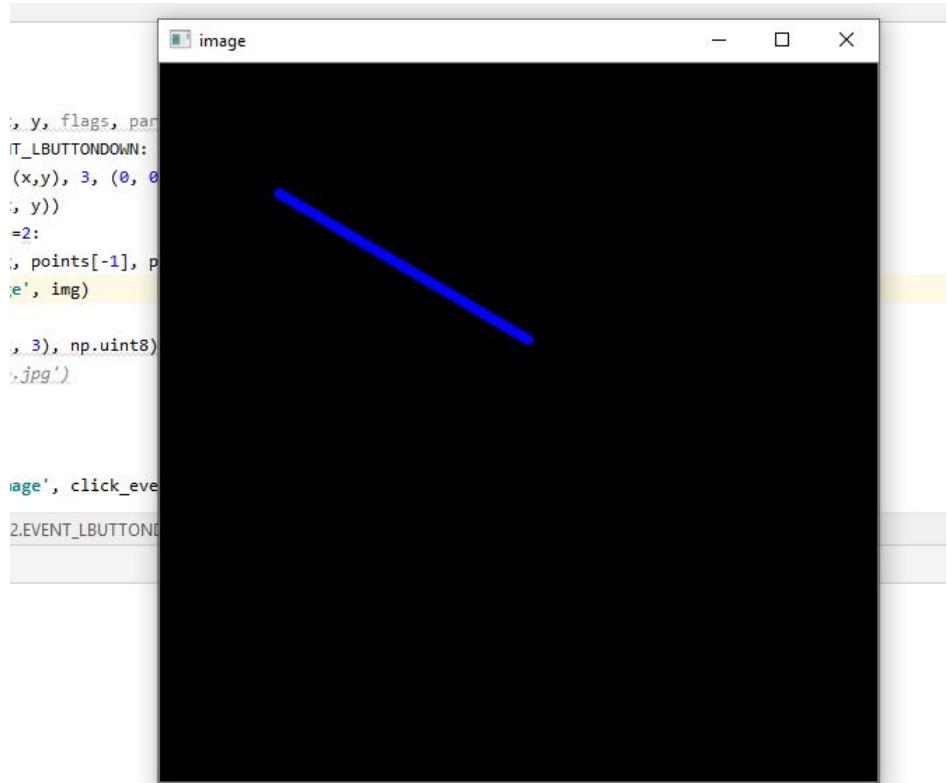
```
1 import numpy as np
2
3 import cv2
4
5 def click_event(event, x, y, flags, param):
6
7     if event == cv2.EVENT_LBUTTONDOWN:
8
9         cv2.circle(img, (x,y), 3, (0, 0, 255), -1)
10
11     points.append((x, y))
12
13 if len(points) >=2:
14
15     cv2.line(img, points[-1], points[-2], [255, 0, 0], 5)
16
17 cv2.imshow('image', img)
18
19 img = np.zeros((512, 512, 3), np.uint8)
20
21 #img = cv2.imread('najib.jpg')
22 cv2.imshow('image', img)
23
24 points = []
25
26 cv2.setMouseCallback('image', click_event)
27
28 cv2.waitKey(0)
29
30 cv2.destroyAllWindows()
```

- i. Import numpy
- ii. import cv2
- iii. buat def dengan nama click event
- iv. jika mouse mengklik kiri maka akan melakukan sesuatu
- v. pada frame akan membuat sebuah titik berwarna merah sesuai lokasi mengklik frame
- vi. jika titik tersebut lebih dari sama dengan 2 maka setiap titik yang terakhir akan terhubung satu sama lain.
- vii. menampilkan frame dengan nama image
- viii. gambar pada frame berwarna hitam
- ix. menampilkan frame kembali
- x. point menghilang setelah berubah menjadi garis atau line
- xi. memanggil fungsi klik pada mouse
- xii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xiii. destroyAllWindows digunakan untuk menutup frame.



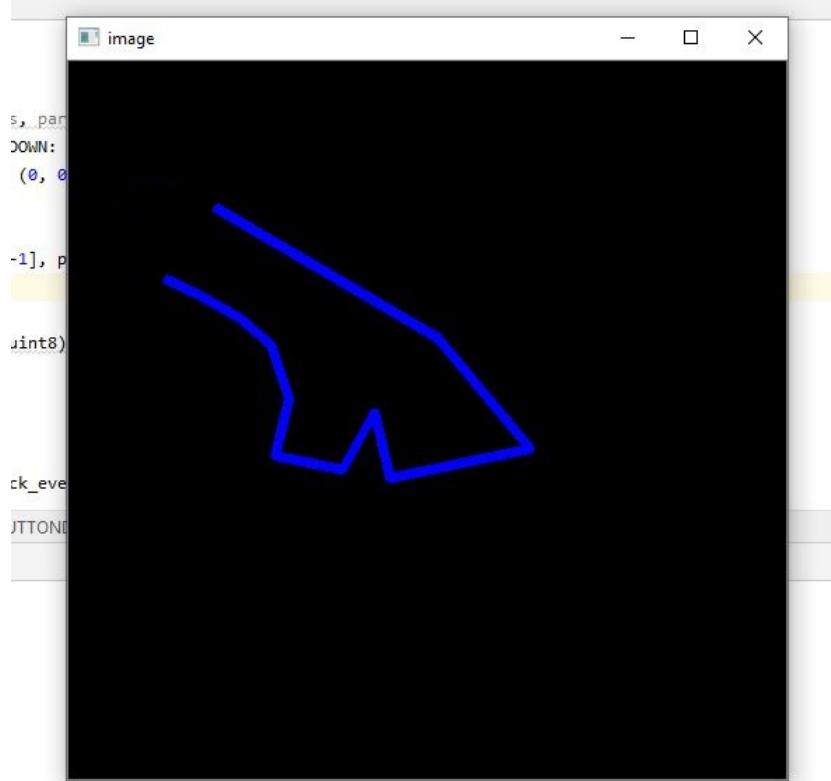
**Gambar 2.28** Event Mouse klik kiri membuat titik dan garis

Ketika frame mengklik pada kiri pada frame maka frame akan menampilkan titik berwarna merah sesuai lokasi mouse mengklik frame.



**Gambar 2.29** Event Mouse klik kiri membuat titik dan garis

Ketika mouse mengklik kembali pada frame di lokasi yang berbeda maka titik merah yang pertama akan menghilang dan pada titik yang pertama akan menghubungkan ke titik yang ke dua menjadi sebuah garis.



**Gambar 2.30** Event Mouse klik kiri membuat titik dan garis

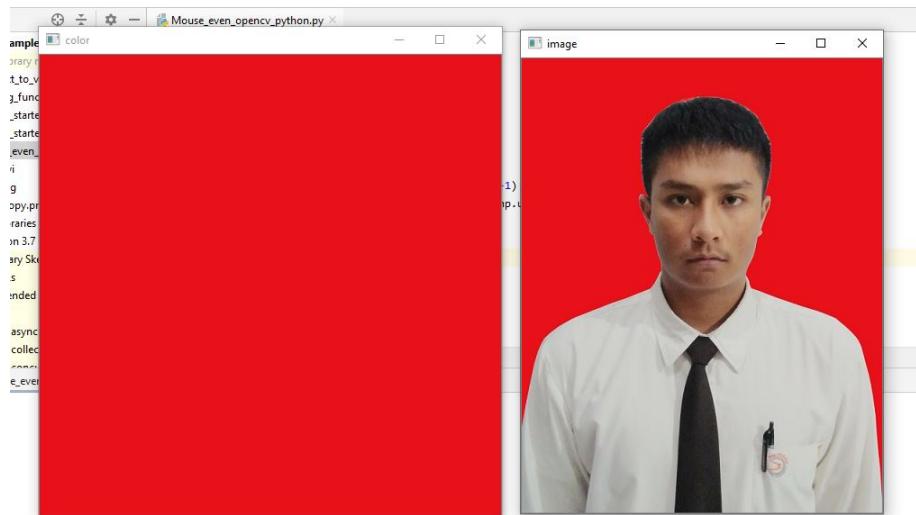
Jadi jika titik merah lebih dari sama dengan 2 maka titik merah akan menghilang dan akan menghubungkan dari titik satu ke titik yang kedua jika mouse mengklik kembali menjadi titik yang ke tiga maka titik yang terakhir akan menghubungkan ke titik yang baru.

## 2.7.6 Membuat frame warna sesuai klik

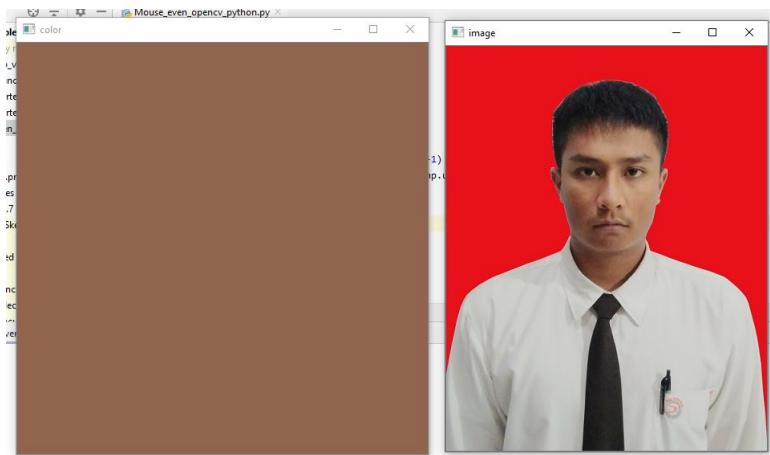
```
1 import numpy as np
2
3 import cv2
4
5 def click_event(event, x, y, flags, param):
6
7     if event == cv2.EVENT_LBUTTONDOWN:
8
9         blue = img[y, x, 0]
10
11         green = img[y, x, 1]
12
13         red = img[y, x, 2]
14
15         cv2.circle(img, (x,y), 3, (0, 0, 255), -1)
16
17     mycolorImage = np.zeros((512, 512, 3), np.uint8)
18
19     mycolorImage[:] = [blue, green, red]
20
21     cv2.imshow('color', mycolorImage)
22
23 #img = np.zeros((512, 512, 3), np.uint8)
24 img = cv2.imread('najib.jpg')
25
26 cv2.imshow('image', img)
27
28 points = []
29
30 cv2.setMouseCallback('image', click_event)
31
32 cv2.waitKey(0)
33
34 cv2.destroyAllWindows()
```

- i. Import numpy
- ii. import cv2
- iii. buat def dengan nama click event
- iv. jika mouse mengklik kiri maka akan melakukan sesuatu
- v. warna biru di deklarasikan dengan 0
- vi. warna hijau di deklarasikan dengan 1
- vii. warna merah di deklarasikan dengan 2
- viii. membuat lingkaran kecil berwarna merah
- ix. membuat warna sesuai lokasi frame yang di klik harus sesuai dengan warna yang di klik
- x. memanggil gambar
- xi. menampilkan frame kembali
- xii. point menghilang setelah berubah menjadi garis atau line
- xiii. memanggil fungsi klik pada mouse

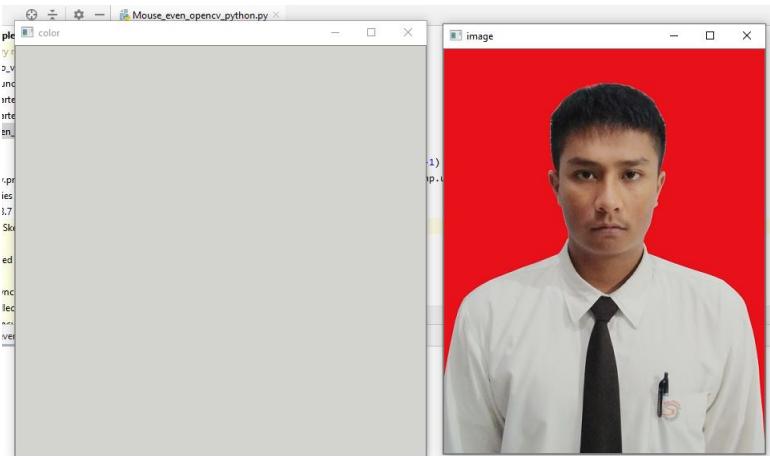
- xiv. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xv. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.31** Membuat frame warna sesuai klik



**Gambar 2.32** Membuat frame warna sesuai klik



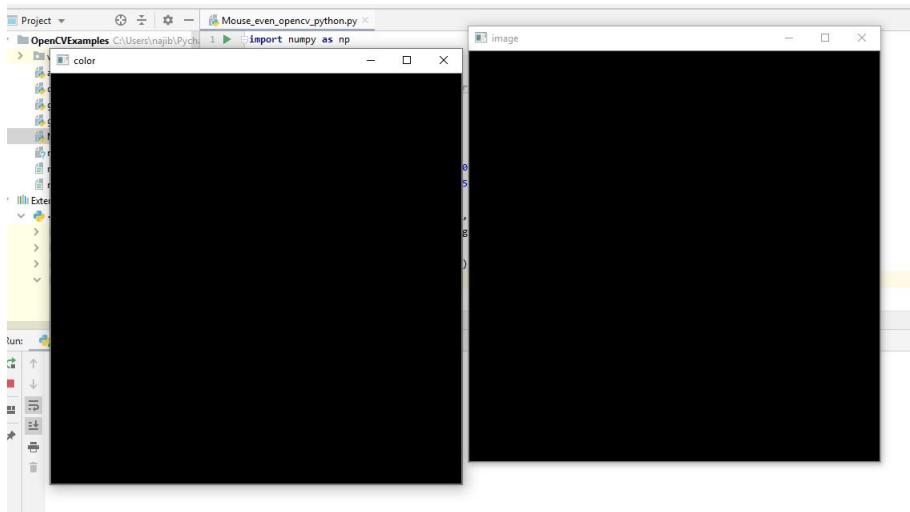
**Gambar 2.33** Membuat frame warna sesuai klik

### 2.7.7 Membuat frame warna sesuai klik 2

```
1 import numpy as np
2
3 import cv2
4
5 def click_event(event, x, y, flags, param):
6
7     if event == cv2.EVENT_LBUTTONDOWN:
8
9         blue = img[y, x, 0]
10
11         green = img[y, x, 1]
12
13         red = img[y, x, 2]
14
15         cv2.circle(img, (x,y), 3, (0, 0, 255), -1)
16
17     mycolorImage = np.zeros((512, 512, 3), np.uint8)
18
19     mycolorImage[:] = [blue, green, red]
20
21     cv2.imshow('color', mycolorImage)
22
23 img = np.zeros((512, 512, 3), np.uint8)
24 #img = cv2.imread('najib.jpg')
25
26 cv2.imshow('image', img)
27
28 points = []
29
30 cv2.setMouseCallback('image', click_event)
31
32 cv2.waitKey(0)
33
34 cv2.destroyAllWindows()
```

- i. Import numpy
- ii. import cv2
- iii. buat def dengan nama click event
- iv. jika mouse mengklik kiri maka akan melakukan sesuatu
- v. warna biru di deklarasikan dengan 0
- vi. warna hijau di deklarasikan dengan 1
- vii. warna merah di deklarasikan dengan 2
- viii. membuat lingkaran kecil berwarna merah
- ix. membuat warna sesuai lokasi frame yang di klik harus sesuai dengan warna yang di klik
- x. menggunakan gambar hitam
- xi. menampilkan frame kembali
- xii. point menghilang setelah berubah menjadi garis atau line
- xiii. memanggil fungsi klik pada mouse

- xiv. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xv. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.34** Membuat frame warna sesuai klik 2

## 2.7.8 Menampilkan Shape, size, dan dtype

```
1 import numpy as np
2
3 import cv2
4
5 img = cv2.imread('messi5.jpg')
6
7 print(img.shape)
8
9 print(img.size)
10
11 print(img.dtype)
12
13 b,g,r = cv2.split(img)
14
15 img = cv2.merge((b,g,r))
16
17 cv2.imshow('image', img)
18
19 cv2.waitKey(0)
20
21 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv2
- iii. mengambil gambar dari file
- iv. menampilkan ukuran lebar dan tinggi frame
- v. menampilkan ukuran dari frame
- vi. menampilkan jenis gambar yang ada pada frame
- vii. membagi tiap warna pada gambar
- viii. menampilkan frame dengan nama image
- ix. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- x. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows a PyCharm interface with two tabs open: 'basic\_operations\_on\_image.py' and 'Mouse\_even\_opencv\_python.py'. The 'basic\_operations\_on\_image.py' tab contains code for reading an image and displaying it. The 'Mouse\_even\_opencv\_python.py' tab contains code for reading an image, splitting it into channels, merging them back together, and displaying it. A blue oval highlights the run history in the bottom left of the interface.

```
basic_operations_on_image.py
1  import numpy as np
2  import cv2
3
4  img = cv2.imread('messi5.jpg')
5
6  print(img.shape)
7  print(img.size)
8  print(img.dtype)
9
10 b,g,r = cv2.split(img)
11
12 img = cv2.merge((b,g,r))
13
14 cv2.imshow('image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

Run: basic_operations_on_image
   ↑
   (342, 548, 3)
562248
↓
   uint8
```

```
Mouse_even_opencv_python.py
1  import numpy as np
2  import cv2
3
4  img = cv2.imread('messi5.jpg')
5
6  print(img.shape)
7  print(img.size)
8  print(img.dtype)
9
10 b,g,r = cv2.split(img)
11
12 img = cv2.merge((b,g,r))
13
14 cv2.imshow('image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

The right side of the interface shows a window titled 'image' displaying a soccer player (Lionel Messi) in action, wearing a Barcelona jersey, kicking a yellow ball. The image is displayed on a green grass field with a blurred background.

Gambar 2.35 Menampilkan Shape, size, dan dtype

## 2.7.9 Mengcopy gambar di dalam satu frame

```
1 import numpy as np
2
3 import cv2
4
5 img = cv2.imread('messi5.jpg')
6
7 print(img.shape)
8
9 print(img.size)
10
11 print(img.dtype)
12
13 b,g,r = cv2.split(img)
14
15 img = cv2.merge((b,g,r))
16
17 bola = img[280:340, 330:390]
18
19 img[273:333, 100:160] = bola
20
21 cv2.imshow('image', img)
22
23 cv2.waitKey(0)
24
25 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv2
- iii. mengambil gambar dari file
- iv. menampilkan ukuran lebar dan tinggi frame
- v. menampilkan ukuran dari frame
- vi. menampilkan jenis gambar yang ada pada frame
- vii. membagi tiap warna pada gambar
- viii. membuat variabel baru untuk mengcopy gambar yang terletak pada titik titik yang ditentukan
- ix. menempatkan gambar yang telah di copy ke titik baru yang ditentukan
- x. menampilkan frame dengan nama image
- xi. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xii. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows a PyCharm interface with two tabs: 'basic\_operations\_on\_image.py' and 'image'. The code in 'basic\_operations\_on\_image.py' reads a file named 'messi5.jpg', prints its shape, size, and dtype, splits it into b, g, r channels, merges them back, and then extracts a region of interest (ROI) from the bottom-left of the image. The resulting ROI is stored in the variable 'bola'. The 'image' tab shows a window titled 'image' displaying a soccer player in a red and blue jersey kicking a yellow and blue ball. The PyCharm status bar at the bottom indicates 'WEEKmedia'.

```
basic_operations_on_image.py
1 import cv2
2
3
4 img = cv2.imread('messi5.jpg')
5
6 print(img.shape)
7 print(img.size)
8 print(img.dtype)
9 b,g,r = cv2.split(img)
10 img = cv2.merge((b,g,r))
11
12 bola = img[280:340, 330:390]
13 img[273:333, 100:160] = bola
14
```

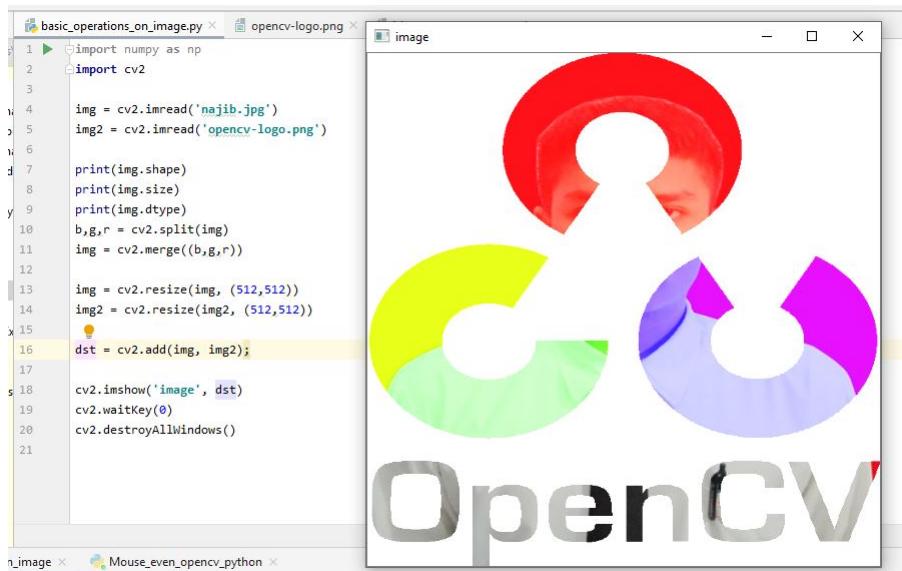
**Gambar 2.36** Mengcopy gambar di dalam satu frame

## 2.8 Menyatukan gambar dalam satu frame

### 2.8.1 Menyatukan 2 gambar dalam 1 frame

```
1 import numpy as np
2
3 import cv2
4
5 img = cv2.imread('najib.jpg')
6
7 img2 = cv2.imread('opencv-logo.png')
8
9 print(img.shape)
10
11 print(img.size)
12
13 print(img.dtype)
14
15 b,g,r = cv2.split(img)
16
17 img = cv2.merge((b,g,r))
18
19 img = cv2.resize(img, (512,512))
20
21 img2 = cv2.resize(img2, (512,512))
22
23 dst = cv2.add(img, img2);
24
25 cv2.imshow('image', dst)
26
27 cv2.waitKey(0)
28
29 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv2
- iii. mengambil gambar dari file
- iv. mengambil gambar kedua untuk di satukan dalam satu frame
- v. menampilkan ukuran lebar dan tinggi frame
- vi. menampilkan ukuran dari frame
- vii. menampilkan jenis gambar yang ada pada frame
- viii. membagi tiap warna pada gambar
- ix. mengubah ukuran gambar yang ke1 harus sama satu sama lain
- x. mengubah ukuran gambar yang ke2 harus sama satu sama lain
- xi. menggabungkan gambar menggunakan add dan sebutkan kedua gambar
- xii. menampilkan frame dengan nama image dan panggil variable dst
- xiii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xiv. destroyAllWindows digunakan untuk menutup frame.



```
basic_operations_on_image.py
1 import numpy as np
2 import cv2
3
4 img = cv2.imread('najib.jpg')
5 img2 = cv2.imread('opencv-logo.png')
6
7 print(img.shape)
8 print(img.size)
9 print(img.dtype)
10 b,g,r = cv2.split(img)
11 img = cv2.merge((b,g,r))
12
13 img = cv2.resize(img, (512,512))
14 img2 = cv2.resize(img2, (512,512))
15
16 dst = cv2.add(img, img2);
17
18 cv2.imshow('image', dst)
19 cv2.waitKey(0)
20 cv2.destroyAllWindows()
```

**Gambar 2.37** Menyatukan 2 gambar dalam 1 frame

## 2.8.2 Menggabungkan 2 gambar dengan kontras transparan

```
1 import numpy as np
2
3 import cv2
4
5 img = cv2.imread('najib.jpg')
6
7 img2 = cv2.imread('opencv-logo.png')
8
9 print(img.shape)
10
11 print(img.size)
12
13 print(img.dtype)
14
15 b,g,r = cv2.split(img)
16
17 img = cv2.merge((b,g,r))
18
19 img = cv2.resize(img, (512,512))
20
21 img2 = cv2.resize(img2, (512,512))
22
23 #dst = cv2.add(img, img2);
24 dst = cv2.addWeighted(img, .5, img2, .5, 0);
25
26 cv2.imshow('image', dst)
27
28 cv2.waitKey(0)
29
30 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv2
- iii. mengambil gambar dari file
- iv. mengambil gambar kedua untuk di satukan dalam satu frame
- v. menampilkan ukuran lebar dan tinggi frame
- vi. menampilkan ukuran dari frame
- vii. menampilkan jenis gambar yang ada pada frame
- viii. membagi tiap warna pada gambar
- ix. mengubah ukuran gambar yang ke1 harus sama satu sama lain
- x. mengubah ukuran gambar yang ke2 harus sama satu sama lain
- xi. untuk kontras ransparan gunakan addWeighted dan tentukan kontrasnya sesuai gambar
- xii. menampilkan frame dengan nama image dan panggil variable dst
- xiii. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xiv. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows a Python code editor and a resulting image window. The code in the editor is:

```
1 import numpy as np
2 import cv2
3
4 img = cv2.imread('najib.jpg')
5 img2 = cv2.imread('opencv-logo.png')
6
7 print(img.shape)
8 print(img.size)
9 print(img.dtype)
10 b,g,r = cv2.split(img)
11 img = cv2.merge((b,g,r))
12
13 img = cv2.resize(img, (512,512))
14 img2 = cv2.resize(img2, (512,512))
15
16 dst = cv2.add(img, img2);
17 dst = cv2.addWeighted(img, .5, img2, .5, 0);
18
19 cv2.imshow('image', dst)
20 cv2.waitKey(0)
21 cv2.destroyAllWindows()
22
```

The resulting image window titled "image" displays a composite image where a man's face from a photo is overlaid onto the OpenCV logo, with the two images having different contrast levels.

**Gambar 2.38** Menggabungkan 2 gambar dengan kontras transparan

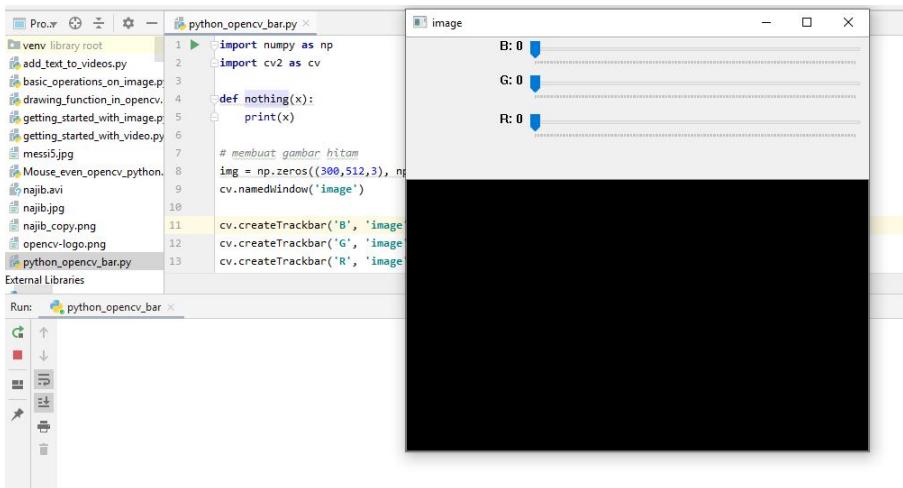
## 2.9 Trackbar

### 2.9.1 Membuat Trackbar

```
1 import numpy as np
2
3 import cv2 as cv
4
5 def nothing(x):
6     print(x)
7
8 # membuat gambar hitam
9
10 img = np.zeros((300,512,3), np.uint8)
11
12 cv.namedWindow('image')
13
14 cv.createTrackbar('B', 'image', 0, 255, nothing)
15
16 cv.createTrackbar('G', 'image', 0, 255, nothing)
17
18 cv.createTrackbar('R', 'image', 0, 255, nothing)
19
20 while(1):
21
22     cv.imshow('image', img)
23
24     k = cv.waitKey(1) & 0xFF
25
26     if k == 27:
27
28         break
29
30 cv.destroyAllWindows()
```

- i. import numpy
- ii. import cv
- iii. membuat variable untuk menampilkan
- iv. membuat gambar hitam menggunakan zeros dan ukuran frame yang di inginkan
- v. buat jendela dengan nama image
- vi. membuat Trackbar B digunakan untuk kontras warna biru dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- vii. membuat Trackbar G digunakan untuk kontras warna hijau dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- viii. membuat Trackbar R digunakan untuk kontras warna merah dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser

- ix. membuat pengulangan
- x. menampilkan frame dengan isi frame img yang telah di buat
- xi. membuat waitKey jika sama dengan 27 maka akan berakhir
- xii. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.39** Membuat Trackbar

The screenshot shows a PyCharm interface with two windows. The left window is a code editor for `python_opencv_bar.py`, displaying the following Python code:

```
1 import numpy as np
2 import cv2 as cv
3
4 def nothing(x):
5     print(x)
6
7     # membuat gambar hitam
8     img = np.zeros((300,512,3), np.uint8)
9     cv.namedWindow('image')
10
11    cv.createTrackbar('B', 'image', 68, 255, nothing)
12    cv.createTrackbar('G', 'image', 80, 255, nothing)
13    cv.createTrackbar('R', 'image', 202, 255, nothing)
```

The right window is titled "image" and displays a black image. Three trackbars are visible at the top of the window, labeled B: 68, G: 80, and R: 202, each with a blue slider.

Gambar 2.40 Membuat Trackbar

## 2.9.2 Membuat Trackbar dengan fungsi warna

```
1 import numpy as np
2
3 import cv2 as cv
4
5 def nothing(x):
6
7     print(x)
8
9 # membuat gambar hitam
10
11 img = np.zeros((300,512,3), np.uint8)
12
13 cv.namedWindow('image')
14
15 cv.createTrackbar('B', 'image', 0, 255, nothing)
16
17 cv.createTrackbar('G', 'image', 0, 255, nothing)
18
19 cv.createTrackbar('R', 'image', 0, 255, nothing)
20
21 while(1):
22
23     cv.imshow('image', img)
24
25     k = cv.waitKey(1) & 0xFF
26
27     if k == 27:
28
29         break
30
31     b = cv.getTrackbarPos('B', 'image')
32
33     g = cv.getTrackbarPos('G', 'image')
34
35     r = cv.getTrackbarPos('R', 'image')
36
37     img[:] = [b, g, r]
38
39 cv.destroyAllWindows()
```

- i. import numpy
- ii. import cv
- iii. membuat variable untuk menampilkan
- iv. membuat gambar hitam menggunakan zeros dan ukuran frame yang di inginkan
- v. buat jendela dengan nama image
- vi. membuat Trackbar B digunakan untuk kontras warna biru dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser

- vii. membuat Trackbar G digunakan untuk kontras warna hijau dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang geser
  - viii. membuat Trackbar R digunakan untuk kontras warna merah dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang geser
  - ix. membuat pengulangan
    - x. menampilkan frame dengan isi frame img yang telah dibuat
    - xi. membuat waitKey jika sama dengan 27 maka akan berakhir
    - xii. untuk mengisi Trackbar sesuai warna kontras warna yang ditentukan kita gunakan getTrackbarPos dan definisikan di awal sesuai warnanya biru hijau dan merah
  - xiii. kemudian definisikan kalau variable bgr tersebut adalah warna biru hijau dan merah
  - xiv. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows the PyCharm IDE interface with the following details:

- Project Bar:** Shows the project structure with files like `add_text_to_videos.py`, `basic_operations_on_image.p`, `drawing_function_in_opencv.p`, `getting_started_with_image.p`, `getting_started_with_video.py`, `messi5.jpg`, `Mouse_event_opencv_python.p`, `najib.avi`, `najib.jpg`, `najib_copy.png`, `opencv-logo.png`, and `python_opencv_bar.py`.
- Code Editor:** The file `python_opencv_bar.py` is open, displaying Python code for creating a window with three trackbars (B, G, R) and updating an image based on their values.
- Run Bar:** Shows the command `python python_opencv_bar`.
- Toolbars and Status Bar:** Standard PyCharm toolbar icons and status bar information.

```
# membuat gambar hitam
img = np.zeros((300,512,3), np.uint8)
cv.namedWindow("image")

cv.createTrackbar('B', 'image', 0, 255, nothing)
cv.createTrackbar('G', 'image', 0, 255, nothing)
cv.createTrackbar('R', 'image', 0, 255, nothing)

while(1):
    cv.imshow('image', img)
    k = cv.waitKey(1) & 0xFF
    if k == 27:
        break

    b = cv.getTrackbarPos('B', 'image')
    g = cv.getTrackbarPos('G', 'image')
    r = cv.getTrackbarPos('R', 'image')

    img[:, :] = [b, g, r]

cv.destroyAllWindows()
```

**Gambar 2.41** Membuat Trackbar dengan fungsi warna

The screenshot shows the PyCharm IDE interface. On the left, the project structure and file browser are visible, with the file `python_opencv_bar.py` selected. The main editor window displays the Python code for creating a color selection trackbar. To the right, a preview window titled "image" shows a grayscale image with three trackbars for B (Blue), G (Green), and R (Red) color channels. The B trackbar is at 98, G is at 101, and R is at 154. The bottom part of the image shows the terminal output with the command `python python_opencv_bar.py` and the resulting numerical values 107, 105, 104, 103, and 102.

```
# membuat gambar hitam
img = np.zeros((300,512,3), np.uint8)
cv.namedWindow('image')

cv.createTrackbar('B', 'image', 0, 255, nothing)
cv.createTrackbar('G', 'image', 0, 255, nothing)
cv.createTrackbar('R', 'image', 0, 255, nothing)

while(1):
    cv.imshow('image', img)
    k = cv.waitKey(1) & 0xFF
    if k == 27:
        break

    b = cv.getTrackbarPos('B', 'image')
    g = cv.getTrackbarPos('G', 'image')
    r = cv.getTrackbarPos('R', 'image')

    img[:] = [b, g, r]

cv.destroyAllWindows()
while(1)
```

Run python\_opencv\_bar

107  
106  
105  
104  
103  
102

Gambar 2.42 Membuat Trackbar dengan fungsi warna

### 2.9.3 Membuat Trackbar ON dan OFF

```
1 import numpy as np
2
3 import cv2 as cv
4
5 def nothing(x):
6
7     print(x)
8
9 # membuat gambar hitam
10
11 img = np.zeros((300,512,3), np.uint8)
12
13 cv.namedWindow('image')
14
15 cv.createTrackbar('B', 'image', 0, 255, nothing)
16
17 cv.createTrackbar('G', 'image', 0, 255, nothing)
18
19 cv.createTrackbar('R', 'image', 0, 255, nothing)
20
21 switch = '0 : OFF\n 1 : ON'
22
23 cv.createTrackbar(switch, 'image', 0, 1, nothing)
24
25 while(1):
26
27     cv.imshow('image', img)
28
29     k = cv.waitKey(1) & 0xFF
30
31     if k == 27:
32         break
33
34     b = cv.getTrackbarPos('B', 'image')
35
36     g = cv.getTrackbarPos('G', 'image')
37
38     r = cv.getTrackbarPos('R', 'image')
39
40     s = cv.getTrackbarPos(switch, 'image')
41
42     if s == 0:
43
44         img[:] = 0
45
46     else:
47
48         img[:] = [b, g, r]
49
50 cv.destroyAllWindows()
```

- i. import numpy
- ii. import cv
- iii. membuat variable untuk menampilkan

- iv. membuat gambar hitam menggunakan zeros dan ukuran frame yang di inginkan
- v. buat jendela dengan nama image
- vi. membuat Trackbar B digunakan untuk kontras warna biru dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- vii. membuat Trackbar G digunakan untuk kontras warna hijau dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- viii. membuat Trackbar R digunakan untuk kontras warna merah dari 0 sampai 255, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- ix. membuat tampilan on dan off
- x. membuat Trackbar on dan off dari 0 sampai 1, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- xi. membuat pengulangan
- xii. menampilkan frame dengan isi frame img yang telah di buat
- xiii. membuat waitKey jika sama dengan 27 maka akan berakhir
- xiv. untuk mengisi Trackbar sesuai warna kontras warna yang di tentukan kita gunakan getTrackbarPos dan definisikan di awal sesuai warnanya biru hijau dan merah
- xv. kemudian jika keadaan of maka tidak akan ada kontras warna jika kursor biru hijau dan merah di gerakan, dan jika kursor sudah on maka warna akan menyesuaikan
- xvi. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows the PyCharm IDE interface. On the left, the project structure is visible with files like `python_opencv_bar.py`, `messi5.jpg`, and `najib.avi`. The code editor displays the script `python_opencv_bar.py`:

```
1 import numpy as np
2 import cv2 as cv
3
4 def nothing(x):
5     print(x)
6
7 # membuat gambar hitam
8 img = np.zeros((300,512,3), np.uint8)
9 cv.namedWindow('image')
10
11 cv.createTrackbar('B', 'image', 0, 255, nothing)
12 cv.createTrackbar('G', 'image', 0, 255, nothing)
13 cv.createTrackbar('R', 'image', 0, 255, nothing)
14
15 switch = '0 : OFF\n 1 : ON'
16 cv.createTrackbar(switch, 'image', 0, 1, nothing)
17
18 while(1):
19     cv.imshow('image', img)
20     k = cv.waitKey(1) & 0xFF
21     if k == 27:
```

To the right of the code editor is a window titled "image" containing a black image. Three trackbars are present at the top of the window, labeled B: 120, G: 170, and R: 66. Below the trackbars is a status bar showing "0 : OFF...ON: 0".

Gambar 2.43 Membuat Trackbar ON dan OFF

The screenshot shows a PyCharm interface with a code editor and a preview window.

**Code Editor:**

```
python_opencv_bar.py
1 import numpy as np
2 import cv2 as cv
3
4 def nothing(x):
5     print(x)
6
7 # membuat gambar hitam
8 img = np.zeros((300,512,3), np.uint8)
9 cv.namedWindow('image')
10
11 cv.createTrackbar('B', 'image', 0, 255, nothing)
12 cv.createTrackbar('G', 'image', 0, 255, nothing)
13 cv.createTrackbar('R', 'image', 0, 255, nothing)
14
15 switch = '0 : OFF\n1 : ON'
16 cv.createTrackbar(switch, 'image', 0, 1, nothing)
17
18 while(1):
19     cv.imshow('image', img)
20     k = cv.waitKey(1) & 0xFF
21     if k == 27:
```

**Run:** python\_opencv\_bar

**Preview Window:**

The preview window titled "image" displays a black image. Three trackbars are visible: "B" (Blue) at 120, "G" (Green) at 170, and "R" (Red) at 66. A fourth trackbar, "switch", has values "0 : OFF" and "1 : ON". The status bar at the bottom of the window shows "0 : OFF...ON: 1".

Gambar 2.44 Membuat Trackbar ON dan OFF

## 2.9.4 Trackbar menampilkan text kontras

```
1 import numpy as np
2
3 import cv2 as cv
4
5 def nothing(x):
6
7     print(x)
8
9 # membuat gambar hitam
10
11 cv.namedWindow('image')
12
13 cv.createTrackbar('CP', 'image', 10, 400, nothing)
14
15 switch = '0 : OFF\n1 : ON'
16
17 cv.createTrackbar(switch, 'image', 0, 1, nothing)
18
19 while(1):
20
21     img = cv.imread('najib.jpg')
22
23     pos = cv.getTrackbarPos('CP', 'image')
24
25     font = cv.FONT_HERSHEY_SIMPLEX
26
27     cv.putText(img, str(pos), (50, 150), font, 6, (0, 255,
255), 10)
28
29     k = cv.waitKey(1) & 0xFF
30
31     if k == 27:
32
33         break
34
35     s = cv.getTrackbarPos(switch, 'image')
36
37     if s == 0:
38
39         pass
40
41     else:
42
43         img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
44
45     img = cv.imshow('image', img)
46
47 cv.destroyAllWindows()
```

- i. import numpy
- ii. import cv
- iii. membuat variable untuk menampilkan
- iv. buat jendela dengan nama image

- v. membuat Trackbar CP dari 10 sampai 400, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- vi. membuat tampilan on dan off
- vii. membuat Trackbar on dan off dari 0 sampai 1, kemudian data di tampilkan menggunakan nothing sesuai kursor yang di geser
- viii. membuat pengulangan
- ix. menampilkan frame dengan isi frame img yang telah dipanggil dari file
- x. mengambil fungsi Trackbar menggunakan getTrackbarPos pada CP
- xi. kemudian atur font text yang akan di gunakan
- xii. atur text ukuran, wrna, font, dan mengambil string dari variable pos
- xiii. membuat waitKey jika sama dengan 27 maka akan berakhir
- xiv. kemudian mengambil fungsi trackbar kedua yaitu switch
- xv. jika switch 0 maka tidak akan mengubah apapun
- xvi. jika switch 1 maka akan merubah kontras warna menjadi abu abu
- xvii. menampilkan frame
- xviii. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows the PyCharm IDE interface. On the left, the project structure is visible with files like 'python\_opencv\_bar.py' selected. The main window displays the code for 'python\_opencv\_bar.py'. The code imports cv2, defines a function 'nothing', and creates a window 'image'. It adds a trackbar 'CP' with a range from 10 to 400. A switch trackbar is also created. The main loop reads 'najib.jpg', gets the position of the 'CP' trackbar, and puts text '10' at that position. The image window shows a person's face with the number '10' overlaid in yellow, circled in yellow.

```
2 import cv2 as cv
3
4 def nothing(x):
5     print(x)
6
7 # membuat gambar hitam
8 cv.namedWindow('image')
9
10 cv.createTrackbar('CP', 'image', 10, 400, nothing)
11
12 switch = '0 : OFF\n1 : ON'
13 cv.createTrackbar(switch, 'image', 0, 1, nothing)
14
15 while(1):
16     img = cv.imread('najib.jpg')
17     pos = cv.getTrackbarPos('CP', 'image')
18     font = cv.FONT_HERSHEY_SIMPLEX
19     cv.putText(img, str(pos), (50, 150), font, 1, (255, 255, 255), 2)
20
21     k = cv.waitKey(1) & 0xFF
22     if k == 27:
```

Gambar 2.45 Trackbar menampilkan text kontras

The screenshot shows the PyCharm IDE interface. On the left, the project structure displays files like 'python\_opencv\_bar.py' and various image files. The code editor window shows the script 'python\_opencv\_bar.py' with the following content:

```
1 import cv2 as cv
2
3 def nothing(x):
4     print(x)
5
6
7 # membuat gambar hitam
8 cv.namedWindow('image')
9
10 cv.createTrackbar('CP', 'image', 10, 400, nothing)
11
12 switch = '0 : OFF\n1 : ON'
13 cv.createTrackbar(switch, 'image', 0, 1, nothing)
14
15
16 while(1):
17     img = cv.imread('najib.jpg')
18     pos = cv.getTrackbarPos('CP', 'image')
19     font = cv.FONT_HERSHEY_SIMPLEX
20
21     cv.putText(img, str(pos), (50, 150), font, 1, (255, 255, 255), 2)
22
23     k = cv.waitKey(1) & 0xFF
24     if k == 27:
```

The right side of the interface shows the resulting image titled 'image'. The image is a portrait of a man in a white shirt and black tie. A large yellow number '179' is overlaid on the top right of his head. A trackbar at the top of the image window is labeled 'CP: 179' with a slider set to 179. Below it is another trackbar labeled '0 : OFF...ON: 0'.

Gambar 2.46 Trackbar menampilkan text kontras

The screenshot shows the PyCharm IDE interface. On the left, the project structure displays files like `python_opencv_bar.py`, `najib.jpg`, and `najib_copy.png`. The code editor window contains the following Python script:

```
python_opencv_bar.py
1 import cv2 as cv
2
3 def nothing(x):
4     print(x)
5
6
7 # membuat gambar hitam
8 cv.namedWindow('image')
9
10 cv.createTrackbar('CP', 'image', 10, 400, nothing)
11
12 switch = '0 : OFF\n1 : ON'
13 cv.createTrackbar(switch, 'image', 0, 1, nothing)
14
15 while(1):
16     img = cv.imread('najib.jpg')
17     pos = cv.getTrackbarPos('CP', 'image')
18     font = cv.FONT_HERSHEY_SIMPLEX
19     cv.putText(img, str(pos), (50, 150), font, 1, (255, 255, 255), 2)
20
21     k = cv.waitKey(1) & 0xFF
22     if k == 27:
```

The right side of the interface shows a window titled "image" displaying a grayscale portrait of a man. A large white text "179" is overlaid on the image, indicating the current value of the trackbar.

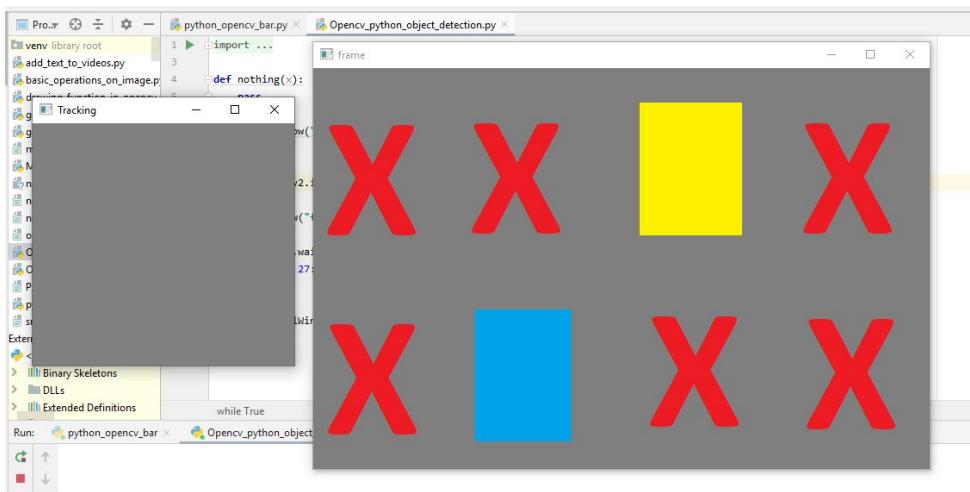
Gambar 2.47 Trackbar menampilkan text kontras

## 2.10 Object Detection dan Object Tracking menggunakan HSV Color Space

### 2.10.1 Membuat frame kosong

```
1 import cv2
2
3 import numpy as np
4
5 def nothing(x):
6
7     pass
8
9 cv2.namedWindow("Tracking")
10
11 while True:
12
13     frame = cv2.imread('PARKIR.png')
14
15     cv2.imshow("frame", frame)
16
17     key = cv2.waitKey(1)
18
19     if key == 27:
20
21         break
22
23 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv
- iii. membuat variable untuk menampilkan
- iv. buat jendela atau frame baru dengan nama Tracking
- v. buat perulangan jika true maka akan melanjutkan perulangan
- vi. mengambil gambar dari file
- vii. menampilkan gambar pada frame dengan nama frame
- viii. membuat waitKey jika sama dengan 27 maka akan berakhir
- ix. destroyAllWindows digunakan untuk menutup frame.



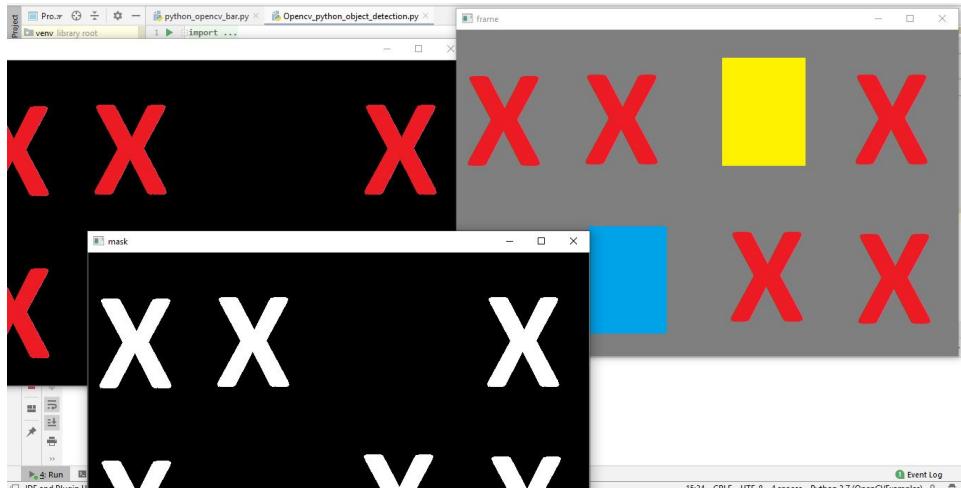
**Gambar 2.48** Membuat frame kosong

## 2.10.2 Hsv color space

```
1 import cv2
2
3 import numpy as np
4
5 def nothing(x):
6
7     pass
8
9 #cv2.namedWindow("Tracking")
10
11 while True:
12
13     frame = cv2.imread('PARKIR.png')
14
15     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
16
17     l_b = np.array([110, 50, 50])
18
19     u_b = np.array([130, 255, 255])
20
21     mask = cv2.inRange(hsv, l_b, u_b)
22
23     res = cv2.bitwise_and(frame, frame, mask=mask)
24
25     cv2.imshow("frame", frame)
26
27     cv2.imshow("mask", mask)
28
29     cv2.imshow("res", res)
30
31     key = cv2.waitKey(1)
32
33     if key == 27:
34
35         break
36
37 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv
- iii. membuat variable untuk menampilkan
- iv. buat perulangan jika true maka akan melanjutkan perintah perulangan
- v. mengambil gambar dari file
- vi. membuat variable dengan memanggil fungsi hsv
- vii. membuat variable lb untuk menentukan angka pertama yaitu h ke dua s dan ketiga v
- viii. membuat variable ub untuk menentukan angka pertama yaitu h ke dua s dan ketiga v

- ix. variable mask di gunakan untuk menggabungkan hsv lb dan ub, mask juga membuat kontras warna menjadi hitam putih
- x. res digunakan untuk menggabungkan mask dan frame yang pertama
- xi. menampilkan frame asli, frame mask, dan frame res
- xii. membuat waitKey jika sama dengan 27 maka akan berakhir
- xiii. destroyAllWindows digunakan untuk menutup frame.



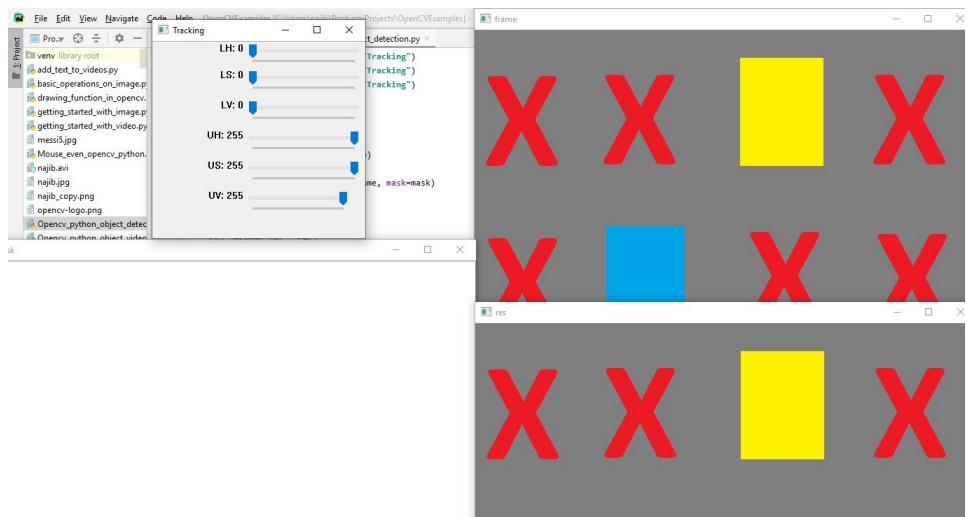
Gambar 2.49 Hsv color space

### 2.10.3 Hsv color space menggunakan Trackbar

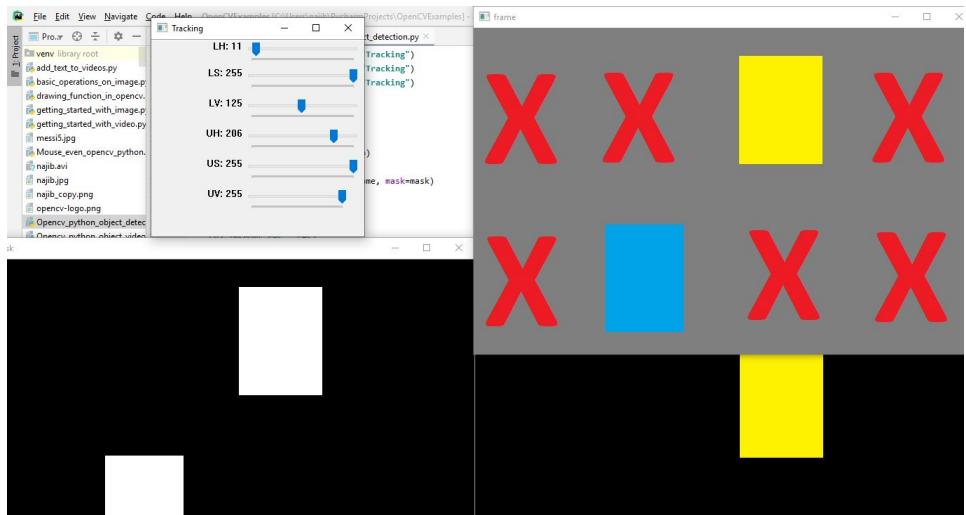
```
1 import cv2
2
3 import numpy as np
4
5 def nothing(x):
6     pass
7
8 cv2.namedWindow("Tracking")
9
10 cv2.createTrackbar("LH", "Tracking", 0, 255, nothing)
11
12 cv2.createTrackbar("LS", "Tracking", 0, 255, nothing)
13
14 cv2.createTrackbar("LV", "Tracking", 0, 255, nothing)
15
16 cv2.createTrackbar("UH", "Tracking", 255, 255, nothing)
17
18 cv2.createTrackbar("US", "Tracking", 255, 255, nothing)
19
20 cv2.createTrackbar("UV", "Tracking", 255, 255, nothing)
21
22 while True:
23     frame = cv2.imread('PARKIR.png')
24
25     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
26
27     l_h = cv2.getTrackbarPos("LH", "Tracking")
28     l_s = cv2.getTrackbarPos("LS", "Tracking")
29     l_v = cv2.getTrackbarPos("LV", "Tracking")
30
31     u_h = cv2.getTrackbarPos("UH", "Tracking")
32     u_s = cv2.getTrackbarPos("US", "Tracking")
33     u_v = cv2.getTrackbarPos("UV", "Tracking")
34
35     l_b = np.array([l_h, l_s, l_v])
36     u_b = np.array([u_h, u_s, u_v])
37
38     mask = cv2.inRange(hsv, l_b, u_b)
39
40     res = cv2.bitwise_and(frame, frame, mask=mask)
41
42     cv2.imshow("frame", frame)
43     cv2.imshow("mask", mask)
44     cv2.imshow("res", res)
45
46     key = cv2.waitKey(1)
47
48     if key == 27:
49         break
50
51 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv

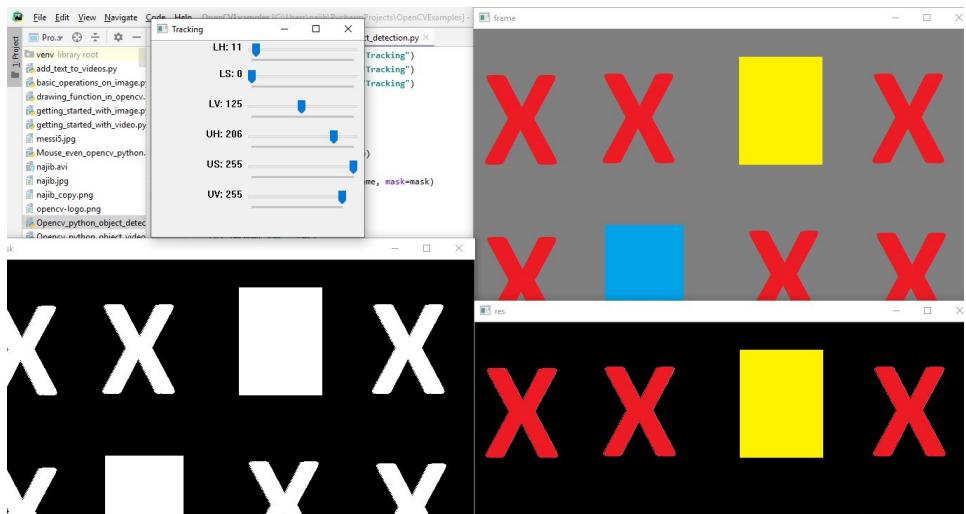
- iii. membuat variable untuk menampilkan
- iv. membuat frame baru dengan nama Tracking
- v. membuat isi dari frame Tracking untuk menentukan kontras warna yang di inginkan sesuai hsv
- vi. buat perulangan jika true maka akan melanjutkan perintah perulangan
- vii. mengambil gambar dari file
- viii. membuat variable dengan memanggil fungsi hsv, fungsi hsf digunakan untuk menampilkan warna tertentu saja jadi jika kita ingin menampilkan warna biru saja maka warna lain akan menjadi hitam bisa kita atur dengan mudah menggunakan traking tadi, jika sudah menemukan nomornya kita dapat masukan ke kede yang kita buat
- ix. kemudian ambil data dari Tracking menggunakan getTrackbarPos
- x. membuat variable lb untuk menentukan angka pertama yaitu h kedua s dan ketiga v
- xi. membuat variable ub untuk menentukan angka pertama yaitu h kedua s dan ketiga v
- xii. variable mask di gunakan untuk menggabungkan hsv lb dan ub, mask juga membuat kontras warna menjadi hitam putih
- xiii. res digunakan untuk menggabungkan mask dan frame yang pertama
- xiv. menampilkan frame asli, frame mask, dan frame res
- xv. membuat waitKey jika sama dengan 27 maka akan berakhir
- xvi. destroyAllWindows digunakan untuk menutup frame.



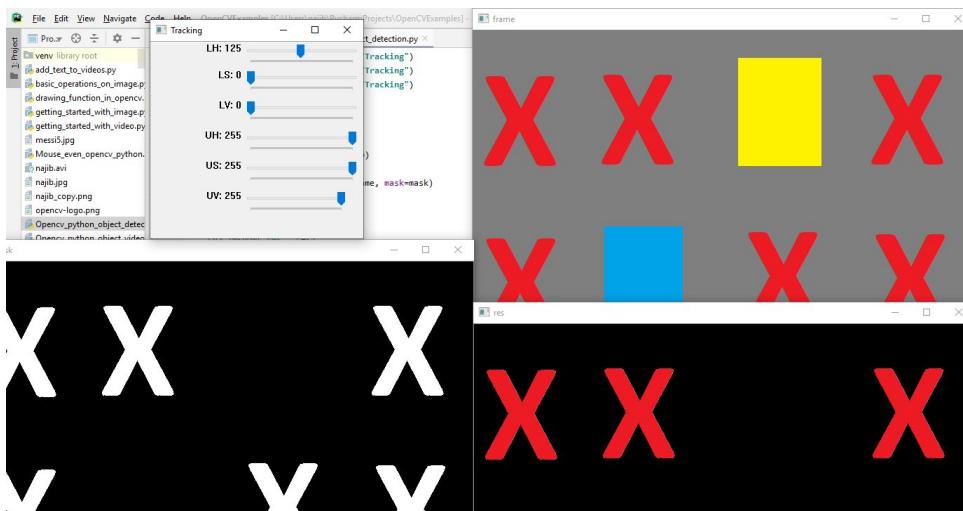
**Gambar 2.50** Hsv color space menggunakan Trackbar



**Gambar 2.51** Hsv color space menggunakan Trackbar



**Gambar 2.52** Hsv color space menggunakan Trackbar



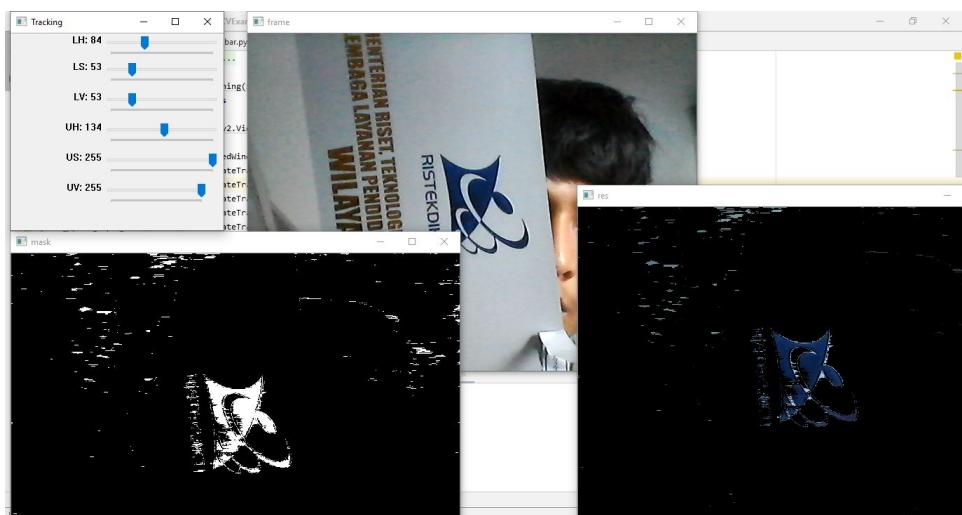
**Gambar 2.53** Hsv color space menggunakan Trackbar

## 2.10.4 Menentukan kontras warna menggunakan Trackbar pada video

```
1 import cv2
2
3 import numpy as np
4
5 def nothing(x):
6     pass
7
8 cap = cv2.VideoCapture(0);
9
10 cv2.namedWindow("Tracking")
11
12 cv2.createTrackbar("LH", "Tracking", 0, 255, nothing)
13
14 cv2.createTrackbar("LS", "Tracking", 0, 255, nothing)
15
16 cv2.createTrackbar("LV", "Tracking", 0, 255, nothing)
17
18 cv2.createTrackbar("UH", "Tracking", 255, 255, nothing)
19
20 cv2.createTrackbar("US", "Tracking", 255, 255, nothing)
21
22 cv2.createTrackbar("UV", "Tracking", 255, 255, nothing)
23
24 while True:
25     #frame = cv2.imread('smarties.png')
26     _, frame = cap.read()
27
28     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
29
30     l_h = cv2.getTrackbarPos("LH", "Tracking")
31     l_s = cv2.getTrackbarPos("LS", "Tracking")
32     l_v = cv2.getTrackbarPos("LV", "Tracking")
33
34     u_h = cv2.getTrackbarPos("UH", "Tracking")
35     u_s = cv2.getTrackbarPos("US", "Tracking")
36     u_v = cv2.getTrackbarPos("UV", "Tracking")
37
38     l_b = np.array([l_h, l_s, l_v])
39     u_b = np.array([u_h, u_s, u_v])
40
41     mask = cv2.inRange(hsv, l_b, u_b)
42
43     res = cv2.bitwise_and(frame, frame, mask=mask)
44
45     cv2.imshow("frame", frame)
46     cv2.imshow("mask", mask)
47     cv2.imshow("res", res)
48
49     key = cv2.waitKey(1)
50
51     if key == 27:
52         break
53
```

```
54 cap.release()  
55  
56 cv2.destroyAllWindows()
```

- i. import numpy
- ii. import cv
- iii. membuat variable untuk menampilkan
- iv. gunakan videocapture
- v. membuat frame baru dengan nama Tracking
- vi. membuat isi dari frame Tracking untuk menentukan kontras warna yang di inginkan sesuai hsv
- vii. buat perulangan jika true maka akan melanjutkan perintah perulangan
- viii. membaca video
- ix. membuat variable dengan memanggil fungsi hsv, fungsi hsf digunakan untuk menampilkan warna tertentu saja jadi jika kita ingin menampilkan warna biru saja maka warna lain akan menjadi hitam bisa kita atur dengan mudah menggunakan traking tadi, jika sudah menemukan nomornya kita dapat masukan ke kede yang kita buat
- x. kemudian ambil data dari Tracking menggunakan getTrackbarPos
- xi. membuat variable lb untuk menentukan angka pertama yaitu h kedua s dan ketiga v
- xii. membuat variable ub untuk menentukan angka pertama yaitu h kedua s dan ketiga v
- xiii. variable mask di gunakan untuk menggabungkan hsv lb dan ub, mask juga membuat kontras warna menjadi hitam putih
- xiv. res digunakan untuk menggabungkan mask dan frame yang pertama
- xv. menampilkan frame asli, frame mask, dan frame res
- xvi. membuat waitKey jika sama dengan 27 maka akan berakhir
- xvii. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.54** Menentukan kontras warna menggunakan Trackbar pada video

## 2.11 Image Thresholding

### 2.11.1 Membuat kontras warna menggunakan Binary

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('gradient.png',0)
7
8 _, th1 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
9
10
11 cv.imshow("Image", img)
12
13 cv.imshow("th1", th1)
14
15
16 cv.waitKey(0)
17
18 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. membuat frame image untuk menampilkan gambar
- vi. membuat frame untuk gambar yang baru
- vii. membuat waitKey supaya frame tidak langsung tertutup
- viii. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows a Python code editor with a file named `simple_thresholding.py`. The code imports cv2 and numpy, reads an image, applies a binary threshold, and displays the result. Two windows are shown: one titled `th1` displaying a binary image where the background is black and the foreground is white, and another window titled `Image` showing the original grayscale input image.

```
simple_thresholding.py
1 import cv2 as cv
2 import numpy as np
3
4 img = cv.imread('messi5.jpg')
5 t = cv.inRange(img, 0, 50)
6 cv.imshow('th1', t) # BINARY)
7 cv.imshow('Image', img)
8 cv.waitKey(0)
9 cv.destroyAllWindows()
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
```

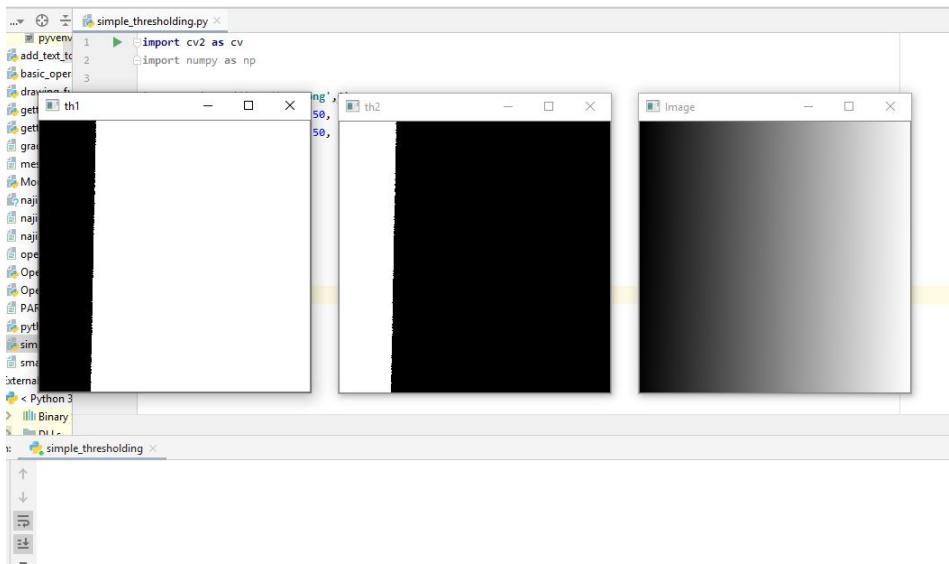
**Gambar 2.55** Membuat kontras warna menggunakan Binary

Pada gambar ini menunjukan pada gambar terdapat warna apa saja dan menggunakan Binary warna jadi hanya terbagi menjadi 2 dan di tentukan ukurannya sesuai kebutuhan.

## 2.11.2 Membuat kontras warna menggunakan Binary Inv

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('gradient.png',0)
7
8 _, th1 = cv.threshold(img, 50, 255, cv.THRESH_BINARY)
9
10 _, th2 = cv.threshold(img, 50, 255, cv.THRESH_BINARY_INV)
11
12
13 cv.imshow("Image", img)
14
15 cv.imshow("th1", th1)
16
17 cv.imshow("th2", th2)
18
19
20 cv.waitKey(0)
21
22 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. membuat gambar dengan menggunakan event binary inv yaitu kebalikan dari binary
- vi. membuat frame image untuk menampilkan gambar
- vii. membuat frame untuk gambar yang baru th1
- viii. membuat frame untuk gambar yang baru th2
- ix. membuat waitKey supaya frame tidak langsung tertutup
- x. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.56** Membuat kontras warna menggunakan Binary Inv

Pada gambar menunjukkan perbedaan antara binary dan binary inv yaitu kebalikannya warna putihnya di sebelah kiri.

### 2.11.3 Membuat kontras warna menggunakan Trunc

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('gradient.png',0)
7
8 _, th1 = cv.threshold(img, 50, 255, cv.THRESH_BINARY)
9
10 _, th2 = cv.threshold(img, 200, 255, cv.THRESH_BINARY_INV)
11
12 _, th3 = cv.threshold(img, 127, 255, cv.THRESH_TRUNC)
13
14
15 cv.imshow("Image", img)
16
17 cv.imshow("th1", th1)
18
19 cv.imshow("th2", th2)
20
21 cv.imshow("th3", th3)
22
23
24 cv.waitKey(0)
25
26 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. membuat gambar dengan menggunakan event binary inv yaitu kebalikan dari binary
- vi. membuat gambar dengan menggunakan event Trunc membuat kontras
- vii. membuat frame image untuk menampilkan gambar
- viii. membuat frame untuk gambar yang baru th1
- ix. membuat frame untuk gambar yang baru th2
- x. membuat frame untuk gambar yang baru th3
- xi. membuat waitKey supaya frame tidak langsung tertutup
- xii. destroyAllWindows digunakan untuk menutup frame.

The screenshot shows a Python code editor with a script named `simple_thresholding.py`. The code imports cv2 and numpy, reads an image, and applies three different thresholding methods: `cv.threshold`, `cv.truncate`, and `cv.threshold` again. It then displays the original image and the three thresholded versions in separate windows: `th1`, `th2`, and `th3`. Below the code editor is a terminal window showing the command `python simple_thresholding.py`.

```
simple_thresholding.py
1 import cv2 as cv
2 import numpy as np
3
4 img = cv.imread('gradient.jpg')
5 th1 = cv.threshold(img, 50, 255, cv.THRESH_BINARY)
6 th2 = cv.threshold(img, 50, 255, cv.THRESH_TRUNC)
7 th3 = cv.threshold(img, 50, 255, cv.THRESH_BINARY)
8
9
10 cv.imshow("Image", img)
11 cv.imshow("th1", th1)
12 cv.imshow("th2", th2)
13 cv.imshow("th3", th3)
14
15 cv.waitKey(0)
16 cv.destroyAllWindows()
```

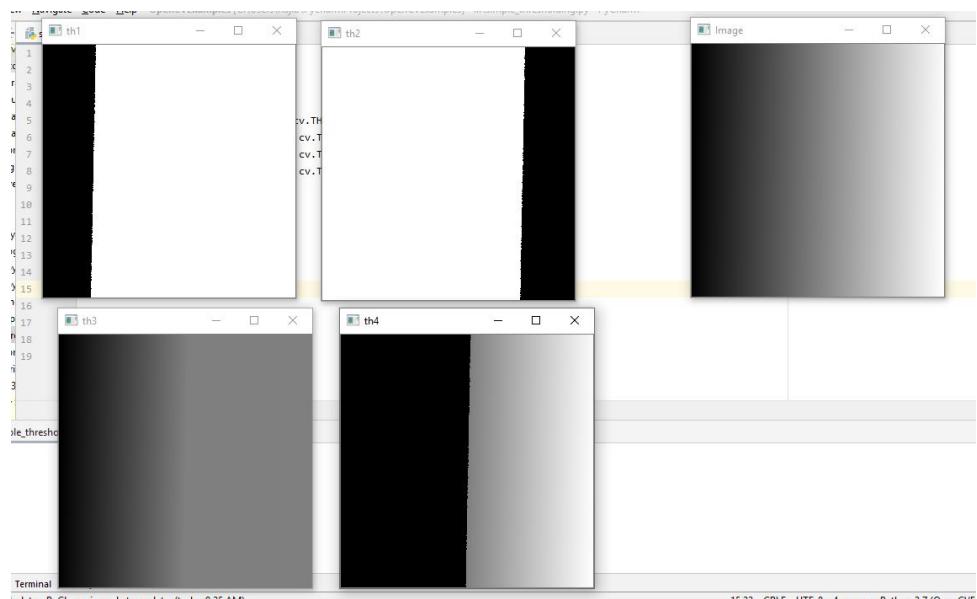
**Gambar 2.57** Membuat kontras warna menggunakan Trunc

Pada gambar menunjukan event Trunc mengubah gambar menjadi warna terrendahnya berada di tengah

#### 2.11.4 Membuat kontras warna menggunakan Tozero

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('gradient.png',0)
7
8 _, th1 = cv.threshold(img, 50, 255, cv.THRESH_BINARY)
9
10 _, th2 = cv.threshold(img, 200, 255, cv.THRESH_BINARY_INV)
11
12 _, th3 = cv.threshold(img, 127, 255, cv.THRESH_TRUNC)
13
14 _, th4 = cv.threshold(img, 127, 255, cv.THRESH_TOZERO)
15
16
17 cv.imshow("Image", img)
18
19 cv.imshow("th1", th1)
20
21 cv.imshow("th2", th2)
22
23 cv.imshow("th3", th3)
24
25 cv.imshow("th4", th4)
26
27 cv.waitKey(0)
28
29 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. membuat gambar dengan menggunakan event binary inv yaitu kebalikan dari binary
- vi. membuat gambar dengan menggunakan event Trunc membuat kontras
- vii. membuat gambar dengan menggunakan event Tozero berguna untuk memperjelas warna tergelap
- viii. membuat frame image untuk menampilkan gambar
- ix. membuat frame untuk gambar yang baru th1
- x. membuat frame untuk gambar yang baru th2
- xi. membuat frame untuk gambar yang baru th3
- xii. membuat frame untuk gambar yang baru th4
- xiii. membuat waitKey supaya frame tidak langsung tertutup
- xiv. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.58** Membuat kontras warna menggunakan Tozero

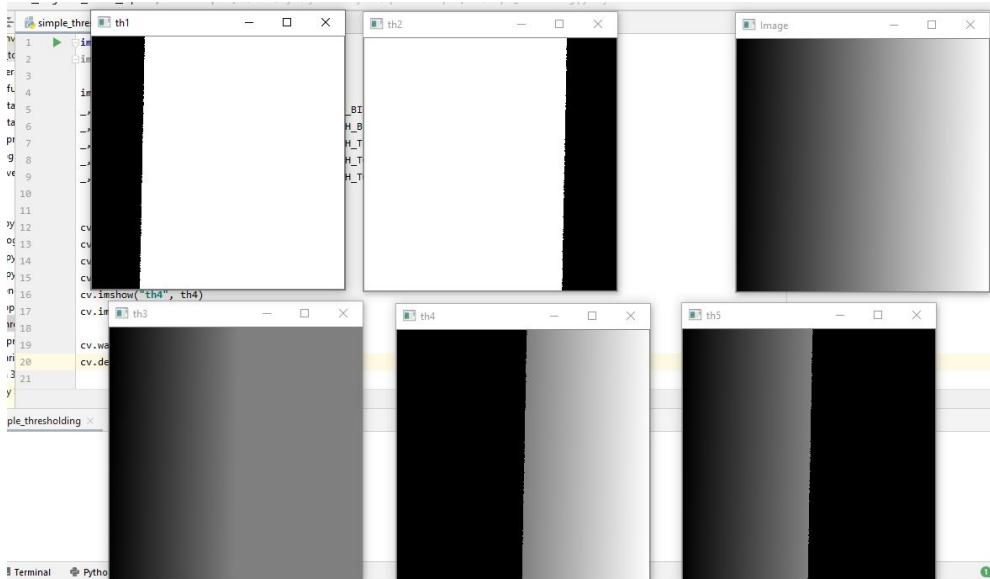
pada gambar menampilkan perbedaan antara Tozero dan yang lainnya Tozero menunjukkan menampilkan kontras tergela dan yang sebelahnya tetap sesuai gambar.

## 2.11.5 Membuat kontras warna menggunakan Tozero Inv

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('gradient.png',0)
7
8 _, th1 = cv.threshold(img, 50, 255, cv.THRESH_BINARY)
9
10 _, th2 = cv.threshold(img, 200, 255, cv.THRESH_BINARY_INV)
11
12 _, th3 = cv.threshold(img, 127, 255, cv.THRESH_TRUNC)
13
14 _, th4 = cv.threshold(img, 127, 255, cv.THRESH_TOZERO)
15
16 _, th5 = cv.threshold(img, 127, 255, cv.THRESH_TOZERO_INV)
17
18
19 cv.imshow("Image", img)
20
21 cv.imshow("th1", th1)
22
23 cv.imshow("th2", th2)
24
25 cv.imshow("th3", th3)
26
27 cv.imshow("th4", th4)
28
29 cv.imshow("th5", th5)
30
31
32 cv.waitKey(0)
33
34 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. membuat gambar dengan menggunakan event binary inv yaitu kebalikan dari binary
- vi. membuat gambar dengan menggunakan event Trunc membuat kontras
- vii. membuat gambar dengan menggunakan event Tozero berguna untuk memperjelas warna tergelap
- viii. membuat gambar dengan menggunakan event Tozero Inv kebalikan dari tonzero
- ix. membuat frame image untuk menampilkan gambar

- x. membuat frame untuk gambar yang baru th1
- xi. membuat frame untuk gambar yang baru th2
- xii. membuat frame untuk gambar yang baru th3
- xiii. membuat frame untuk gambar yang baru th4
- xiv. membuat frame untuk gambar yang baru th5
- xv. membuat waitKey supaya frame tidak langsung tertutup
- xvi. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.59** Membuat kontras warna menggunakan Tonzero Inv

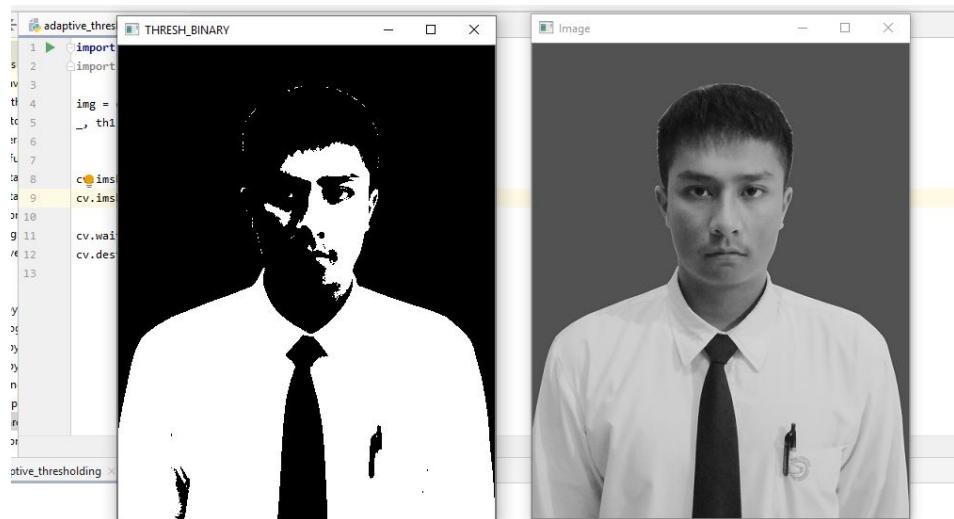
Tonzero Inv merupakan kebalikan dari tonzero menampilkan kontras tergelap di sebelah kanan

## 2.12 Adative Thresholding

### 2.12.1 Mengubah warna gambar menggunakan Binary

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('najib.jpg',0)
7
8 _, th1 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
9
10
11 cv.imshow("Image", img)
12
13 cv.imshow("THRESH_BINARY", th1)
14
15
16 cv.waitKey(0)
17
18 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. membuat frame image untuk menampilkan gambar
- vi. membuat frame untuk gambar yang baru
- vii. membuat waitKey supaya frame tidak langsung tertutup
- viii. destroyAllWindows digunakan untuk menutup frame.



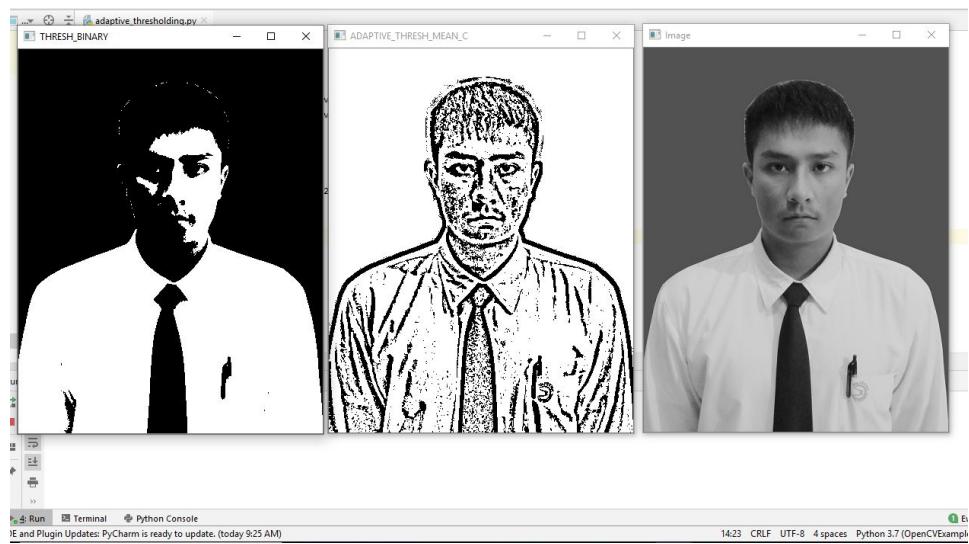
**Gambar 2.60** Mengubah warna gambar menggunakan Binary

Pada sebuah foto warna yang mendekati dengan putih akan berubah menjadi putih dan sebaliknya jika warna mendekati dengan warna hitam maka warna akan berubah menjadi warna hitam.

## 2.12.2 Menggunakan Adaptive Thresh Mean

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('najib.jpg',0)
7
8 _, th1 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
9
10 th2 = cv.adaptiveThreshold(img, 255, cv.ADAPTIVE_THRESH_MEAN_C,
11     , cv.THRESH_BINARY, 11, 2);
12
13 cv.imshow("Image", img)
14
15 cv.imshow("THRESH_BINARY", th1)
16
17 cv.imshow("ADAPTIVE_THRESH_MEAN_C", th2)
18
19
20 cv.waitKey(0)
21
22 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. menggunakan event Adaptive Thresh mean
- vi. membuat frame image untuk menampilkan gambar
- vii. membuat frame untuk gambar yang baru
- viii. membuat frame untuk adaptive
- ix. membuat waitKey supaya frame tidak langsung tertutup
- x. destroyAllWindows digunakan untuk menutup frame.



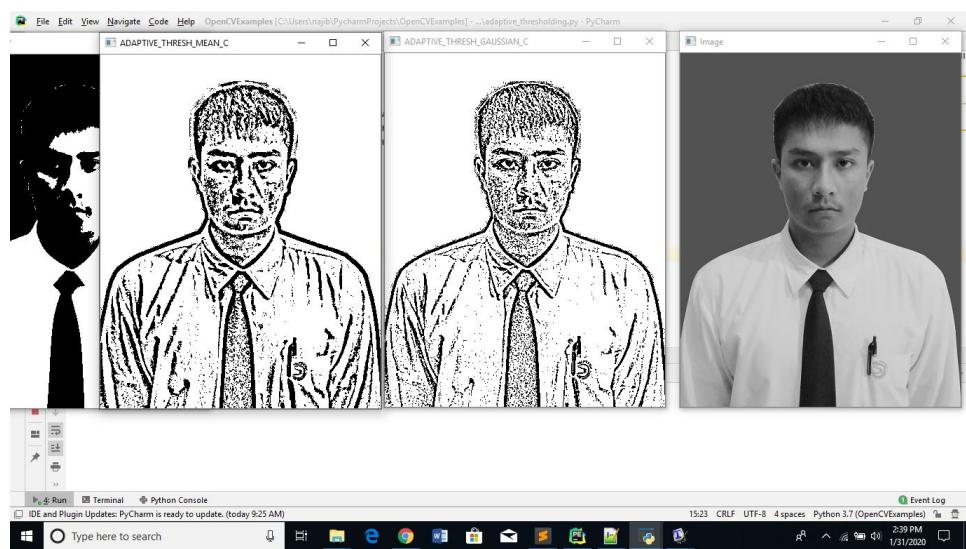
**Gambar 2.61** Menggunakan Adaptive Thresh Mean

pada gambar gambar menyesuaikan kontras hitamnya saja yang tidak ada kontras maka akan berubah menjadi putih

### 2.12.3 Menggunakan Adaptive Thresh Gaussian

```
1 import cv2 as cv
2
3 import numpy as np
4
5
6 img = cv.imread('najib.jpg',0)
7
8 _, th1 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
9
10 th2 = cv.adaptiveThreshold(img, 255, cv.ADAPTIVE_THRESH_MEAN_C
11     , cv.THRESH_BINARY, 11, 2);
12
13 th3 = cv.adaptiveThreshold(img, 255, cv.
14     ADAPTIVE_THRESH_GAUSSIAN_C, cv.THRESH_BINARY, 11, 2);
15
16 cv.imshow("Image", img)
17
18 cv.imshow("THRESH_BINARY", th1)
19
20 cv.imshow("ADAPTIVE_THRESH_MEAN_C", th2)
21
22 cv.imshow("ADAPTIVE_THRESH_GAUSSIAN_C", th3)
23
24 cv.waitKey(0)
25
26 cv.destroyAllWindows()
```

- i. import cv2
- ii. import numpy
- iii. membaca gambar dengan kontras warna hitam putih
- iv. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- v. menggunakan event Adaptive Thresh mean
- vi. menggunakan event Adaptive Thresh Gaussian
- vii. membuat frame image untuk menampilkan gambar
- viii. membuat frame untuk gambar yang baru
- ix. membuat frame untuk adaptive mean
- x. membuat frame untuk adaptive gaussian
- xi. membuat waitKey supaya frame tidak langsung tertutup
- xii. destroyAllWindows digunakan untuk menutup frame.



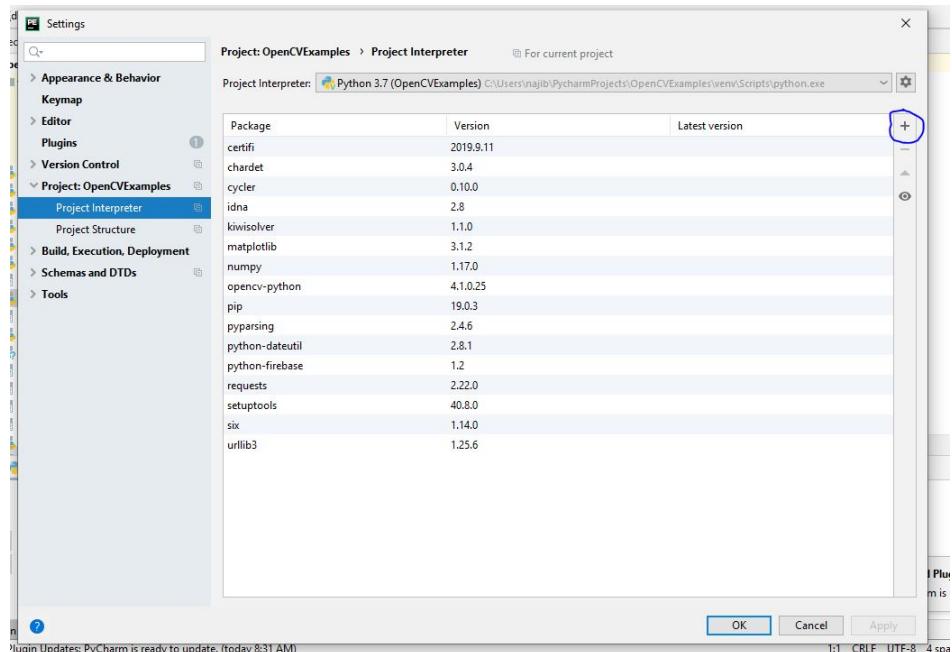
**Gambar 2.62** Menggunakan Adaptive Thresh Gaussian

Event adaptive Thresh Gaussian terlihat lebih smooth dibandingkan dengan event Adaptive Thresh Mean.

## 2.13 Menggunakan Matplotlib pada OpenCV

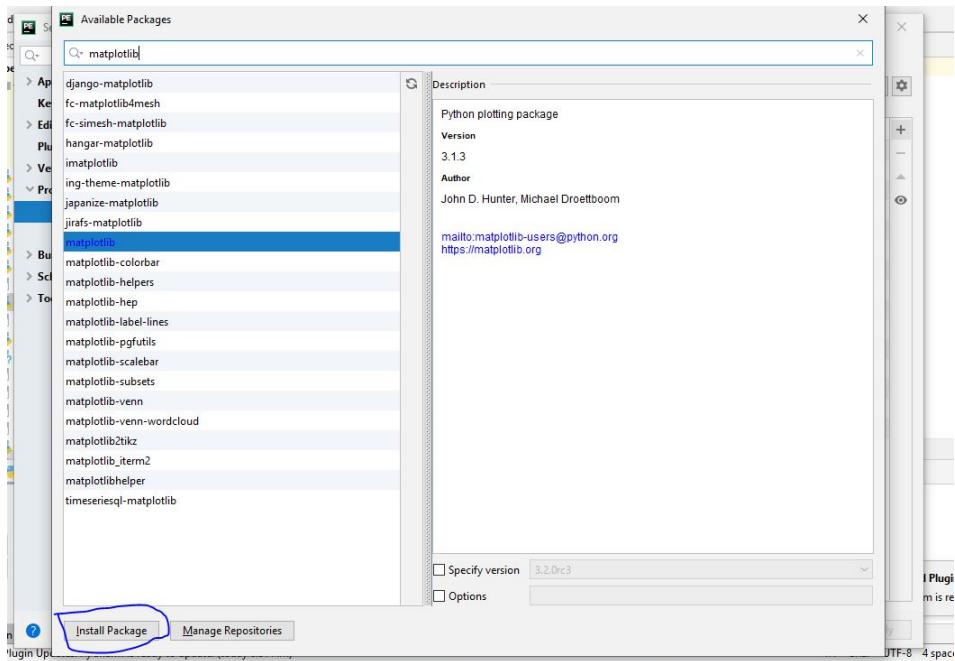
### 2.13.1 Installasi Matplotlib

Menginstall Matplotlib pada pycharm yang perlu kita lakukan masuk pada menu settings kemudian masuk pada menu Project Interpreter lalu klik tambah untuk menambahkan library yang ada pada project kita.



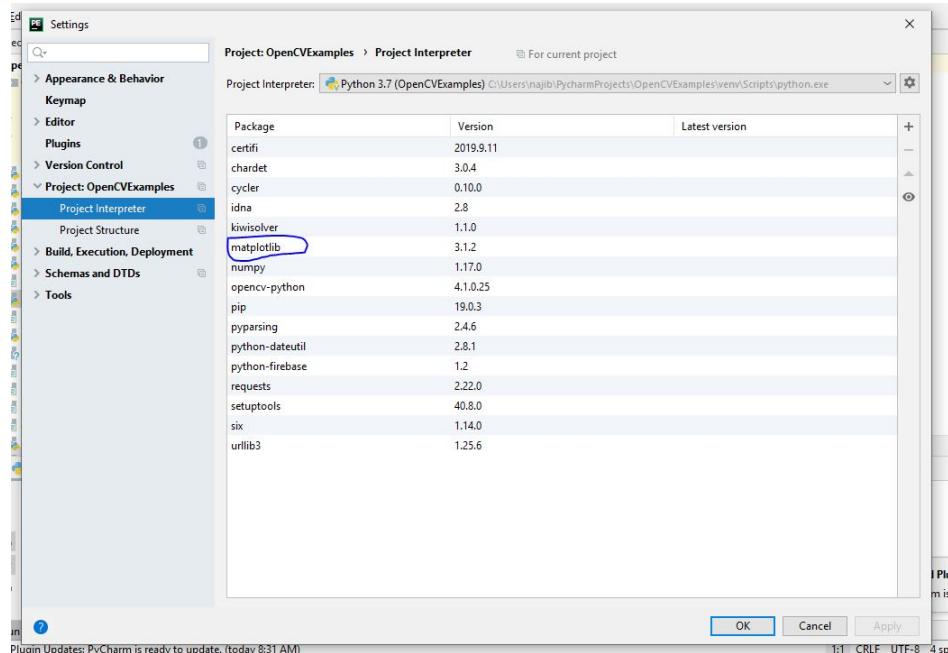
Gambar 2.63 Installasi matplotlib

kemudian cari pada pencarian yaitu matplotlib jika sudah menemukan kita hanya perlu klik laku insatall package.



Gambar 2.64 Instalasi matplotlib

jika matplotlib sudah terinstall maka akan di tampilkan pada halaman daftar library yang di gunakan pada project seperti pada gambar.

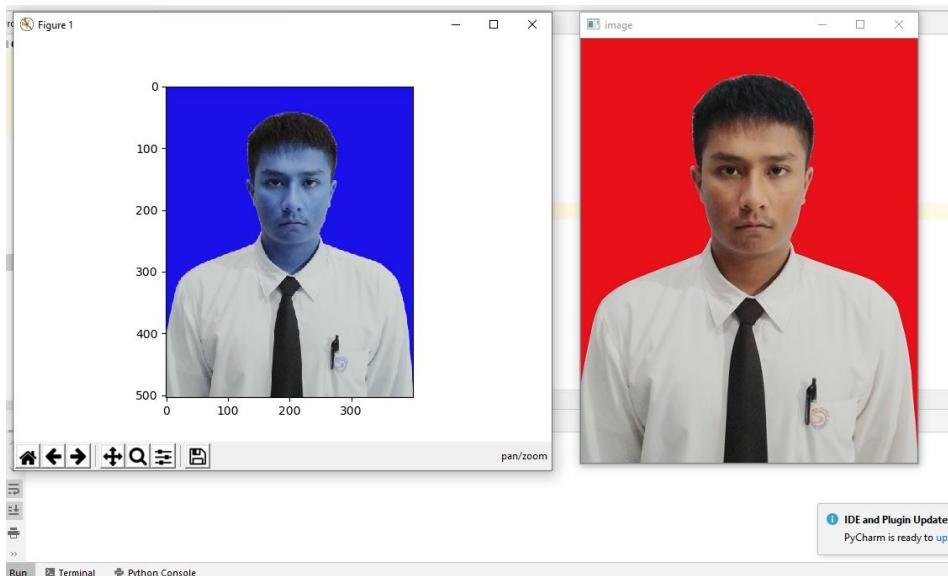


**Gambar 2.65** Installasi matplotlib

### 2.13.2 Menampilkan gambar menggunakan matplotlib

```
1 import cv2
2
3 from matplotlib import pyplot as plt
4
5
6 img = cv2.imread('najib.jpg', -1)
7
8 cv2.imshow('image', img)
9
10
11 plt.imshow(img)
12
13 plt.show()
14
15
16 cv2.waitKey(0)
17
18 cv2.destroyAllWindows()
```

- i. import cv
- ii. import matplotlib
- iii. mengambil gambar pada file menggunakan imread dengan kontras -1 yang artinya sesuai asli gambar
- iv. membuat frame dengan nama image
- v. menampilkan gambar
- vi. menampilkan gambar sesuai matplotlib
- vii. membuat waitKey supaya frame tidak langsung tertutup
- viii. destroyAllWindows digunakan untuk menutup frame.



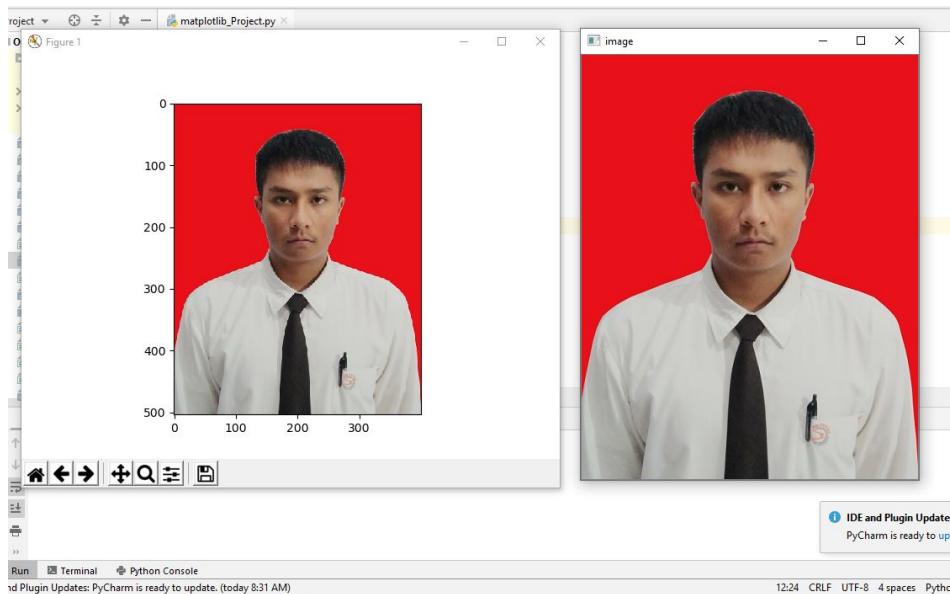
**Gambar 2.66** Menampilkan gambar menggunakan matplotlib

pada gambar yang di tampilkan menggunakan plot pada framenya terdapat berbagai tools dapat di zoom, mengubah ukuran gambar yang di tampilkan sesuai yang di inginkan dan dapat belihat titik x dan y

### 2.13.3 Mengubah warna yang ditampilkan matplotlib

```
1 import cv2
2
3 from matplotlib import pyplot as plt
4
5
6 img = cv2.imread('najib.jpg', -1)
7
8 cv2.imshow('image', img)
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12
13 plt.imshow(img)
14
15 plt.show()
16
17
18 cv2.waitKey(0)
19
20 cv2.destroyAllWindows()
```

- i. import cv
- ii. import matplotlib
- iii. mengambil gambar pada file menggunakan imread dengan kontras -1 yang artinya sesuai asli gambar
- iv. membuat frame dengan nama image
- v. membuat tampilan yang di tampilkan menggunakan matplotlib warnanya sesuai aslinya
- vi. menampilkan gambar
- vii. menampilkan gambar sesuai matplotlib
- viii. membuat waitKey supaya frame tidak langsung tertutup
- ix. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.67** Mengubah warna yang ditampilkan matplotlib

Pada gambar sebelumnya gambar yang di tampilkan menggunakan matplotlib berwarna biru, menggunakan cvtcolor mwmbuat yang di tampilkan sesuai dengan aslinya

untuk mengubah ukuran kita dapat mengubahnya seperti pada gambar sesuai yang di inginkan

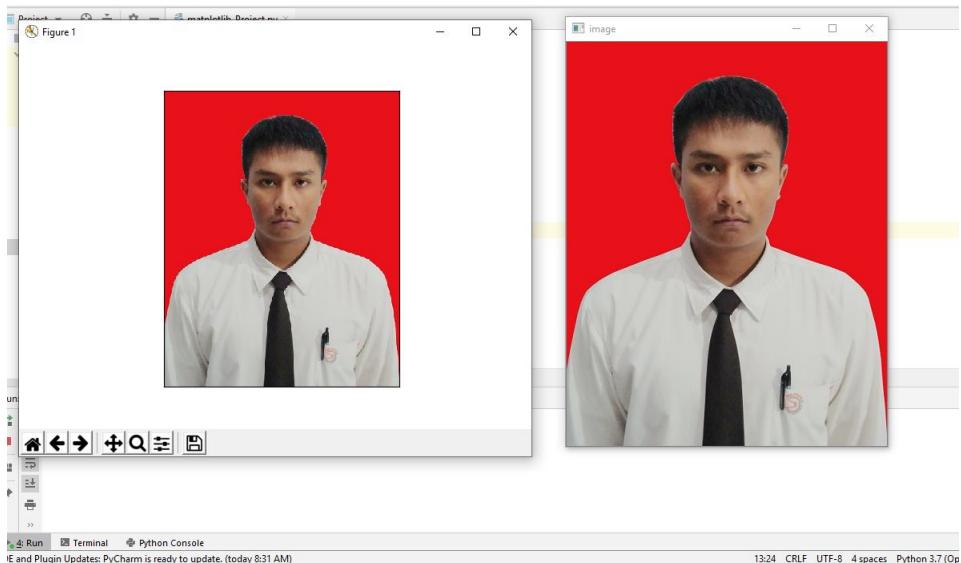


**Gambar 2.68** Mengubah warna yang ditampilkan matplotlib

## 2.13.4 Menghilangkan koordinat x dan y pada tampilan matplotlib

```
1 import cv2
2
3 from matplotlib import pyplot as plt
4
5
6 img = cv2.imread('najib.jpg', -1)
7
8 cv2.imshow('image', img)
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12
13 plt.imshow(img)
14
15 plt.xticks([]), plt.yticks([])
16
17 plt.show()
18
19
20 cv2.waitKey(0)
21
22 cv2.destroyAllWindows()
```

- i. import cv
- ii. import matplotlib
- iii. mengambil gambar pada file menggunakan imread dengan kontras -1 yang artinya sesuai asli gambar
- iv. membuat frame dengan nama image
- v. membuat tampilan yang di tampilkan menggunakan matplotlib warnanya sesuai aslinya
- vi. menampilkan gambar xticks dan yticks untuk menghilangkan nomor nomor koordinat x dan y yang di tampilkan oleh matplotlib
- vii. menampilkan gambar sesuai matplotlib
- viii. membuat waitKey supaya frame tidak langsung tertutup
- ix. destroyAllWindows digunakan untuk menutup frame.



**Gambar 2.69** Menghilangkan koordinat x dan y pada tampilan matplotlib

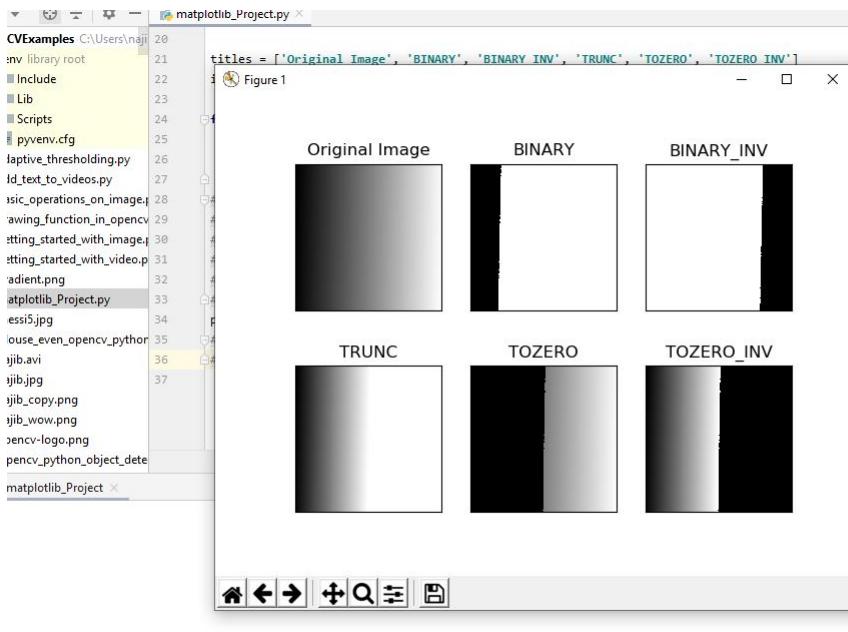
Pada gambar sebelumnya gambar yang di tampilkan oleh matplotlib terdapat nomor nomor koordinat x dan y, menggunakan xticks dan yticks koordinat tersebut di hilangkan

### 2.13.5 Menggabungkan frame menjadi satu frame

```
1 import cv2 as cv
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv.imread('gradient.png',0)
9
10 _, th1 = cv.threshold(img, 50, 255, cv.THRESH_BINARY)
11 _, th2 = cv.threshold(img, 200, 255, cv.THRESH_BINARY_INV)
12 _, th3 = cv.threshold(img, 127, 255, cv.THRESH_TRUNC)
13 _, th4 = cv.threshold(img, 127, 255, cv.THRESH_TOZERO)
14 _, th5 = cv.threshold(img, 127, 255, cv.THRESH_TOZERO_INV)
15
16 titles = ['Original Image', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
17
18 images = [img, th1, th2, th3, th4, th5]
19
20 for i in range(6):
21
22     plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
23
24     plt.title(titles[i])
25
26     plt.xticks([]),plt.yticks([])
27
28 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. membaca gambar dengan kontras warna hitam putih
- v. membuat gambar dengan menggunakan event binary untuk menampilkan warna hitamnya 127 yang artinya setengah dari frame
- vi. membuat gambar dengan menggunakan event binary inv yaitu kebalikan dari binary
- vii. membuat gambar dengan menggunakan event Trunc membuat kontras
- viii. membuat gambar dengan menggunakan event Tozero berguna untuk memperjelas warna tergelap
- ix. membuat gambar dengan menggunakan event Tozero Inv kebalikan dari tonzero

- x. membuat judul pada setiap gambar di satu frame
- xi. memetakan gambar gambar yang akan di tampilkan
- xii. menampilkan 6 gambar pada satu frame
- xiii. menampilkan gambar dengan 2 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xiv. menampilkan judul sesuai urutan judul
- xv. tidak menampilkan koordinat matplotlib
- xvi. menampilkan gambar



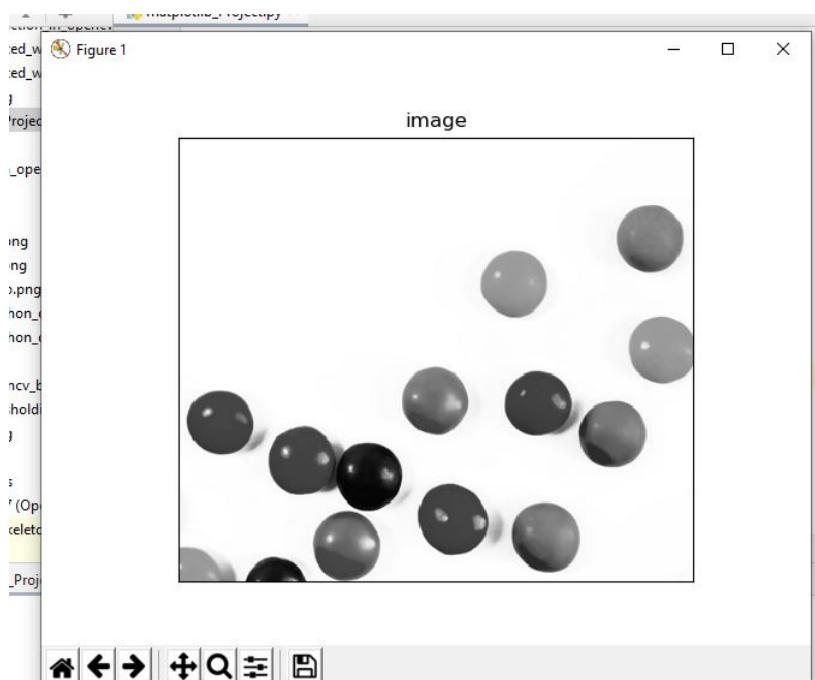
**Gambar 2.70** Menggabungkan frame menjadi satu frame

## 2.14 Morphologikan Transformations

### 2.14.1 Menggunakan imread grayscale

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9
10
11 titles = ['image']
12
13 images = [img]
14
15
16 for i in range(1):
17
18     plt.subplot(1, 1, i+1), plt.imshow(images[i], 'gray')
19
20     plt.title(titles[i])
21
22     plt.xticks([]), plt.yticks([])
23
24 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. membuat judul gambar
- vi. memanggil gambar
- vii. menampilkan 1 gambar pada satu frame
- viii. menampilkan gambar dengan 1 kolom 1 baris menampilkan gambar sesuai urutan gambar
- ix. menampilkan judul sesuai urutan judul
- x. tidak menampilkan koordinat matplotlib
- xi. menampilkan gambar



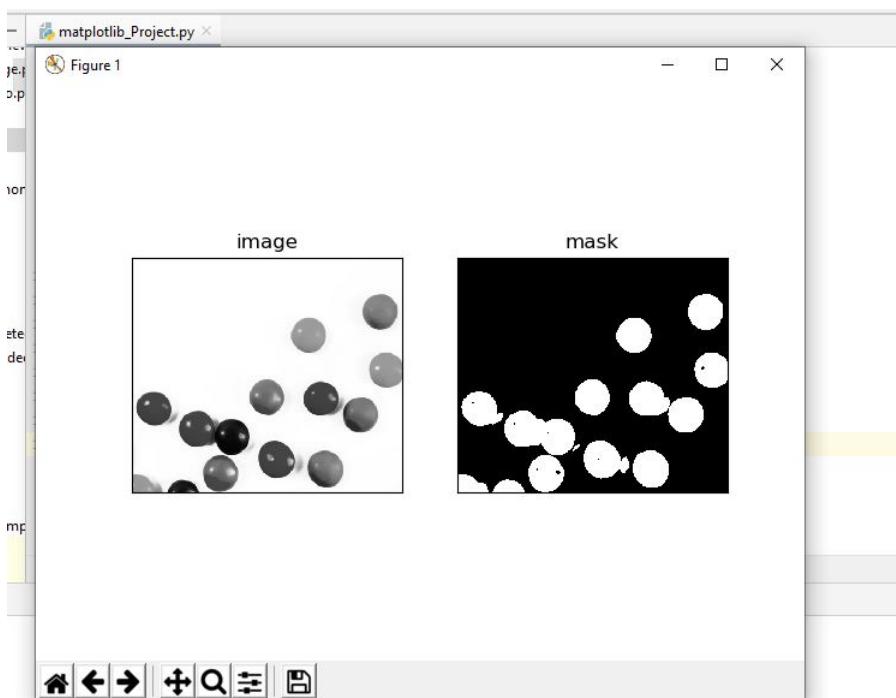
**Gambar 2.71** Menggunakan imread grayscale

Gambar berubah menjadi hitam putih

## 2.14.2 Menggunakan Thresh Binary Inv

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
10
11
12 titles = ['image', 'mask']
13
14 images = [img, mask]
15
16
17 for i in range(2):
18     plt.subplot(1, 2, i+1), plt.imshow(images[i], 'gray')
19
20     plt.title(titles[i])
21
22     plt.xticks([]), plt.yticks([])
23
24 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna
- vi. membuat judul gambar
- vii. memanggil gambar
- viii. menampilkan 2 gambar pada satu frame
- ix. menampilkan gambar dengan 1 kolom 2 baris menampilkan gambar sesuai urutan gambar
- x. menampilkan judul sesuai urutan judul
- xi. tidak menampilkan koordinat matplotlib
- xii. menampilkan gambar



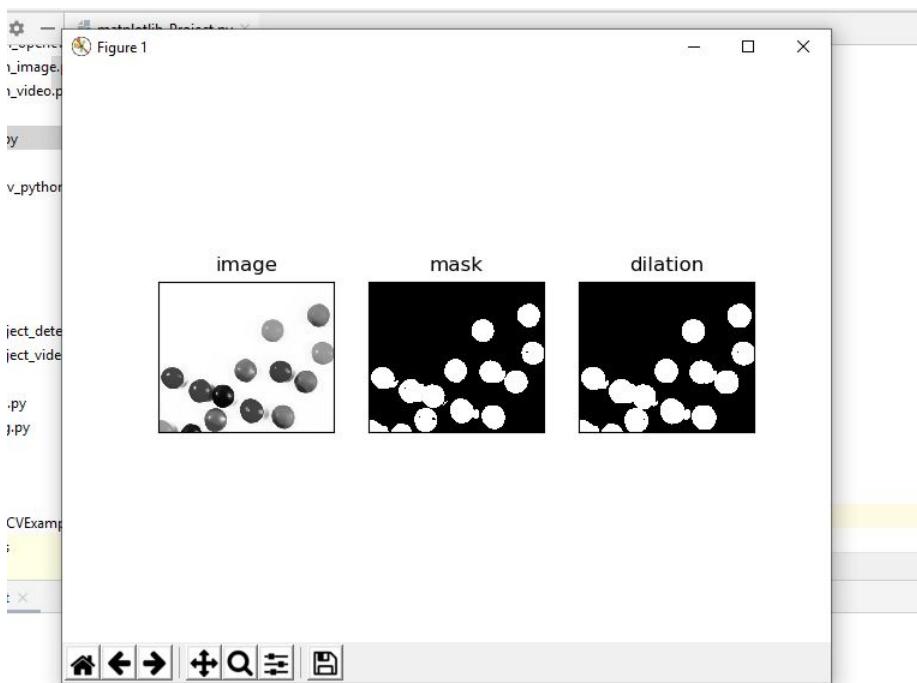
**Gambar 2.72** Menggunakan Thresh Binary Inv

### 2.14.3 Menggunakan Dilate

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9
10 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
11
12
13 kernel = np.ones((2,2), np.uint8)
14
15 dilation = cv2.dilate(mask, kernel, iterations=2)
16
17 titles = ['image', 'mask', 'dilation']
18
19 images = [img, mask, dilation]
20
21
22 for i in range(3):
23
24     plt.subplot(1, 3, i+1), plt.imshow(images[i], 'gray')
25
26     plt.title(titles[i])
27
28     plt.xticks([]), plt.yticks([])
29
30 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna
- vi. membuat karnal menggunakan numpy dengan type warna uint8
- vii. dilation di gunakan untuk membuat gambar lebih terlihat smooth, menghilangkan bintik bintik kecil
- viii. membuat judul gambar
- ix. memanggil gambar
- x. menampilkan 3 gambar pada satu frame
- xi. menampilkan gambar dengan 1 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xii. menampilkan judul sesuai urutan judul
- xiii. tidak menampilkan koordinat matplotlib

## xiv. menampilkan gambar



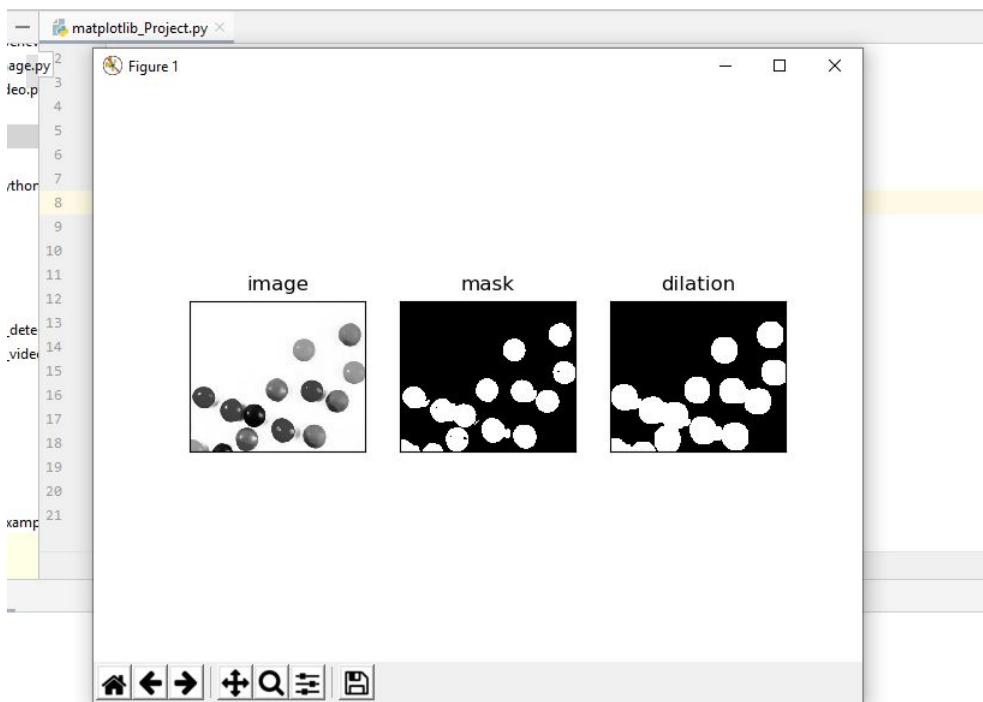
**Gambar 2.73** Menggunakan Dilate

#### 2.14.4 Mengubah ukuran karnal

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9
10 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
11
12
13 kernel = np.ones((5,5), np.uint8)
14
15 dilation = cv2.dilate(mask, kernel, iterations=2)
16
17 titles = ['image', 'mask', 'dilation']
18
19 images = [img, mask, dilation]
20
21
22 for i in range(3):
23     plt.subplot(1, 3, i+1), plt.imshow(images[i], 'gray')
24     plt.title(titles[i])
25     plt.xticks([]), plt.yticks([])
26
27 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna
- vi. membuat karnal menggunakan numpy dengan type warna uint8, semakin besar ukuran karnal semakin smooth gambar yang di hasilkan tetap sesuai kebutuhan penggunaan
- vii. dilation di gunakan untuk membuat gambar lebih terlihat smooth, menghilangkan bintik bintik kecil
- viii. membuat judul gambar
- ix. memanggil gambar
- x. menampilkan 4 gambar pada satu frame
- xi. menampilkan gambar dengan 2 kolom 2 baris menampilkan gambar sesuai urutan gambar

- xii. menampilkan judul sesuai urutan judul
- xiii. tidak menampilkan koordinat matplotlib
- xiv. menampilkan gambar



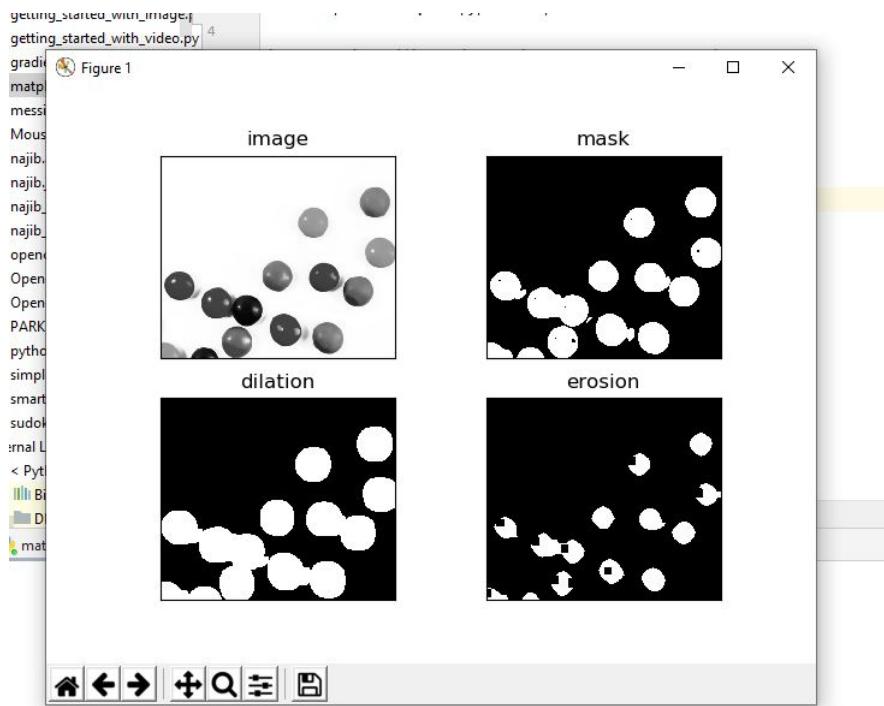
**Gambar 2.74** Mengubah ukuran karnal

## 2.14.5 Menggunakan Erode

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9
10 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
11
12 kernal = np.ones((5,5), np.uint8)
13
14 dilation = cv2.dilate(mask, kernal, iterations=2)
15
16 erosion = cv2.erode(mask, kernal, iterations=3)
17
18 titles = ['image', 'mask', 'dilation', 'erosion']
19
20 images = [img, mask, dilation, erosion]
21
22
23 for i in range(4):
24
25     plt.subplot(2, 2, i+1), plt.imshow(images[i], 'gray')
26
27     plt.title(titles[i])
28
29     plt.xticks([]), plt.yticks([])
30
31 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna
- vi. membuat karnal menggunakan numpy dengan type warna uint8, semakin besar ukuran karnal semakin smooth gambar yang dihasilkan tetap sesuai kebutuhan penggunaan
- vii. dilation di gunakan untuk membuat gambar lebih terlihat smooth, menghilangkan bintik bintik kecil
- viii. erosion atau yang pada event di sebut erode berguna untuk sedikit warna yang mendekati putih akan berubah menjadi hitam
- ix. membuat judul gambar
- x. memanggil gambar

- xi. menampilkan 5 gambar pada satu frame
- xii. menampilkan gambar dengan 2 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xiii. menampilkan judul sesuai urutan judul
- xiv. tidak menampilkan koordinat matplotlib
- xv. menampilkan gambar



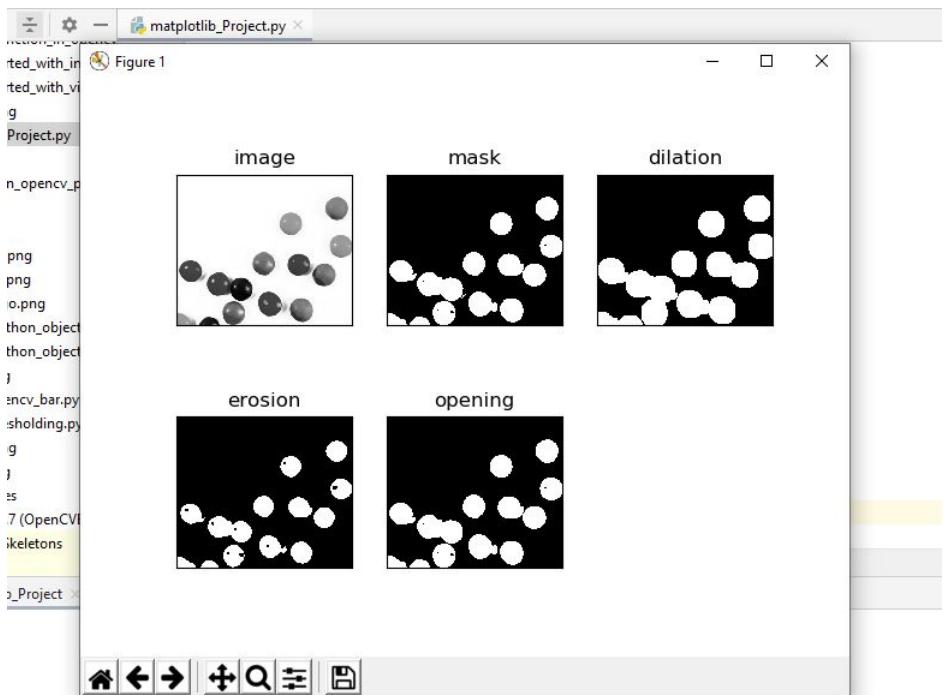
Gambar 2.75 Menggunakan Erode

## 2.14.6 Menggunakan MorphologyEx Open

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9
10 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
11
12
13 kernal = np.ones((5,5), np.uint8)
14
15
16 dilation = cv2.dilate(mask, kernal, iterations=2)
17
18 erosion = cv2.erode(mask, kernal, iterations=1)
19
20 opening = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernal)
21
22 titles = ['image', 'mask', 'dilation', 'erosion', 'opening']
23
24 images = [img, mask, dilation, erosion, opening]
25
26
27 for i in range(5):
28
29     plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
30
31     plt.title(titles[i])
32
33     plt.xticks([]), plt.yticks([])
34
35 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna
- vi. membuat karnal menggunakan numpy dengan type warna uint8, semakin besar ukuran karnal semakin smooth gambar yang di hasilkan tetap sesuai kebutuhan penggunaan
- vii. dilation di gunakan untuk membuat gambar lebih terlihat smooth, menghilangkan bintik bintik kecil
- viii. erosion atau yang pada event di sebut erode berguna untuk sedikit warna yang mendekati putih akan berubah menjadi hitam

- ix. morphology hampir sama dengan Thresh Binary Inv
- x. membuat judul gambar
- xi. memanggil gambar
- xii. menampilkan 6 gambar pada satu frame
- xiii. menampilkan gambar dengan 2 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xiv. menampilkan judul sesuai urutan judul
- xv. tidak menampilkan koordinat matplotlib
- xvi. menampilkan gambar



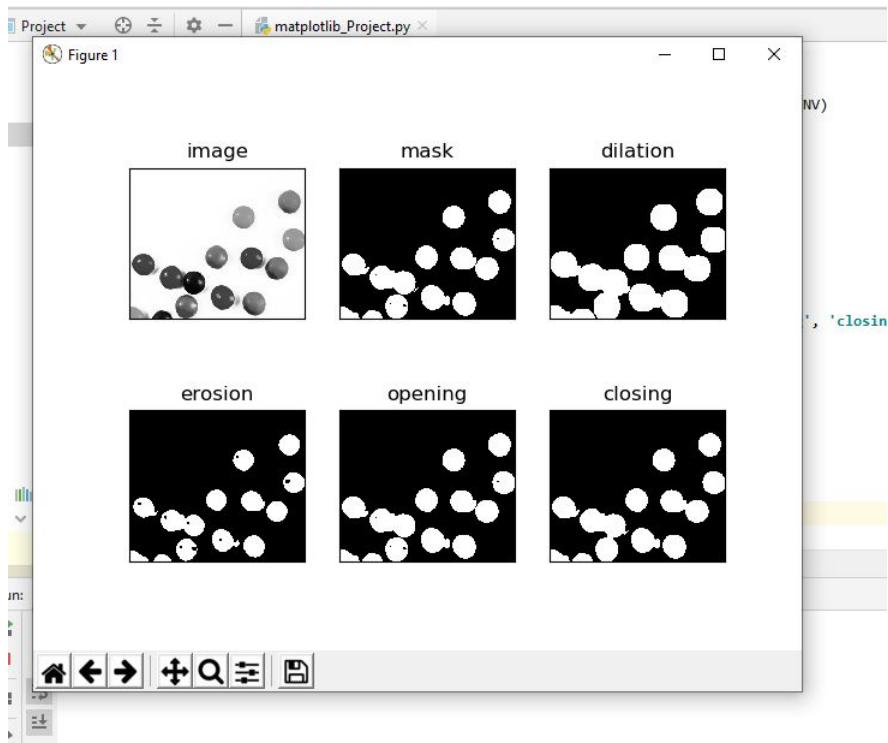
**Gambar 2.76** Menggunakan MorphologyEx Open

## 2.14.7 Menggunakan MorphologyEx Close

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9
10 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
11
12 kernel = np.ones((5,5), np.uint8)
13
14 dilation = cv2.dilate(mask, kernel, iterations=2)
15
16 erosion = cv2.erode(mask, kernel, iterations=1)
17
18 opening = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
19
20 closing = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
21
22
23 titles = ['image', 'mask', 'dilation', 'erosion', 'opening', 'closing']
24
25 images = [img, mask, dilation, erosion, opening, closing]
26
27
28 for i in range(6):
29
30     plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
31
32     plt.title(titles[i])
33
34     plt.xticks([]), plt.yticks([])
35
36 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna
- vi. membuat karnal menggunakan numpy dengan type warna uint8, semakin besar ukuran karnal semakin smooth gambar yang dihasilkan tetap sesuai kebutuhan penggunaan
- vii. dilation di gunakan untuk membuat gambar lebih terlihat smooth, menghilangkan bintik bintik kecil

- viii. erosion atau yang pada event di sebut erode berguna untuk sedikit warna yang mendekati putih akan berubah menjadi hitam
- ix. morphology open hampir sama dengan Thresh Binary Inv
- x. morphology close lebis smooth dibanding open
- xi. membuat judul gambar
- xii. memanggil gambar
- xiii. menampilkan 7 gambar pada satu frame
- xiv. menampilkan gambar dengan 2 kolom 4 baris menampilkan gambar sesuai urutan gambar
- xv. menampilkan judul sesuai urutan judul
- xvi. tidak menampilkan koordinat matplotlib
- xvii. menampilkan gambar



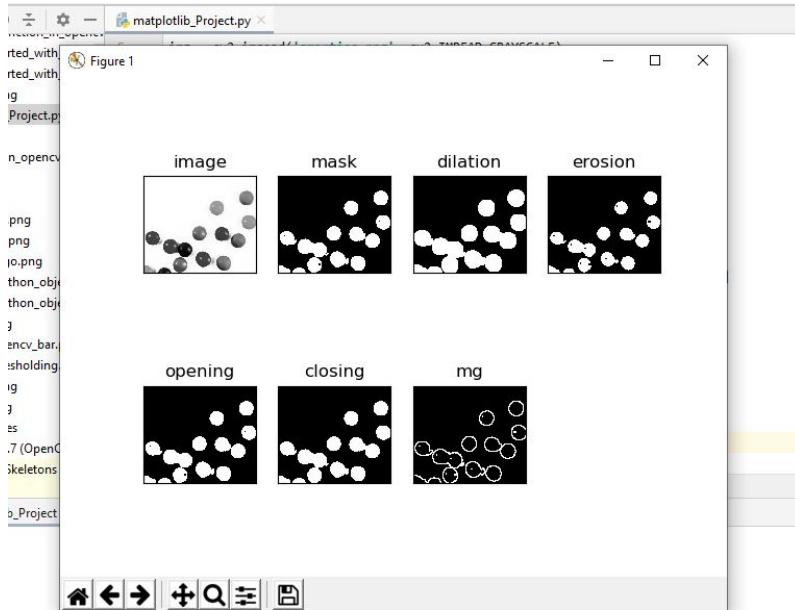
**Gambar 2.77** Menggunakan MorphologyEx Close

## 2.14.8 Menggunakan MorphologyEx Gradient

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
10
11
12 kernel = np.ones((5,5), np.uint8)
13
14
15 dilation = cv2.dilate(mask, kernel, iterations=2)
16
17 erosion = cv2.erode(mask, kernel, iterations=1)
18
19 opening = cv2.morphologyEx(mask, cv2.MORPHOPEN, kernel)
20
21 closing = cv2.morphologyEx(mask, cv2.MORPHCLOSE, kernel)
22
23 mg = cv2.morphologyEx(mask, cv2.MORPH_GRADIENT, kernel)
24
25
26 titles = ['image', 'mask', 'dilation', 'erosion', 'opening', 'closing', 'mg']
27
28 images = [img, mask, dilation, erosion, opening, closing, mg]
29
30
31 for i in range(7):
32     plt.subplot(2, 4, i+1), plt.imshow(images[i], 'gray')
33
34     plt.title(titles[i])
35
36     plt.xticks([]), plt.yticks([])
37
38 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna

- vi. membuat karnal menggunakan numpy dengan type warna uint8, semakin besar ukuran karnal semakin smooth gambar yang dihasilkan tetap sesuai kebutuhan penggunaan
- vii. dilation di gunakan untuk membuat gambar lebih terlihat smooth, menghilangkan bintik bintik kecil
- viii. erosion atau yang pada event di sebut erode berguna untuk sedikit warna yang mendekati putih akan berubah menjadi hitam
- ix. morphology open hampir sama dengan Thresh Binary Inv
- x. morphology close lebis smooth dibanding open
- xi. morphology gradient event ini mengambil warna luarnya saja warna dalamnya di ubah menjadi hitam
- xii. membuat judul gambar
- xiii. memanggil gambar
- xiv. menampilkan 8 gambar pada satu frame
- xv. menampilkan gambar dengan 2 kolom 4 baris menampilkan gambar sesuai urutan gambar
- xvi. menampilkan judul sesuai urutan judul
- xvii. tidak menampilkan koordinat matplotlib
- xviii. menampilkan gambar



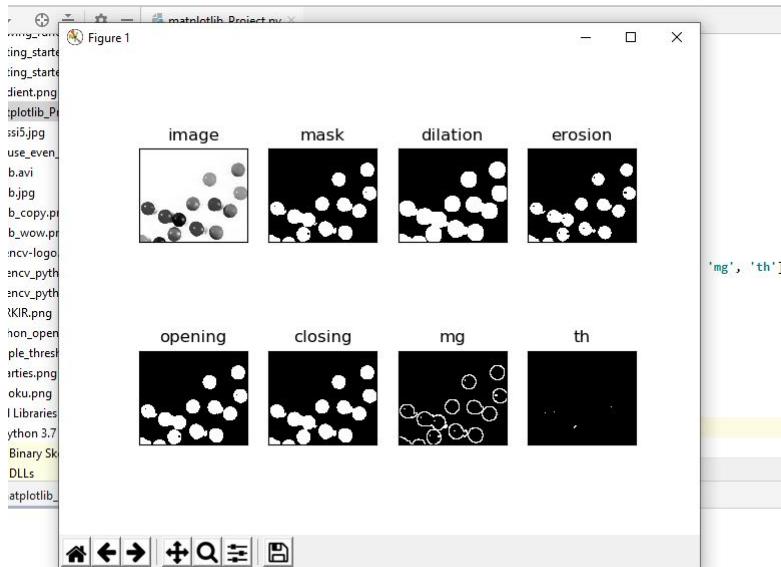
**Gambar 2.78** Menggunakan MorphologyEx Gradient

## 2.14.9 Menggunakan MorphologyEx Tophat

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('smarties.png', cv2.IMREAD_GRAYSCALE)
9 _, mask = cv2.threshold(img, 220, 255, cv2.THRESH_BINARY_INV)
10
11
12 kernel = np.ones((5,5), np.uint8)
13
14 dilation = cv2.dilate(mask, kernel, iterations=2)
15
16 erosion = cv2.erode(mask, kernel, iterations=1)
17
18 opening = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
19
20 closing = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
21
22 mg = cv2.morphologyEx(mask, cv2.MORPH_GRADIENT, kernel)
23
24 th = cv2.morphologyEx(mask, cv2.MORPH_TOPHAT, kernel)
25
26 titles = ['image', 'mask', 'dilation', 'erosion', 'opening', 'closing', 'mg', 'th']
27
28 images = [img, mask, dilation, erosion, opening, closing, mg, th]
29
30
31 for i in range(8):
32     plt.subplot(2, 4, i+1), plt.imshow(images[i], 'gray')
33     plt.title(titles[i])
34     plt.xticks([]), plt.yticks([])
35
36
37
38
39 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. memanggil gambar lalu gambar diberi event Grayscale untuk merubah gambar menjadi hitam putih
- v. mengambil gambar yang sama tetapi menggunakan event Thresh Binary Inv, menghitamkan warna putih dan memutihkan yang berwarna

- vi. membuat karnal menggunakan numpy dengan type warna uint8, semakin besar ukuran karnal semakin smooth gambar yang dihasilkan tetap sesuai kebutuhan penggunaan
- vii. dilation di gunakan untuk membuat gambar lebih terlihat smooth, menghilangkan bintik bintik kecil
- viii. erosion atau yang pada event di sebut erode berguna untuk sedikit warna yang mendekati putih akan berubah menjadi hitam
- ix. morphology open hampir sama dengan Thresh Binary Inv
- x. morphology close lebis smooth dibanding open
- xi. morphology gradient event ini mengambil warna luarnya saja warna dalamnya di ubah menjadi hitam
- xii. Tophat menampilkan warna tertentu
- xiii. membuat judul gambar
- xiv. memanggil gambar
- xv. menampilkan 8 gambar pada satu frame
- xvi. menampilkan gambar dengan 2 kolom 4 baris menampilkan gambar sesuai urutan gambar
- xvii. menampilkan judul sesuai urutan judul
- xviii. tidak menampilkan koordinat matplotlib
- xix. menampilkan gambar



**Gambar 2.79** Menggunakan MorphologyEx Tophat

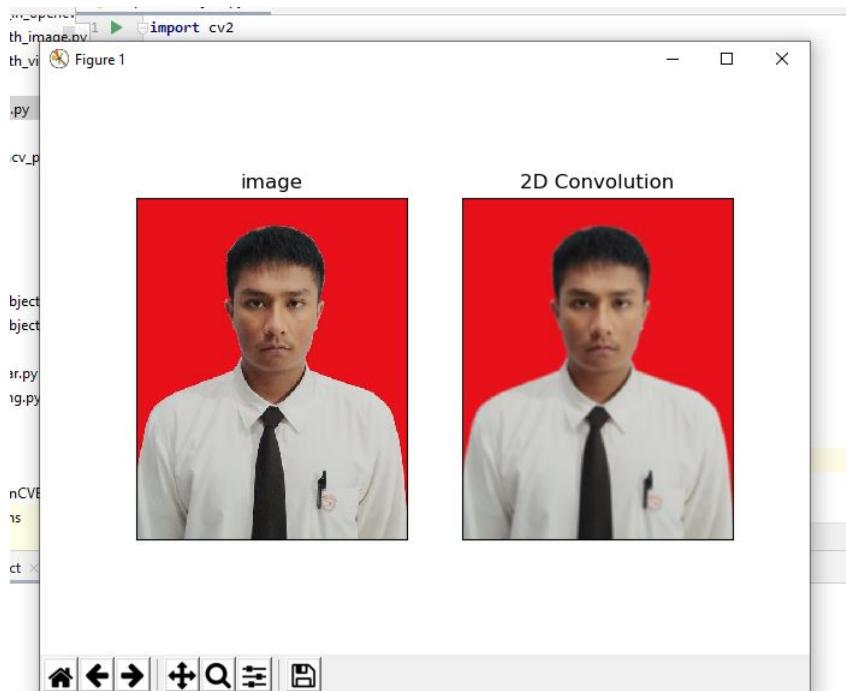
## 2.15 Smoothing Images

### 2.15.1 Menggunakan Filter 2D

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('najib.jpg')
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12
13 kernel = np.ones((5, 5), np.float32)/25
14
15 dst = cv2.filter2D(img, -1, kernel)
16
17
18 titles = ['image', '2D Convolution']
19
20 images = [img, dst]
21
22
23 for i in range(2):
24
25     plt.subplot(1, 2, i+1), plt.imshow(images[i], 'gray')
26
27     plt.title(titles[i])
28
29     plt.xticks([]), plt.yticks([])
30
31 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. pada gambar yang di ambil digunakan kontras warna bgr blue, green, read
- vi. buat karnel numpy yang berisi matriks per 25
- vii. menggunakan event filter 2D
- viii. membuat judul gambar
- ix. memanggil gambar
- x. menampilkan 2 gambar pada satu frame
- xi. menampilkan gambar dengan 1 kolom 2 baris menampilkan gambar sesuai urutan gambar
- xii. menampilkan judul sesuai urutan judul

- xiii. tidak menampilkan koordinat matplotlib
- xiv. menampilkan gambar



**Gambar 2.80** Menggunakan Filter 2D

## 2.15.2 Menggunakan Event Blur

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('najib.jpg')
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12
13 kernel = np.ones((5, 5), np.float32)/25
14
15 dst = cv2.filter2D(img, -1, kernel)
16
17 blur = cv2.blur(img, (5, 5))
18
19
20 titles = ['image', '2D Convolution', 'blur']
21
22 images = [img, dst, blur]
23
24
25 for i in range(3):
26
27     plt.subplot(1, 3, i+1), plt.imshow(images[i], 'gray')
28
29     plt.title(titles[i])
30
31     plt.xticks([]), plt.yticks([])
32
33 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. pada gambar yang di ambil digunakan kontras warna bgr blue, green, read
- vi. buat karnel numpy yang berisi matriks per 25
- vii. menggunakan event filter 2D
- viii. menggunakan event blur
- ix. membuat judul gambar
- x. memanggil gambar
- xi. menampilkan 3 gambar pada satu frame
- xii. menampilkan gambar dengan 1 kolom 3 baris menampilkan gambar sesuai urutan gambar

- xiii. menampilkan judul sesuai urutan judul
- xiv. tidak menampilkan koordinat matplotlib
- xv. menampilkan gambar

Figure 1



**Gambar 2.81** Menggunakan Event Blur

### 2.15.3 Menggunakan Event Gaussian Blur

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('najib.jpg')
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12
13 kernel = np.ones((5, 5), np.float32)/25
14
15 dst = cv2.filter2D(img, -1, kernel)
16
17 blur = cv2.blur(img, (5, 5))
18
19 gblur = cv2.GaussianBlur(img, (5, 5), 0)
20
21
22 titles = ['image', '2D Convolution', 'blur', 'GaussianBlur']
23
24 images = [img, dst, blur, gblur]
25
26
27 for i in range(4):
28
29     plt.subplot(2, 2, i+1), plt.imshow(images[i], 'gray')
30
31     plt.title(titles[i])
32
33     plt.xticks([]), plt.yticks([])
34
35 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. pada gambar yang di ambil digunakan kontras warna bgr blue, green, read
- vi. buat karnel numpy yang berisi matriks per 25
- vii. menggunakan event filter 2D
- viii. menggunakan event blur
- ix. menggunakan event Gaussian Blur event ini tidak terlalu ngeblur di banding event blur
- x. membuat judul gambar
- xi. memanggil gambar

- xii. menampilkan 4 gambar pada satu frame
- xiii. menampilkan gambar dengan 2 kolom 2 baris menampilkan gambar sesuai urutan gambar
- xiv. menampilkan judul sesuai urutan judul
- xv. tidak menampilkan koordinat matplotlib
- xvi. menampilkan gambar



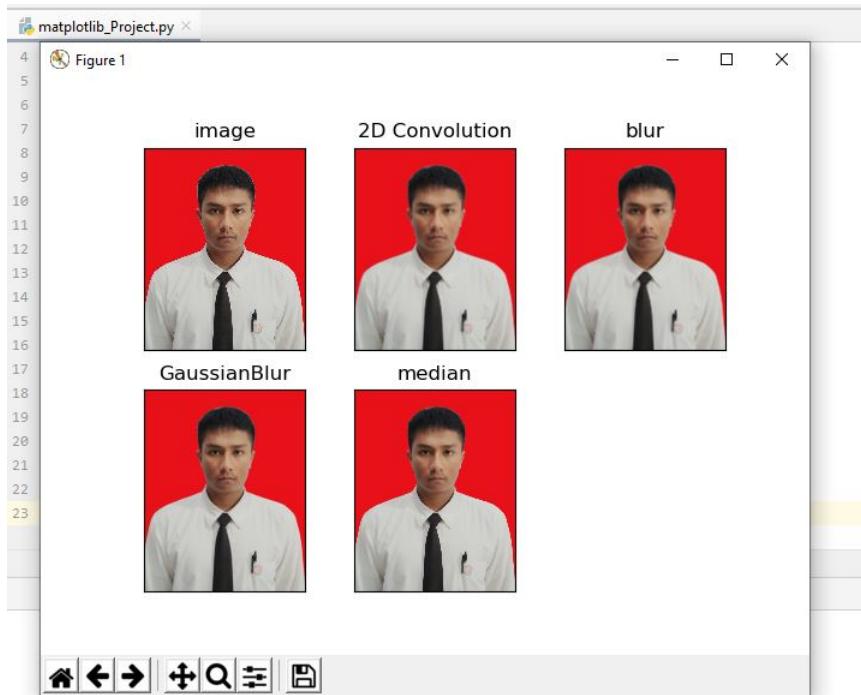
**Gambar 2.82** Menggunakan Event Gaussian Blur

## 2.15.4 Menggunakan Event Median Blur

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('najib.jpg')
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12
13 kernel = np.ones((5, 5), np.float32)/25
14
15 dst = cv2.filter2D(img, -1, kernel)
16
17 blur = cv2.blur(img, (5, 5))
18
19 gblur = cv2.GaussianBlur(img, (5, 5), 0)
20
21 median = cv2.medianBlur(img, 5)
22
23
24 titles = ['image', '2D Convolution', 'blur', 'GaussianBlur', 'median']
25
26 images = [img, dst, blur, gblur, median]
27
28
29 for i in range(5):
30     plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
31
32     plt.title(titles[i])
33
34     plt.xticks([]), plt.yticks([])
35
36 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. pada gambar yang di ambil digunakan kontras warna bgr blue, green, read
- vi. buat karnel numpy yang berisi matriks per 25
- vii. menggunakan event filter 2D
- viii. menggunakan event blur
- ix. menggunakan event Gaussian Blur event ini tidak terlalu ngeblur di banding event blur

- x. menggunakan event median blur
- xi. membuat judul gambar
- xii. memanggil gambar
- xiii. menampilkan 5 gambar pada satu frame
- xiv. menampilkan gambar dengan 2 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xv. menampilkan judul sesuai urutan judul
- xvi. tidak menampilkan koordinat matplotlib
- xvii. menampilkan gambar



**Gambar 2.83** Menggunakan Event Median Blur

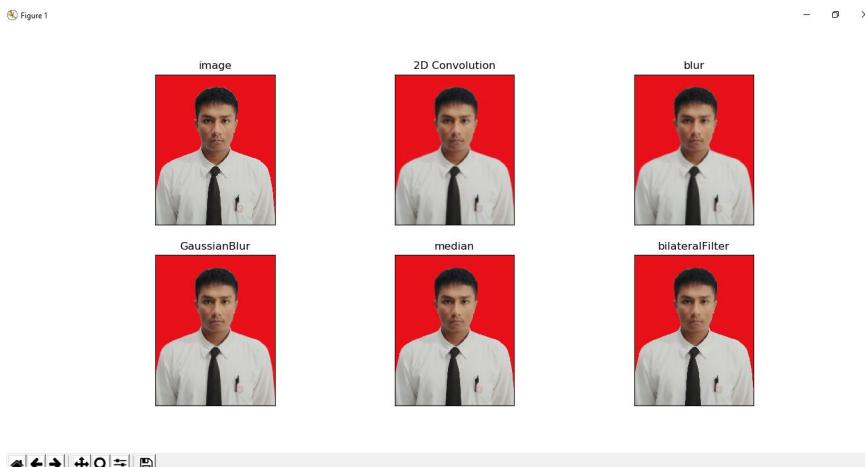
## 2.15.5 Menggunakan Event BilateralFilter

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread('najib.jpg')
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12
13 kernel = np.ones((5, 5), np.float32)/25
14
15 dst = cv2.filter2D(img, -1, kernel)
16
17 blur = cv2.blur(img, (5, 5))
18
19 gblur = cv2.GaussianBlur(img, (5, 5), 0)
20
21 median = cv2.medianBlur(img, 5)
22
23 bilateralFilter = cv2.bilateralFilter(img, 9, 75, 75)
24
25
26 titles = ['image', '2D Convolution', 'blur', 'GaussianBlur', 'median', 'bilateralFilter']
27
28 images = [img, dst, blur, gblur, median, bilateralFilter]
29
30
31 for i in range(6):
32     plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
33
34     plt.title(titles[i])
35
36     plt.xticks([]), plt.yticks([])
37
38 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. pada gambar yang di ambil digunakan kontras warna bgr blue, green, read
- vi. buat karnel numpy yang berisi matriks per 25
- vii. menggunakan event filter 2D
- viii. menggunakan event blur

- ix. menggunakan event Gaussian Blur event ini tidak terlalu ngeblur di banding event blur
- x. menggunakan event median blur
- xi. menggunakan event Bilateral Filter
- xii. membuat judul gambar
- xiii. memanggil gambar
- xiv. menampilkan 5 gambar pada satu frame
- xv. menampilkan gambar dengan 2 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xvi. menampilkan judul sesuai urutan judul
- xvii. tidak menampilkan koordinat matplotlib
- xviii. menampilkan gambar

Figure 1



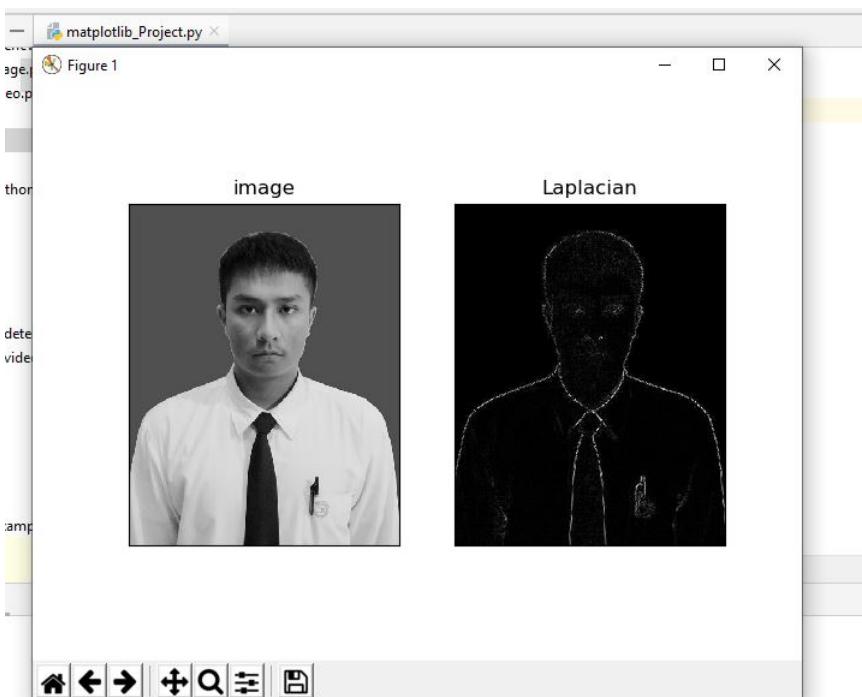
**Gambar 2.84** Menggunakan Event BilateralFilter

## 2.16 Image Gradients

### 2.16.1 Menggunakan Event Laplacian

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread("najib.jpg", cv2.IMREAD_GRAYSCALE)
9
10 lap = cv2.Laplacian(img, cv2.CV_64F)
11
12 lap = np.uint8(np.absolute(lap))
13
14
15 titles = ['image', 'Laplacian']
16
17 images = [img, lap]
18
19
20 for i in range(2):
21     plt.subplot(1, 2, i+1), plt.imshow(images[i], 'gray')
22     plt.title(titles[i])
23
24     plt.xticks([]), plt.yticks([])
25
26 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. menggunakan event laplacian
- vi. menggunakan numpay absolute
- vii. membuat judul gambar
- viii. memanggil gambar
- ix. menampilkan 2 gambar pada satu frame
- x. menampilkan gambar dengan 1 kolom 2 baris menampilkan gambar sesuai urutan gambar
- xi. menampilkan judul sesuai urutan judul
- xii. tidak menampilkan koordinat matplotlib
- xiii. menampilkan gambar

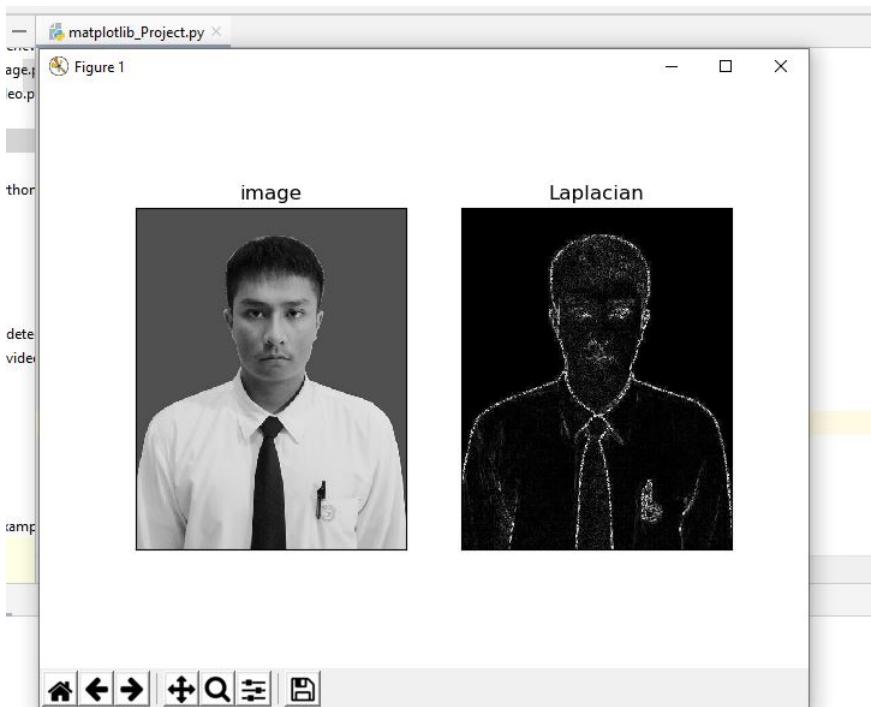


**Gambar 2.85** Menggunakan Event Laplacian

## 2.16.2 Menggunakan Event Laplacian menggunakan Size

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread("najib.jpg", cv2.IMREAD_GRAYSCALE)
9
10 lap = cv2.Laplacian(img, cv2.CV_64F, ksize=3)
11
12 lap = np.uint8(np.absolute(lap))
13
14
15 titles = ['image', 'Laplacian']
16
17 images = [img, lap]
18
19
20 for i in range(2):
21
22     plt.subplot(1, 2, i+1), plt.imshow(images[i], 'gray')
23
24     plt.title(titles[i])
25
26     plt.xticks([]), plt.yticks([])
27
28 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. menggunakan event laplacian gunakan size untuk lebih memperjelas kegunaan laplacian, setelah di tambahkan size gambar berubah menjadi lebih jelas
- vi. menggunakan numpay absolute
- vii. membuat judul gambar
- viii. memanggil gambar
- ix. menampilkan 2 gambar pada satu frame
- x. menampilkan gambar dengan 1 kolom 2 baris menampilkan gambar sesuai urutan gambar
- xi. menampilkan judul sesuai urutan judul
- xii. tidak menampilkan koordinat matplotlib
- xiii. menampilkan gambar



**Gambar 2.86** Menggunakan Event Laplacian menggunakan Size

### 2.16.3 Menggunakan Event Sobel X

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread("najib.jpg", cv2.IMREAD_GRAYSCALE)
9
10 lap = cv2.Laplacian(img, cv2.CV_64F, ksize=3)
11
12 lap = np.uint8(np.absolute(lap))
13
14 sobelX = cv2.Sobel(img, cv2.CV_64F, 1, 0)
15
16
17 sobelX = np.uint8(np.absolute(sobelX))
18
19
20 titles = ['image', 'Laplacian', 'sobelX']
21
22 images = [img, lap, sobelX]
23
24
25 for i in range(3):
26
27     plt.subplot(2, 2, i+1), plt.imshow(images[i], 'gray')
28
29     plt.title(titles[i])
30
31     plt.xticks([]), plt.yticks([])
32
33 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. menggunakan event laplacian gunakan size untuk lebih memperjelas kegunaan laplacian, setelah di tambahkan size gambar berubah menjadi lebih jelas
- vi. menggunakan numpay absolute
- vii. menggunakan event sobel
- viii. menggunakan matriks sobelx agar mengarah ke kanan karena men-garah ke x
- ix. membuat judul gambar
- x. memanggil gambar
- xi. menampilkan 3 gambar pada satu frame

- xii. menampilkan gambar dengan 2 kolom 2 baris menampilkan gambar sesuai urutan gambar
- xiii. menampilkan judul sesuai urutan judul
- xiv. tidak menampilkan koordinat matplotlib
- xv. menampilkan gambar

image



Laplacian



sobelX

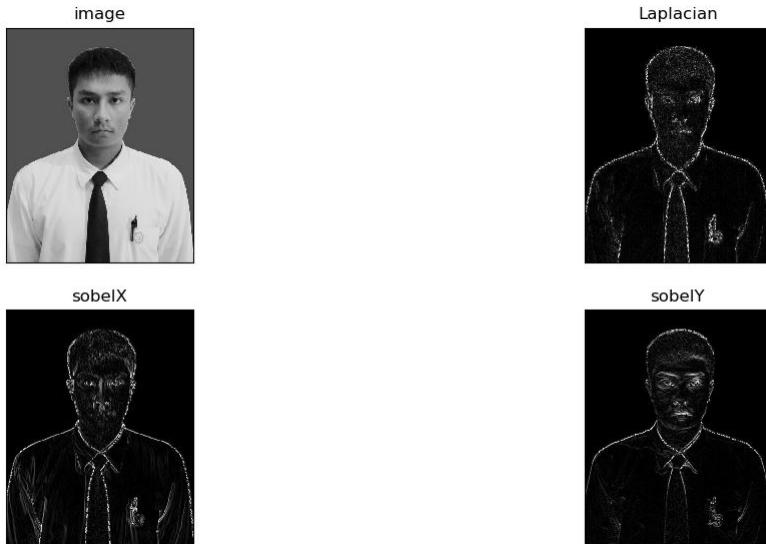
**Gambar 2.87** Menggunakan Event Sobel X

## 2.16.4 Menggunakan Event Sobel Y

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread("najib.jpg", cv2.IMREAD_GRAYSCALE)
9
10 lap = cv2.Laplacian(img, cv2.CV_64F, ksize=3)
11
12 lap = np.uint8(np.absolute(lap))
13
14 sobelX = cv2.Sobel(img, cv2.CV_64F, 1, 0)
15
16 sobelY = cv2.Sobel(img, cv2.CV_64F, 0, 1)
17
18
19 sobelX = np.uint8(np.absolute(sobelX))
20
21 sobelY = np.uint8(np.absolute(sobelY))
22
23
24 titles = ['image', 'Laplacian', 'sobelX', 'sobelY']
25
26 images = [img, lap, sobelX, sobelY]
27
28
29 for i in range(4):
30     plt.subplot(2, 2, i+1), plt.imshow(images[i], 'gray')
31
32     plt.title(titles[i])
33
34     plt.xticks([]), plt.yticks([])
35
36 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. menggunakan event laplacian gunakan size untuk lebih memperjelas kegunaan laplacian, setelah di tambahkan size gambar berubah menjadi lebih jelas
- vi. menggunakan numpay absolute
- vii. menggunakan event sobel beri angka satu pada bagian x
- viii. menggunakan event sobel beri angka satu pada bagian y
- ix. menggunakan matriks sobelx agar mengarah ke kanan dan kiri karena mengarah ke x

- x. menggunakan matriks sobely agar mengarah ke atas dan bawah mengarah ke y
- xi. membuat judul gambar
- xii. memanggil gambar
- xiii. menampilkan 4 gambar pada satu frame
- xiv. menampilkan gambar dengan 2 kolom 2 baris menampilkan gambar sesuai urutan gambar
- xv. menampilkan judul sesuai urutan judul
- xvi. tidak menampilkan koordinat matplotlib
- xvii. menampilkan gambar



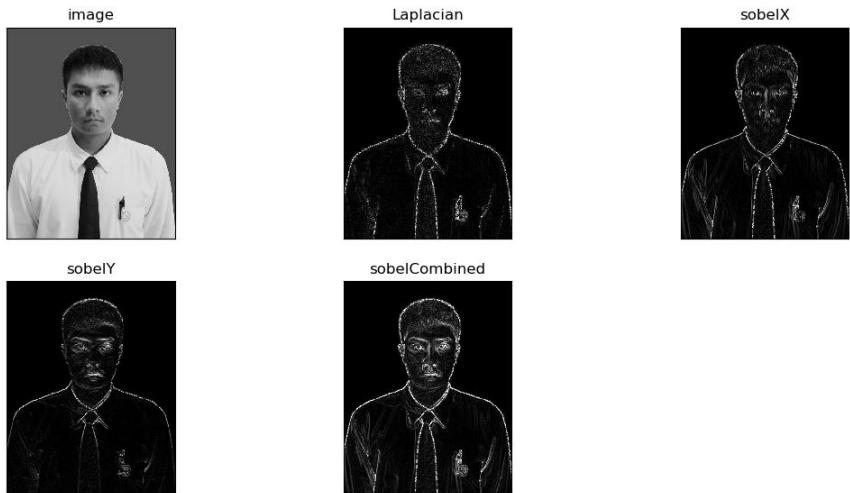
**Gambar 2.88** Menggunakan Event Sobel Y

## 2.16.5 Menggunakan Event Bitwise

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread("najib.jpg", cv2.IMREAD_GRAYSCALE)
9
10 lap = cv2.Laplacian(img, cv2.CV_64F, ksize=3)
11
12 lap = np.uint8(np.absolute(lap))
13
14 sobelX = cv2.Sobel(img, cv2.CV_64F, 1, 0)
15
16 sobelY = cv2.Sobel(img, cv2.CV_64F, 0, 1)
17
18
19 sobelX = np.uint8(np.absolute(sobelX))
20
21 sobelY = np.uint8(np.absolute(sobelY))
22
23
24 sobelCombined = cv2.bitwise_or(sobelX, sobelY)
25
26 titles = ['image', 'Laplacian', 'sobelX', 'sobelY', 'sobelCombined']
27
28 images = [img, lap, sobelX, sobelY, sobelCombined]
29
30
31 for i in range(5):
32
33     plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
34
35     plt.title(titles[i])
36
37     plt.xticks([]), plt.yticks([])
38
39 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. menggunakan event laplacian gunakan size untuk lebih memperjelas kegunaan laplacian, setelah di tambahkan size gambar berubah menjadi lebih jelas
- vi. menggunakan numpy absolute
- vii. menggunakan event sobel beri angka satu pada bagian x
- viii. menggunakan event sobel beri angka satu pada bagian y

- ix. menggunakan matriks sobelx agar mengarah ke kanan dan kiri karena mengarah ke x
- x. menggunakan matriks sobely agar mengarah ke atas dan bawah mengarah ke y
- xi. menggunakan event Bitwise berguna untuk menggabungkan gambar
- xii. membuat judul gambar
- xiii. memanggil gambar
- xiv. menampilkan 5 gambar pada satu frame
- xv. menampilkan gambar dengan 2 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xvi. menampilkan judul sesuai urutan judul
- xvii. tidak menampilkan koordinat matplotlib
- xviii. menampilkan gambar

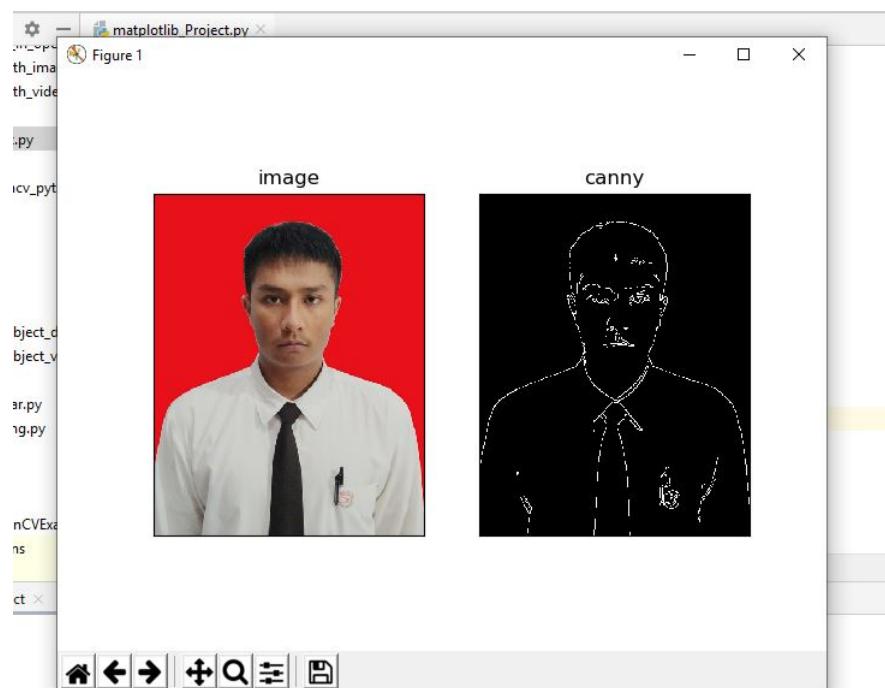


**Gambar 2.89** Menggunakan Event Bitwise

## 2.16.6 Menggunakan Event Canny

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread("najib.jpg")
9
10 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
11
12 canny = cv2.Canny(img, 100, 200)
13
14
15 titles = ['image', 'canny']
16
17 images = [img, canny]
18
19
20 for i in range(2):
21
22     plt.subplot(1, 2, i+1), plt.imshow(images[i], 'gray')
23
24     plt.title(titles[i])
25
26     plt.xticks([]),plt.yticks([])
27
28 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. menggunakan event canny
- vi. membuat judul gambar
- vii. memanggil gambar
- viii. menampilkan 2 gambar pada satu frame
- ix. menampilkan gambar dengan 1 kolom 2 baris menampilkan gambar sesuai urutan gambar
- x. menampilkan judul sesuai urutan judul
- xi. tidak menampilkan koordinat matplotlib
- xii. menampilkan gambar



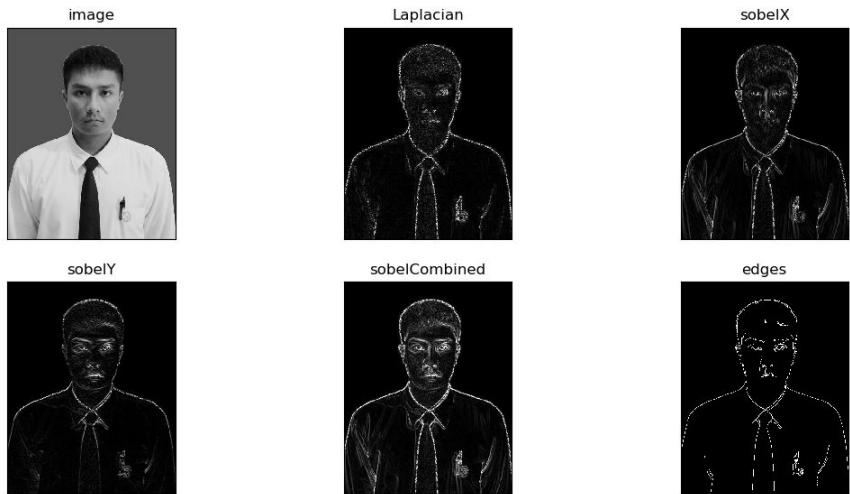
**Gambar 2.90** Menggunakan Event Canny

## 2.16.7 Menggabungkan Event Canny

```
1 import cv2
2
3 import numpy as np
4
5 from matplotlib import pyplot as plt
6
7
8 img = cv2.imread("najib.jpg", cv2.IMREAD_GRAYSCALE)
9
10 lap = cv2.Laplacian(img, cv2.CV_64F, ksize=3)
11
12 lap = np.uint8(np.absolute(lap))
13
14 sobelX = cv2.Sobel(img, cv2.CV_64F, 1, 0)
15
16 sobelY = cv2.Sobel(img, cv2.CV_64F, 0, 1)
17
18 edges = cv2.Canny(img, 100, 200)
19
20
21 sobelX = np.uint8(np.absolute(sobelX))
22
23 sobelY = np.uint8(np.absolute(sobelY))
24
25
26 sobelCombined = cv2.bitwise_or(sobelX, sobelY)
27
28 titles = ['image', 'Laplacian', 'sobelX', 'sobelY', 'sobelCombined', 'edges']
29
30 images = [img, lap, sobelX, sobelY, sobelCombined, edges]
31
32
33 for i in range(6):
34
35     plt.subplot(2, 3, i+1), plt.imshow(images[i], 'gray')
36
37     plt.title(titles[i])
38
39     plt.xticks([]), plt.yticks([])
40
41 plt.show()
```

- i. import cv2
- ii. import numpy
- iii. import matplotlib
- iv. mengambil gambar dari file
- v. menggunakan event laplacian gunakan size untuk lebih memperjelas kegunaan laplacian, setelah di tambahkan size gambar berubah menjadi lebih jelas
- vi. menggunakan numpay absolute

- vii. menggunakan event sobel ber angka satu pada bagian x
- viii. menggunakan event sobel ber angka satu pada bagian y
- ix. menggunakan matriks sobelx agar mengarah ke kanan dan kiri karena mengarah ke x
- x. menggunakan matriks sobely agar mengarah ke atas dan bawah mengarah ke y
- xi. menggunakan event Bitwise berguna untuk menggabungkan gambar
- xii. menggunakan event canny
- xiii. membuat judul gambar
- xiv. memanggil gambar
- xv. menampilkan 6 gambar pada satu frame
- xvi. menampilkan gambar dengan 2 kolom 3 baris menampilkan gambar sesuai urutan gambar
- xvii. menampilkan judul sesuai urutan judul
- xviii. tidak menampilkan koordinat matplotlib
- xix. menampilkan gambar



**Gambar 2.91** Menggabungkan Event Canny