

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

| | |
|----------------------------------|-----------|
| 1 Pengenalan OpenCV | 1 |
| 2 Dasar Dasar Code OpenCV | 41 |

DAFTAR ISI

| | |
|------------------------------|-------|
| Daftar Gambar | xiii |
| Daftar Tabel | xvii |
| Foreword | xxi |
| Kata Pengantar | xxiii |
| Acknowledgments | xxv |
| Acronyms | xxvii |
| List of Symbols | xxix |
| Introduction | xxxi |
| <i>Faisal Najib Abdullah</i> | |

| | |
|---------------------------------|----------|
| 1 Pengenalan OpenCV | 1 |
| 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 |
| 2 | Dasar Dasar Code OpenCV | 41 |
| 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 | 44 |
| 2.2 | Menjalankan kamera leptop | 46 |
| 2.2.1 | Menjalankan video kamera leptop | 46 |
| 2.2.2 | Merubah kontras warna pada video | 48 |
| 2.2.3 | Mengetahui ukuran frame yang ditampilkan | 49 |
| 2.2.4 | Menyimpan video | 52 |
| 2.3 | Menggambar Geometric Pada Foto | 55 |
| 2.3.1 | Membuat garis | 55 |
| 2.3.2 | Membuat warna warna pada garis | 57 |
| 2.3.3 | Membuat garis panah | 60 |
| 2.3.4 | Membuat garis kotak | 62 |
| 2.3.5 | Membuat kotak | 64 |
| 2.3.6 | Membuat garis Lingkaran | 66 |
| 2.3.7 | Membuat Text | 68 |
| 2.4 | Frame Numpay | 70 |
| 2.4.1 | Membuat Frame menggunakan Numpay | 70 |
| 2.4.2 | Berubah Ukuran Frame | 72 |
| 2.4.3 | Menampilkan Text pada Frame video | 74 |
| 2.4.4 | Menampilkln waktu pada frame | 76 |
| 2.4.5 | Menampilkan Event | 78 |
| 2.4.6 | Event Mouse klik kiri | 79 |
| 2.4.7 | Event Mouse klik kiri dan kanan | 82 |
| 2.4.8 | Event Mouse klik kiri dan kanan pada gambar yang dipangggil | 84 |
| 2.4.9 | Event Mouse klik kiri membuat titik dan garis | 86 |

| | | |
|--------|-----------------------------------|-----|
| 2.4.10 | Membuat frame warna sesuai klik | 89 |
| 2.4.11 | Membuat frame warna sesuai klik 2 | 92 |
| 2.4.12 | Berubah Ukuran Frame | 94 |
| 2.4.13 | Berubah Ukuran Frame | 95 |
| 2.4.14 | Berubah Ukuran Frame | 96 |
| 2.4.15 | Berubah Ukuran Frame | 97 |
| 2.4.16 | Berubah Ukuran Frame | 99 |
| 2.4.17 | Berubah Ukuran Frame | 100 |
| 2.4.18 | Berubah Ukuran Frame | 102 |
| 2.4.19 | Berubah Ukuran Frame | 104 |
| 2.4.20 | Berubah Ukuran Frame | 106 |
| 2.4.21 | Berubah Ukuran Frame | 107 |
| 2.4.22 | Berubah Ukuran Frame | 109 |
| 2.4.23 | Berubah Ukuran Frame | 112 |

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 | 43 |
| 2.3 | Menyimpan gambar | 45 |
| 2.4 | Menggunakan kamera leptop | 47 |
| 2.5 | Kontras warna video | 49 |
| 2.6 | Ukuran Frame video | 51 |
| 2.7 | Setelah dijalankan file tersimpan | 53 |
| 2.8 | File Sebelum dijalankan | 54 |
| 2.9 | Video yang sudah di simpan | 54 |
| 2.10 | MMembuat garis | 56 |
| 2.11 | Membuat garis | 58 |
| 2.12 | Nomor Nomor warna | 59 |
| 2.13 | Membuat garis panah | 61 |
| 2.14 | Membuat garis kotak | 63 |
| 2.15 | Membuat kotak | 65 |
| 2.16 | Membuat garis lingkaran | 67 |
| 2.17 | Membuat Text | 69 |
| 2.18 | Membuat Frame Numpy | 71 |
| 2.19 | Merubah ukuran frame | 73 |
| 2.20 | Menampilkan Text pada Frame video | 75 |

| | | |
|------|------------------------------------------------------------|-----|
| 2.21 | Menampilkan waktu pada frame | 77 |
| 2.22 | Menampilkan Event | 78 |
| 2.23 | Event Mouse klik kiri | 80 |
| 2.24 | Event Mouse klik kiri | 80 |
| 2.25 | Event Mouse klik kiri | 81 |
| 2.26 | Event Mouse klik kiri dan kanan | 83 |
| 2.27 | Event Mouse klik kiri dan kanan pada gambar yang dipanggil | 85 |
| 2.28 | Event Mouse klik kiri membuat titik dan garis | 87 |
| 2.29 | Event Mouse klik kiri membuat titik dan garis | 87 |
| 2.30 | Event Mouse klik kiri membuat titik dan garis | 88 |
| 2.31 | Membuat frame warna sesuai klik | 90 |
| 2.32 | Membuat frame warna sesuai klik | 91 |
| 2.33 | Membuat frame warna sesuai klik | 91 |
| 2.34 | Membuat frame warna sesuai klik 2 | 93 |
| 2.35 | Membuat Frame Numpy | 94 |
| 2.36 | Membuat Frame Numpy | 95 |
| 2.37 | Membuat Frame Numpy | 97 |
| 2.38 | Membuat Frame Numpy | 98 |
| 2.39 | Membuat Frame Numpy | 100 |
| 2.40 | Membuat Frame Numpy | 101 |
| 2.41 | Membuat Frame Numpy | 102 |
| 2.42 | Membuat Frame Numpy | 103 |
| 2.43 | Membuat Frame Numpy | 104 |
| 2.44 | Membuat Frame Numpy | 105 |
| 2.45 | Membuat Frame Numpy | 106 |
| 2.46 | Membuat Frame Numpy | 107 |
| 2.47 | Membuat Frame Numpy | 108 |
| 2.48 | Membuat Frame Numpy | 109 |

| | | |
|------|---------------------|-----|
| 2.49 | Membuat Frame Numpy | 110 |
| 2.50 | Membuat Frame Numpy | 111 |
| 2.51 | Membuat Frame Numpy | 112 |
| 2.52 | Membuat Frame Numpy | 113 |
| 2.53 | Membuat Frame Numpy | 114 |
| 2.54 | Membuat Frame Numpy | 115 |

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 | 44 |
| src/cv4.py | 46 |
| src/cv5.py | 48 |
| src/cv6.py | 49 |
| src/cv7.py | 52 |
| src/cv8.py | 55 |
| src/cv9.py | 57 |
| src/cv10.py | 60 |
| src/cv11.py | 62 |
| src/cv12.py | 64 |
| src/cv13.py | 66 |

| | |
|-------------|-----|
| src/cv14.py | 68 |
| src/cv15.py | 70 |
| src/cv16.py | 72 |
| src/cv17.py | 74 |
| src/cv18.py | 76 |
| src/cv19.py | 78 |
| src/cv20.py | 79 |
| src/cv21.py | 82 |
| src/cv22.py | 84 |
| src/cv23.py | 86 |
| src/cv24.py | 89 |
| src/cv25.py | 92 |
| src/cv26.py | 94 |
| src/cv27.py | 95 |
| src/cv28.py | 96 |
| src/cv29.py | 97 |
| src/cv30.py | 99 |
| src/cv31.py | 100 |
| src/cv32.py | 102 |
| src/cv33.py | 104 |
| src/cv34.py | 106 |
| src/cv35.py | 107 |
| src/cv36.py | 109 |
| src/cv37.py | 112 |

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.

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

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 berkonsultasi 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>

| File | Last Modified | Size |
|-------------------------------------------------------|-------------------|----------|
| .. / | | |
| amd64/ | | |
| amd64rc1/ | 20-Oct-2018 15:28 | - |
| amd64rc2/ | 26-Sep-2018 22:44 | - |
| win32/ | 13-Oct-2018 16:41 | - |
| win32rc1/ | 20-Oct-2018 15:28 | - |
| win32rc2/ | 26-Sep-2018 22:45 | - |
| Python-3.7.1.tar.xz | 13-Oct-2018 16:42 | - |
| Python-3.7.1.tar.xz.asc | 20-Oct-2018 06:25 | 16960060 |
| Python-3.7.1.tgz | 20-Oct-2018 06:26 | 833 |
| Python-3.7.1.tgz.asc | 20-Oct-2018 06:25 | 22802018 |
| Python-3.7.1.lrc1.tar.xz | 20-Oct-2018 06:26 | 833 |
| Python-3.7.1.lrc1.tar.xz.asc | 26-Sep-2018 20:17 | 16968876 |
| Python-3.7.1.lrc1.tar.xz.asc | 26-Sep-2018 20:17 | 833 |
| Python-3.7.1.lrc1.tgz | 26-Sep-2018 20:16 | 22800318 |
| Python-3.7.1.lrc1.tgz.asc | 26-Sep-2018 20:17 | 833 |
| Python-3.7.1.lrc2.tar.xz | 13-Oct-2018 07:24 | 16974832 |
| Python-3.7.1.lrc2.tar.xz.asc | 13-Oct-2018 07:24 | 833 |
| Python-3.7.1.lrc2.tgz | 13-Oct-2018 07:23 | 22805187 |
| Python-3.7.1.lrc2.tgz.asc | 13-Oct-2018 07:24 | 833 |
| python-3.7.1-amd64-webinstall.exe | 20-Oct-2018 15:27 | 1333960 |
| python-3.7.1-amd64-webinstall.exe.asc | 20-Oct-2018 15:27 | 836 |
| python-3.7.1-amd64.exe | 20-Oct-2018 15:27 | 26260496 |
| python-3.7.1-amd64.exe.asc | 20-Oct-2018 15:27 | 836 |
| python-3.7.1-embed-amd64.zip | 20-Oct-2018 15:27 | 6879900 |
| python-3.7.1-embed-amd64.zip.asc | 20-Oct-2018 15:27 | 836 |
| python-3.7.1-embed-win32.zip | 20-Oct-2018 15:28 | 6377805 |
| python-3.7.1-embed-win32.zip.asc | 20-Oct-2018 15:28 | 836 |
| python-3.7.1-macosx10.6.pkg | 20-Oct-2018 07:55 | 34360623 |
| python-3.7.1-macosx10.6.pkg.asc | 20-Oct-2018 07:55 | 833 |
| python-3.7.1-macosx10.9.pkg | 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  <http://sourceforge.net/projects/numpy/files/NumPy/1.6.2/numpy-1.6.2-win32-superpack-python2.7.exe>

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 like 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).

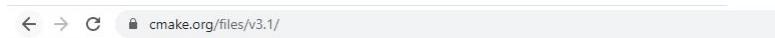
Menggunakan CMake dan kompiler

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

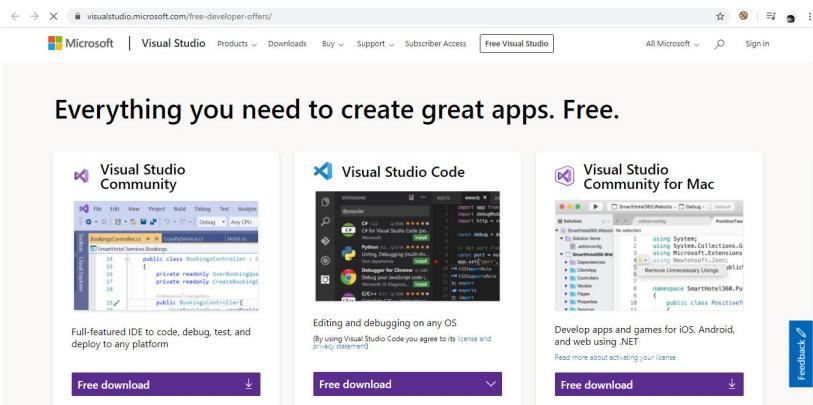
| Name | Last modified | Size | Description |
|-------------------------------------------------------|------------------|------|-------------|
| Parent Directory | | - | |
| cmake-3.1.0-1-src.tar.bz2 | 2014-12-17 13:10 | 4.5M | |
| cmake-3.1.0-1.tar.bz2 | 2014-12-17 13:10 | 9.0M | |
| cmake-3.1.0-Darwin-universal.dmg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

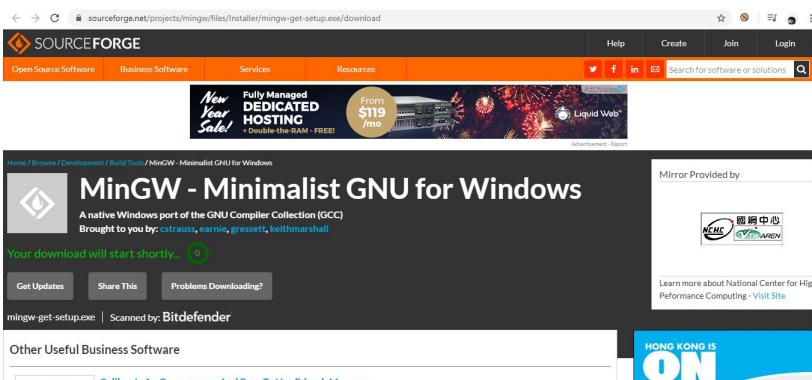


Gambar 1.5 Cari VisualStudio Terbaru pada website ini kemudian install

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

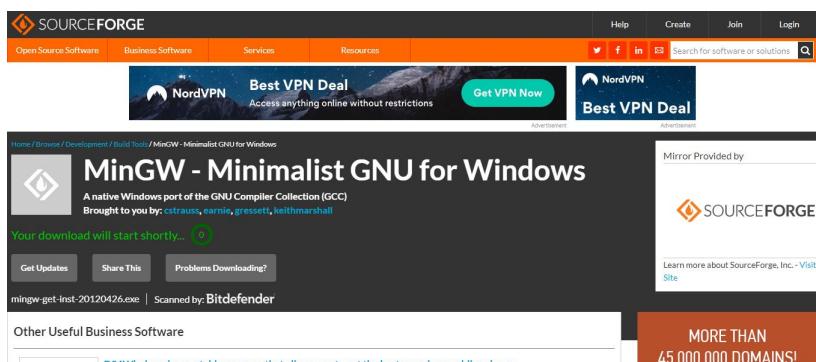
<http://sourceforge.net/projects/mingw/files/Installer/mingw->



Gambar 1.6 Download MinGW

dan

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



Gambar 1.7 Download MinGW

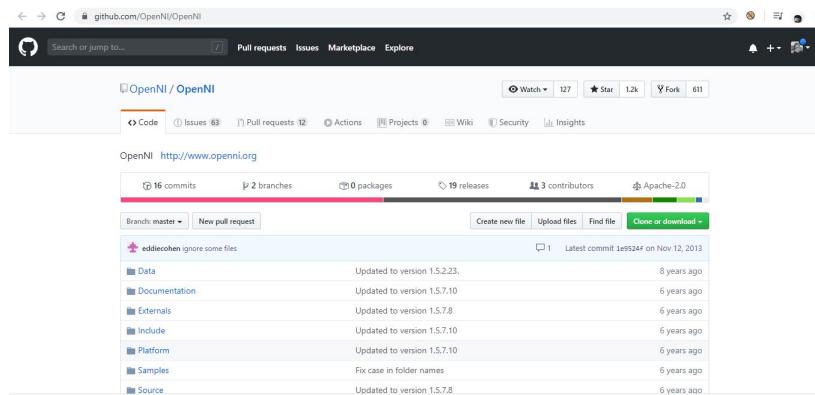
Saat menjalankan penginstal, pastikan jalur tujuan tidak mengandung spasi dan bahwa kompiler C ++ opsional disertakan. Edit variabel PATH sistem dan tambahkan;

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

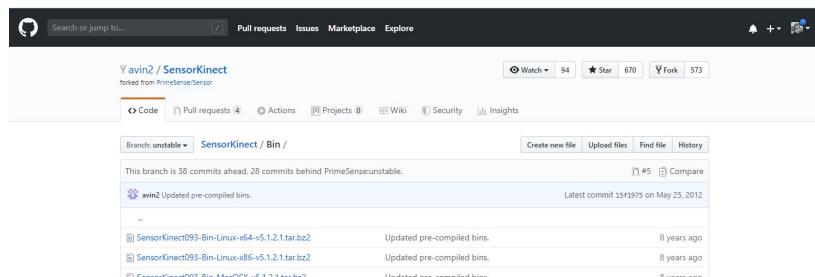
<https://github.com/OpenNI/OpenNI>



Gambar 1.8 Download OpenNI

4. Anda dapat mengunduh dan menginstal SensorKinect 0.93 dari

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

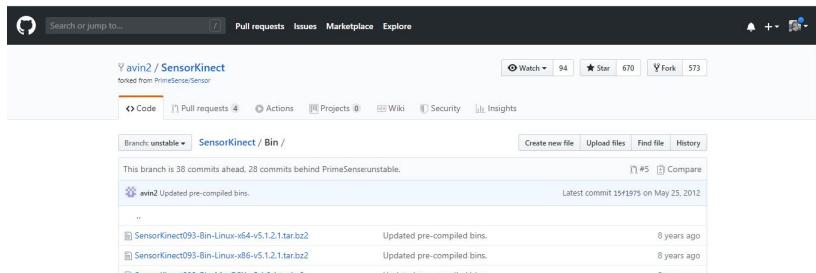


Gambar 1.9 Download SensorKinect

Atau, untuk Python 64-bit, unduh di

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

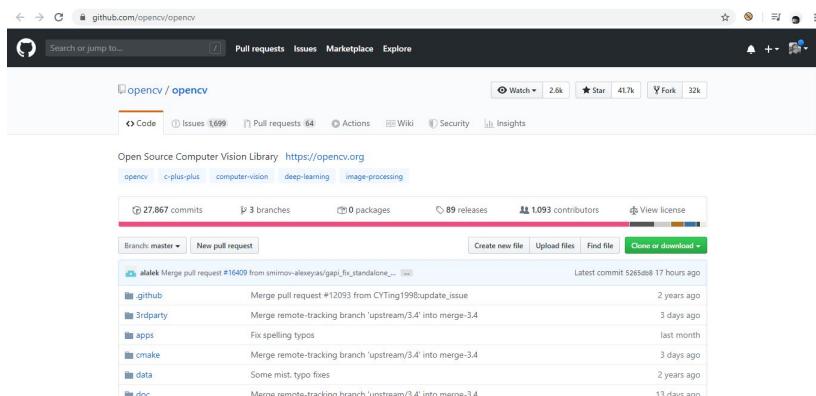
(64-bit). Perhatikan bahwa repositori ini tidak aktif selama lebih dari tiga tahun.



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.

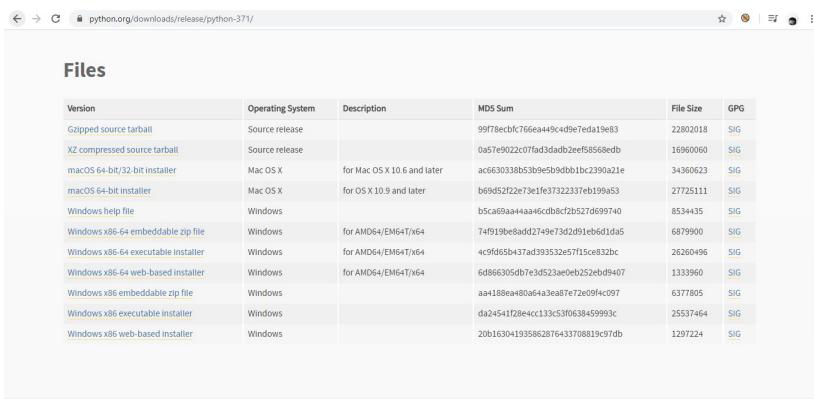
1.3.2 Instalasi pada OS X

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 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 labeled 'Terminal I/O and file text.' containing examples of terminal commands.

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.</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**

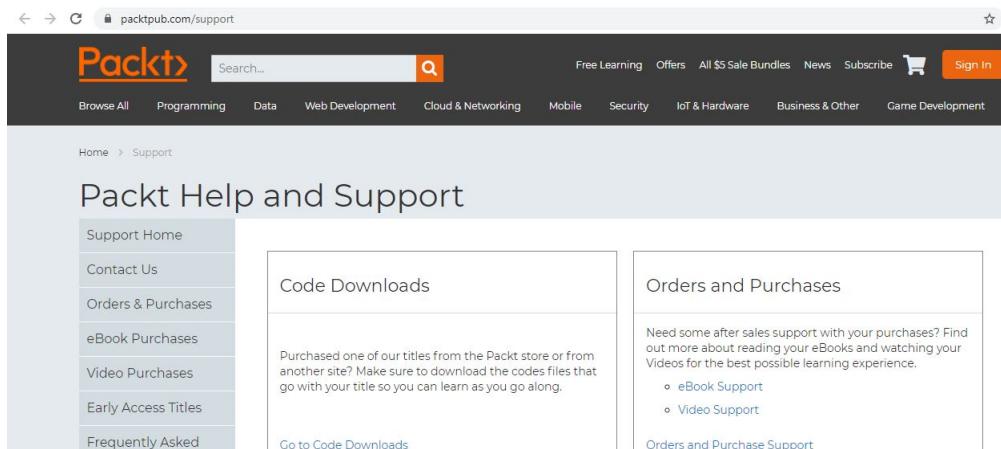
Anda dapat mengunduh file kode contoh untuk semua buku Penerbitan Packt yang telah Anda beli dari akun Anda di

<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.

FreeBSD Handbook

The FreeBSD Documentation Project

Revision: 53811

Copyright © 1995-2020 The FreeBSD Documentation Project

[Copyright](#)

[Legal Notice](#)

Last modified on 2020-01-23 20:41:16 by carlavilla.

Abstract

Welcome to FreeBSD! This handbook covers the installation and day to day use of *FreeBSD 12.1-RELEASE*, *FreeBSD 12.0-RELEASE* and *FreeBSD 11.3-RELEASE*. This book is the result of ongoing work by many individuals. Some sections might be outdated. Those interested in helping to update and expand this document should send email to the [FreeBSD documentation project mailing list](#).

The latest version of this book is available from the [FreeBSD web site](#). Previous versions can be obtained from <https://docs.FreeBSD.org/doc/>. The book can be downloaded in a variety of formats and compression options from the [FreeBSD FTP server](#) or one of the numerous [mirror sites](#). Printed copies can be purchased at the [FreeBSD Mall](#). Searches can be performed on the handbook and other documents on the [search page](#).

Table of Contents

[Preface](#)

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. 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>
```

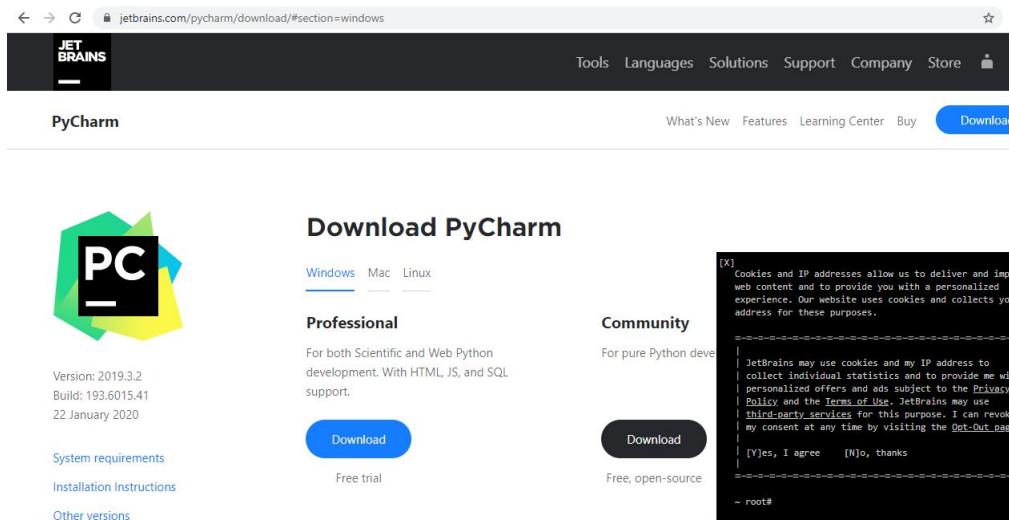
Jadi, jika Anda telah mengikuti prosedur standar dan membuat direktori build di folder unduhan OpenCV Anda, Anda harus menjalankan perintah berikut:

```
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

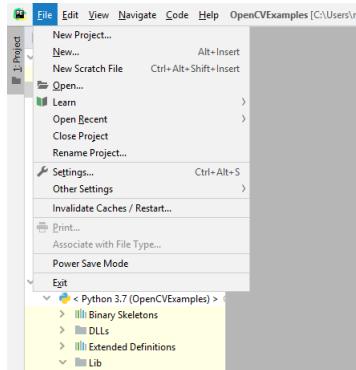
Pycharm merupakan tools untuk menjalankan program python didalamnya sudah terdapat berbagai macam library dari python itu sendiri, kita hanya perlu mencari library yang kita butuhkan kemudian klik install, maka kita tidak perlu melakukan hal hal yang telah di contohkan untuk menginstall opencv pada windows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

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



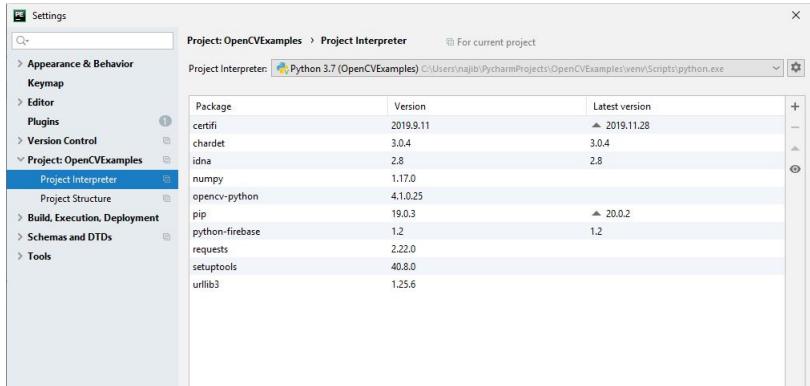
Gambar 1.18 Pycharm

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



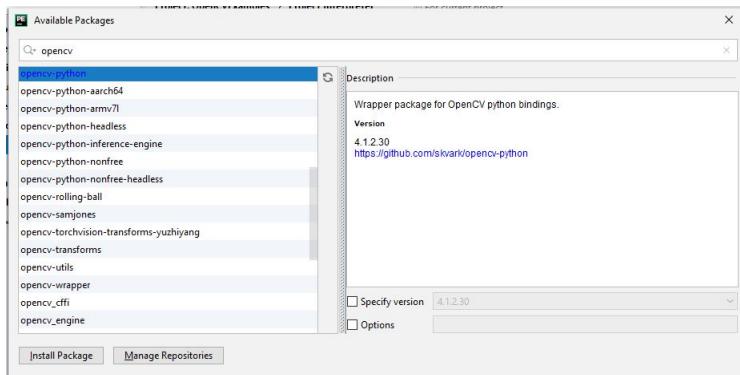
Gambar 1.19 Pycharm Settings

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

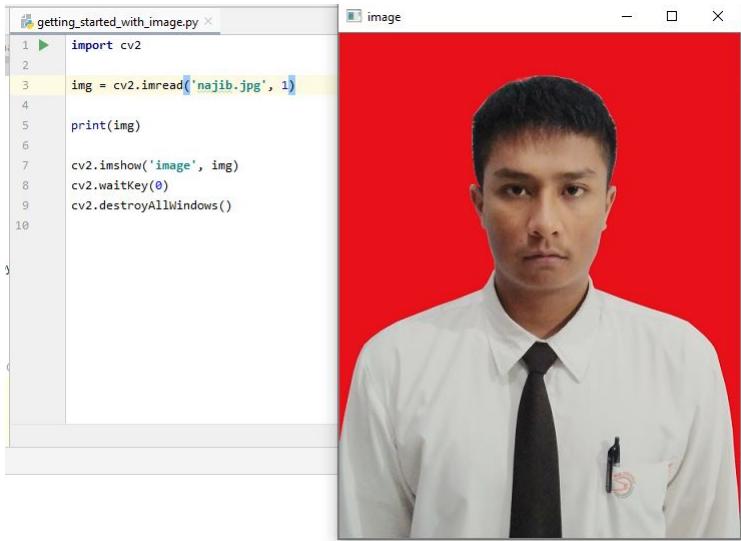
2.1.1 Menampilkan Gambar

```
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.



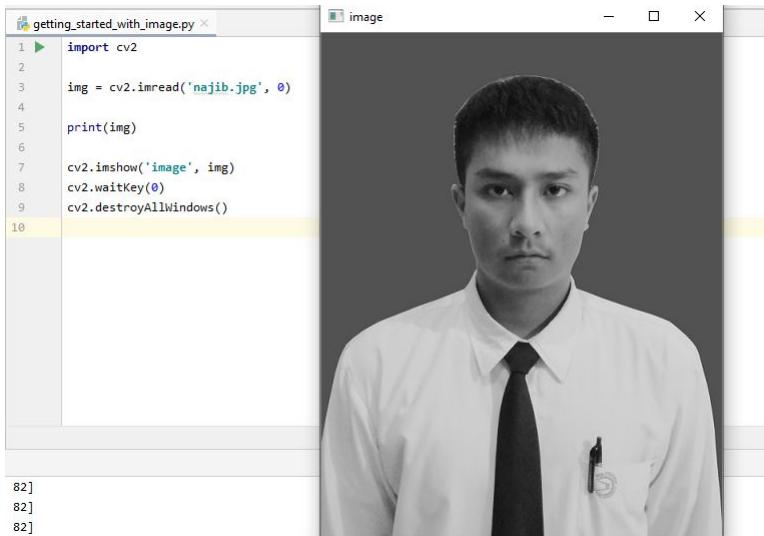
Gambar 2.1 Menampilkan gambar

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 gradiasi 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.



Gambar 2.2 Merubah kontras warna

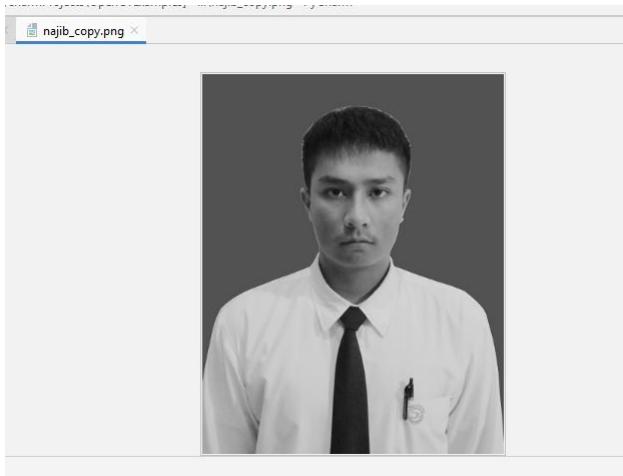
Pada bagian kodingan ini foto di rubah kontras warnanya menjadi hitam putih, pada kodingan sebelumnya yang di rubah hanya satu huruf saja unto

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 img = cv2.imread('najib.jpg', 0)  
3 print(img)  
4 cv2.imshow('image', img)  
5 cv2.waitKey(0)  
6 cv2.destroyAllWindows()  
7 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.



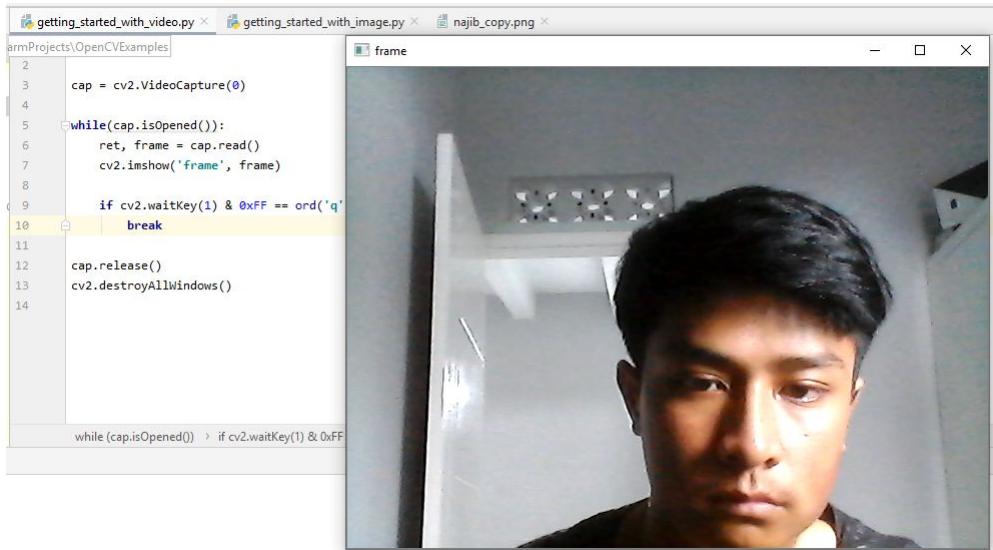
Gambar 2.3 Menyimpan gambar

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 laptop

```
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 laptop.
- 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.



Gambar 2.4 Menggunakan kamera leptop

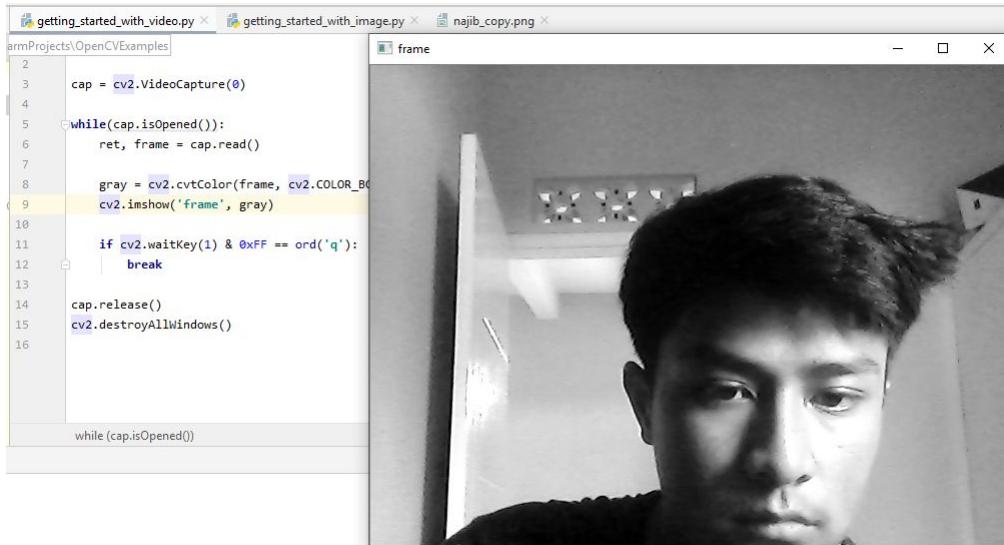
Pada videocapture ini hanya merekam menggunakan kamera leptop saja belum masuk ke pengolahan gambar.

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.

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



Gambar 2.5 Kontras warna video

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 leaptop.

- 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

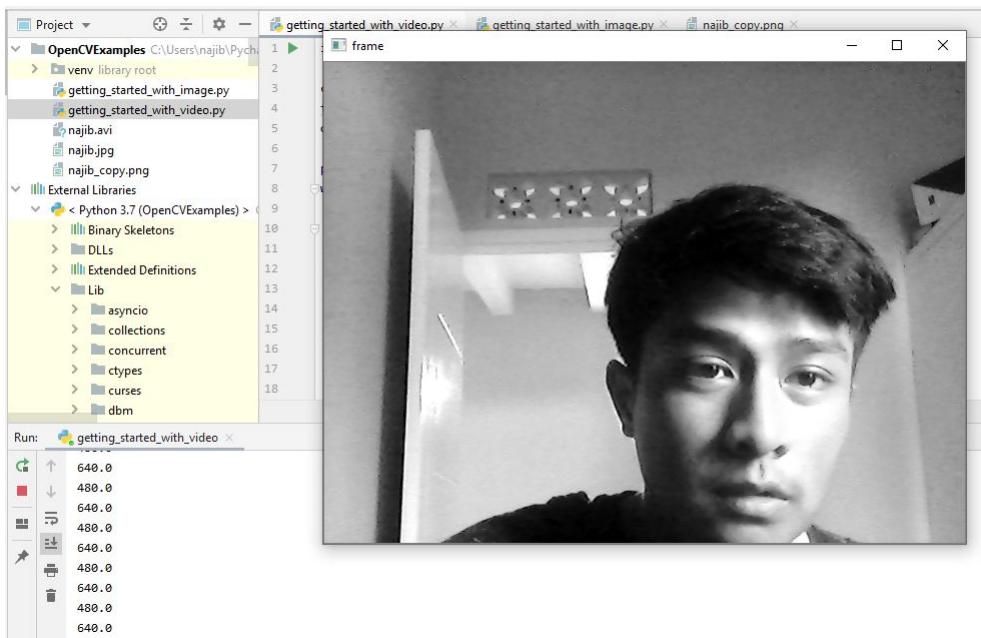
Maka akan ditampilkan secara berulang karna berada pada while dan pada bagian videocapture juga jika tidak di lakukan perulangan maka sekali muncul akan langsung keluar secara otomatis.

2.2.4 Menyimpan video

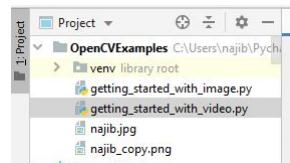
```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 fourcc = cv2.VideoWriter_fourcc(*'XVID')
6
7 out = cv2.VideoWriter('najib.avi', fourcc, 20.0, (640,480))
8
9 print(cap.isOpened())
10
11 while(cap.isOpened()):
12
13     ret, frame = cap.read()
14
15     if ret == True:
16
17         print(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
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

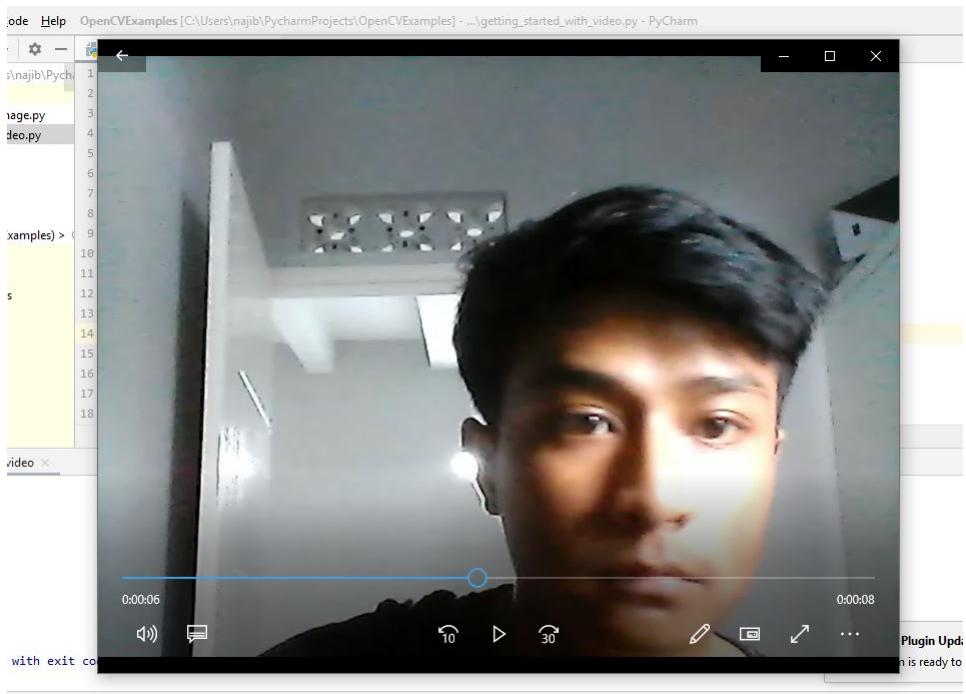
- x. kita cukup print mengambil dari videocapture dan panggil CAP PROP FRAME HEIGHT untuk ukuran tingginya
- xi. buat variable dengan nama gray karna kita mau berubah kontras warnanya menjadi hitam putih, kemudian panggil cvtColor didalam frame dengan warna abu abu.
- xii. kemudian buat frame dengan 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

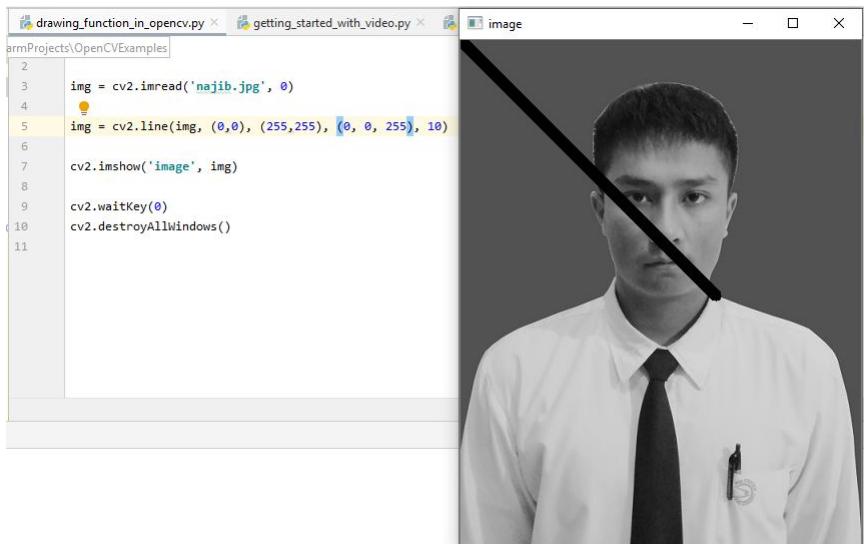
Video tersimpan langsung ke folder yang dituju, video dapat disesuaikan formatnya sesuai yang kita mau.

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 gradasi 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.



The screenshot shows a PyCharm interface with two tabs open: 'drawing_function_in_opencv.py' and 'getting_started_with_video.py'. The code in 'drawing_function_in_opencv.py' is as follows:

```
1  # Importing the required module
2  import cv2
3
4  # Reading the image
5  img = cv2.imread('najib.jpg', 0)
6
7  # Drawing a diagonal line
8  img = cv2.line(img, (0,0), (255,255), (0, 0, 255), 10)
9
10 # Displaying the image
11 cv2.imshow('image', img)
12
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
```

To the right of the code is a window titled 'image' displaying a grayscale portrait of a man. A thick black diagonal line has been drawn across the image from the top-left corner to the bottom-right corner.

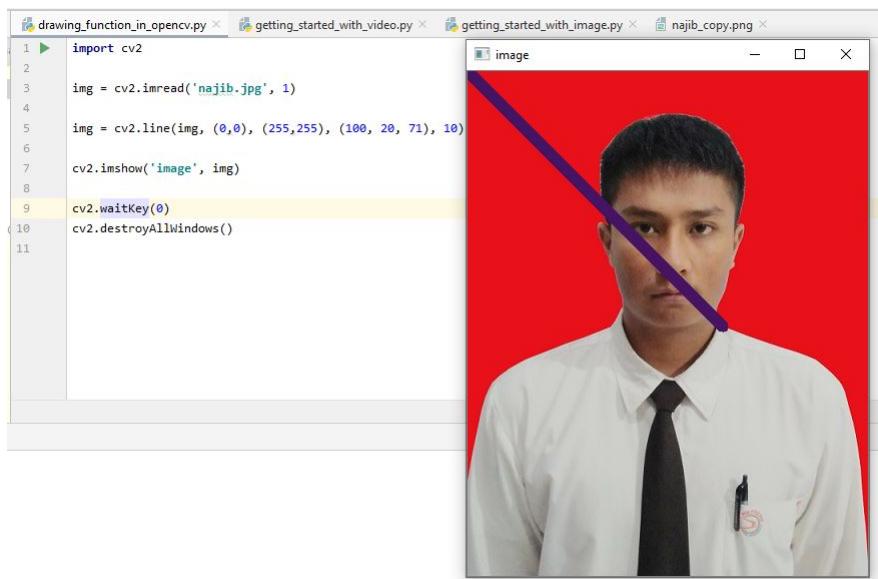
Gambar 2.10 Membuat garis

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()
```

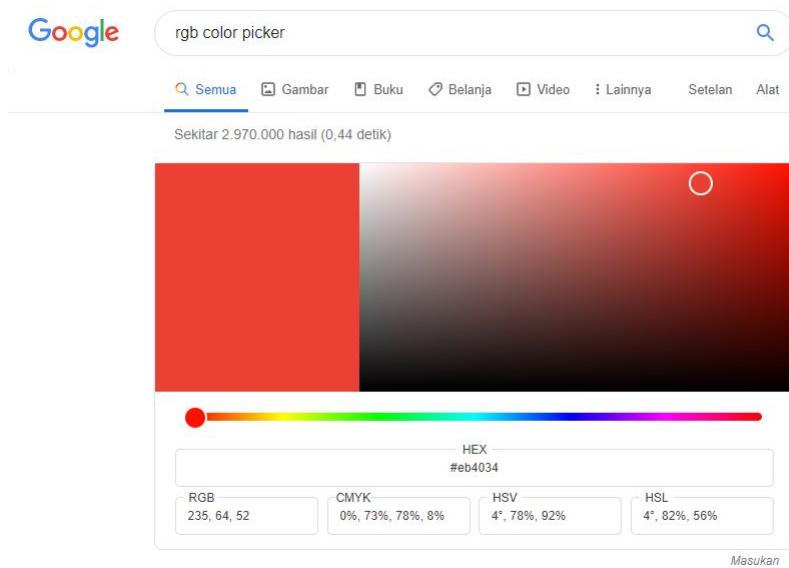
- i. lakukan Import library open cv yaitu cv2
- ii. kemudian panggil file foto menggunakan kode seperti di atas, membuat terlebih dahulu variabel img, kemudian cv2.imread nama file dan nomor untuk gradasi warnanya, pada bagian ini menggunakan angka 1 yang artinya mengikuti foto aslinya.
- iii. kemudian 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.



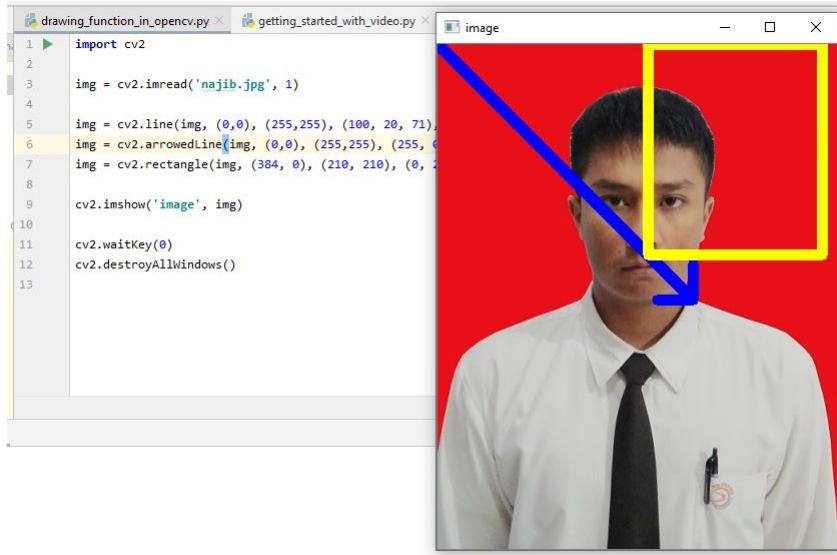
Gambar 2.13 Membuat garis panah

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.



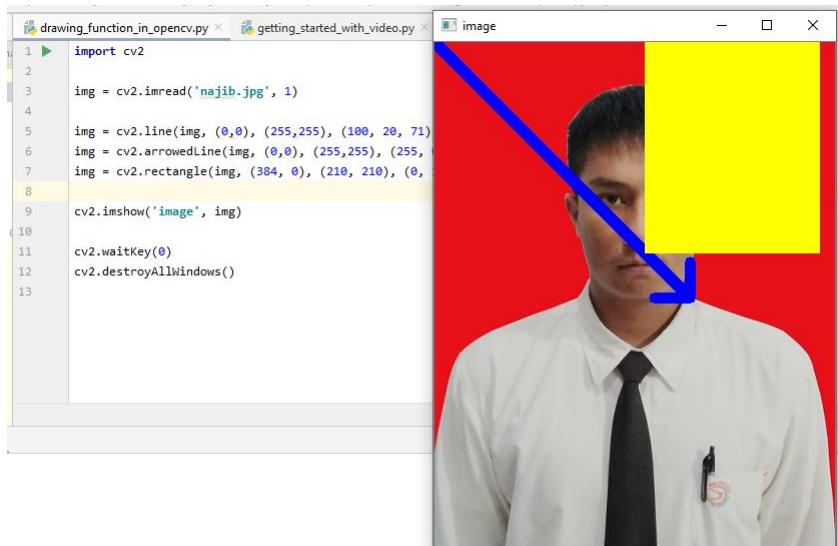
Gambar 2.14 Membuat garis kotak

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.



Gambar 2.15 Membuat kotak

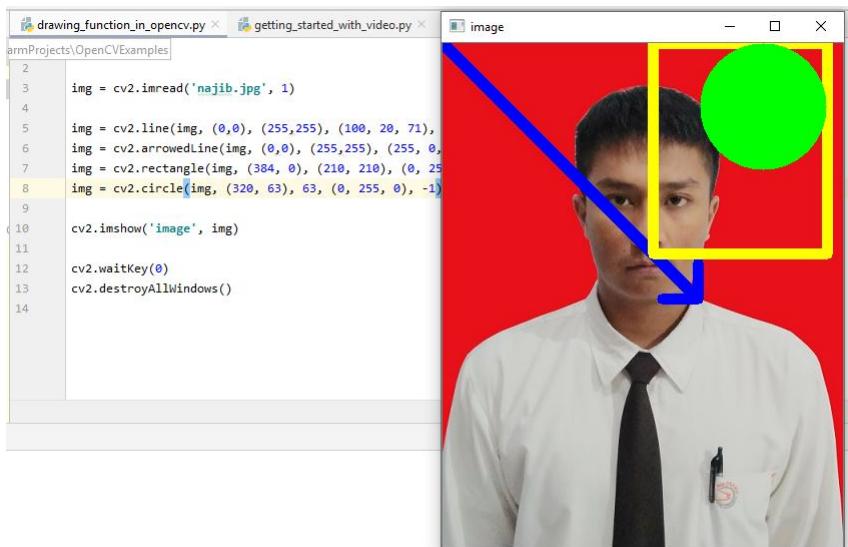
Garis kotak ini bisa kita atur mau warna yang bagaimana, ukuran yang bagaimana, dan posisi yang bagaimana sesuai yang di inginkan dan sesuai kebutuhannya.

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.

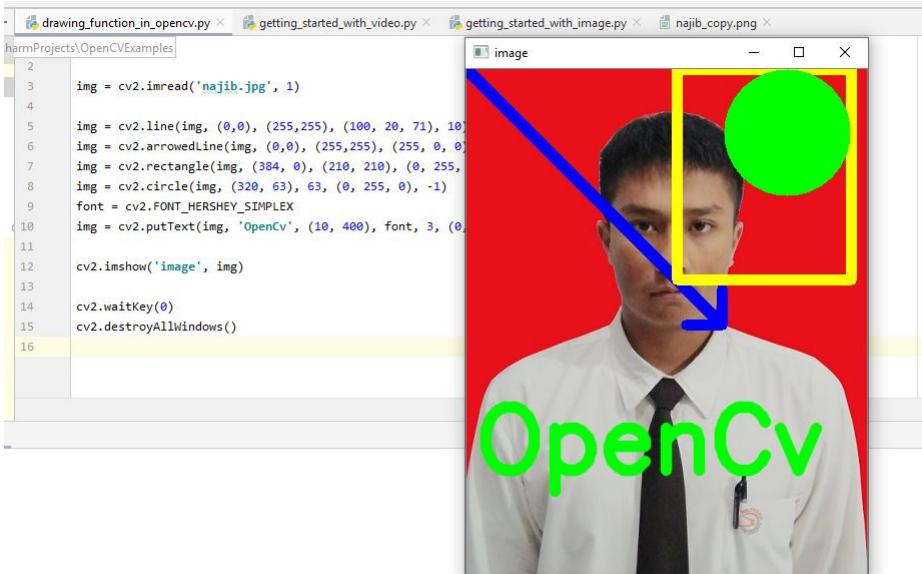
2.3.7 Membuat Text

```
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.



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

2.4.1 Membuat Frame menggunakan 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 img = cv2.circle(img, (320, 63), 63, (0, 255, 0), -1)
16
17 font = cv2.FONT_HERSHEY_SIMPLEX
18
19 img = cv2.putText(img, 'OpenCv', (10, 400), font, 3, (0, 255,
20                      0), 10, cv2.LINE_AA)
21
22 cv2.imshow('image', img)
23
24 cv2.waitKey(0)
25
26 cv2.destroyAllWindows()
```

- i. lakukan import numpay 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()
```

Gambar 2.18 Membuat Frame Numpy

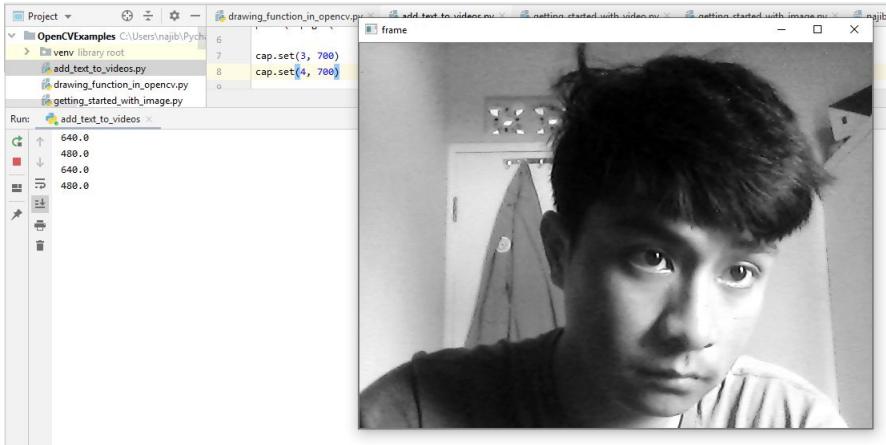
Frame ini dibuat menggunakan matriks yang ada pada library numpy, kita tidak perlu lagi menghitung berapa matriksnya kita cukup gunakan zeros dan ukuran yang di butuhkan.

2.4.2 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.



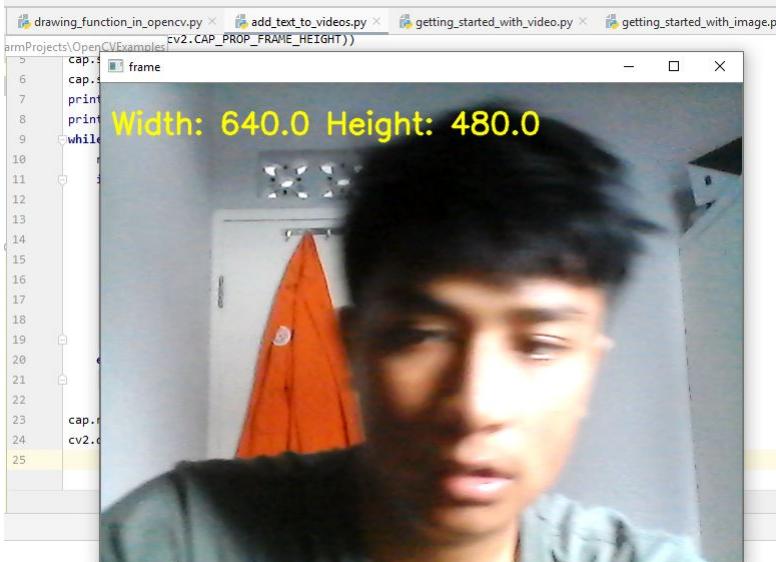
Gambar 2.19 Merubah ukuran frame

2.4.3 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.



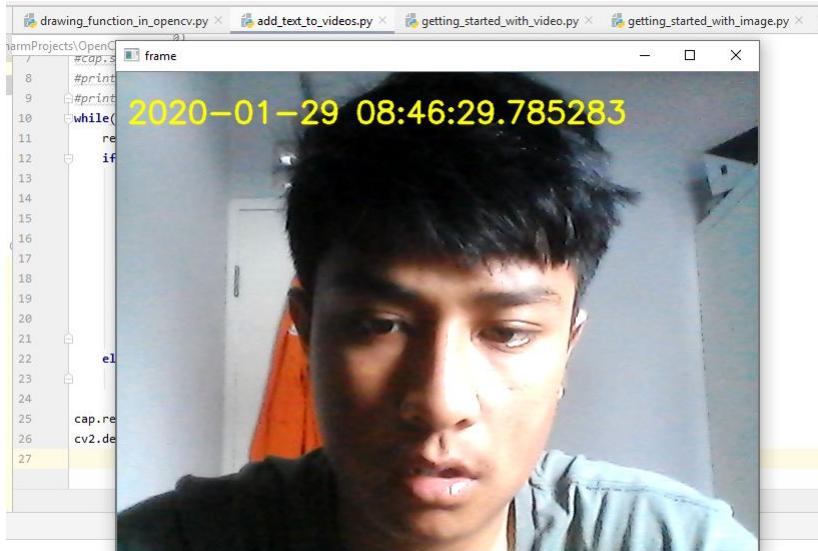
Gambar 2.20 Menampilkan Text pada Frame video

2.4.4 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

2.4.5 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 is the project tree titled 'OpenCVExamples' containing files like 'add_text_to_videos.py', 'drawing_function_in_opencv.py', 'getting_started_with_image.py', and 'getting_started_with_video.py'. The main editor window is titled 'Mouse_even_opencv_python.py' and contains the following Python 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)
```

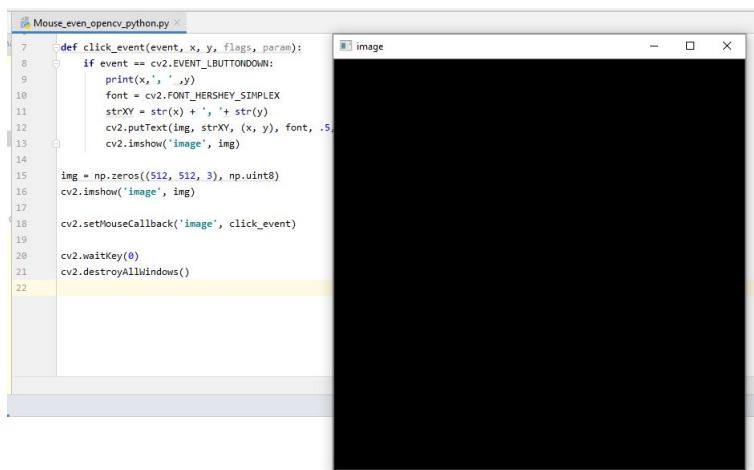
The 'Run' tool window at the bottom shows the output: 'Process finished with exit code 0'. The status bar at the bottom right indicates 'File: 100% 100x100'.

Gambar 2.22 Menampilkan Event

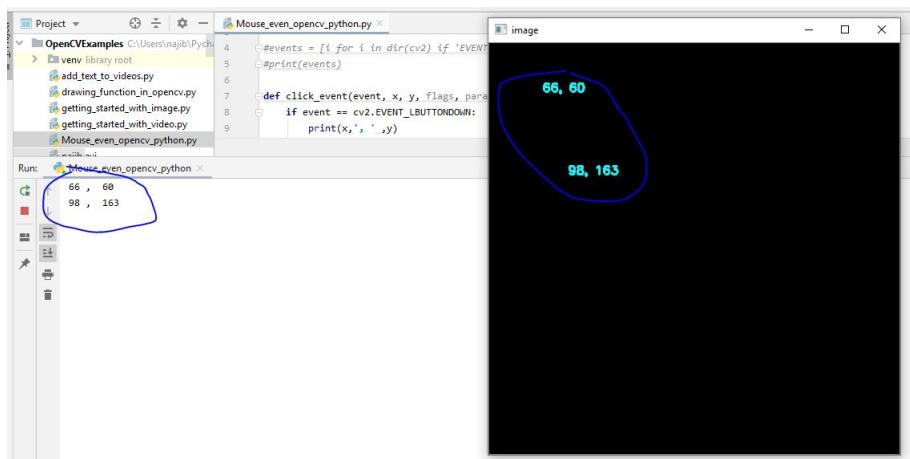
2.4.6 Event Mouse klik kiri

```
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.



Gambar 2.23 Event Mouse klik kiri



Gambar 2.24 Event Mouse klik kiri

The screenshot shows the PyCharm IDE interface. On the left, the Project tool window displays a file named 'Mouse_even_opencv_python.py' under the 'OpenCVExamples' project. The code in the editor window is as follows:

```
5     #print(events)
6
7     def click_event(event, x, y, flags, para:
8         if event == cv2.EVENT_LBUTTONDOWN:
9             print(x, ', ', y)
```

The Run tool window at the bottom shows a list of coordinates from the log:

| x | y |
|-----|-----|
| 66 | 60 |
| 98 | 163 |
| 221 | 94 |
| 175 | 220 |
| 319 | 212 |
| 174 | 293 |
| 74 | 252 |
| 221 | 395 |
| 304 | 296 |
| 428 | 427 |
| 76 | 360 |
| 110 | 447 |
| 468 | 72 |
| 384 | 38 |
| 406 | 135 |
| 319 | 123 |
| 326 | 47 |
| 110 | 447 |
| 76 | 360 |
| 221 | 395 |
| 174 | 293 |
| 319 | 212 |
| 175 | 220 |
| 221 | 94 |
| 98 | 163 |
| 66 | 60 |

A terminal window titled 'image' is open on the right, displaying the same list of coordinates.

Gambar 2.25 Event Mouse klik kiri

2.4.7 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 img = np.zeros((512, 512, 3), np.uint8)
41
42 cv2.imshow('image', img)
43
44 cv2.setMouseCallback('image', click_event)
45
46 cv2.waitKey(0)
```

- 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 yanga 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.

The screenshot shows the PyCharm IDE interface. On the left is the project tree with files like OpenCVExamples, venv, and various Python scripts. The main editor window displays the code for 'Mouse_even_opencv_python.py'. The code handles mouse events, specifically left and right clicks, to draw colored text on a black image. The right side shows a preview window titled 'image' displaying a black canvas with text at coordinates (117, 85) and (226, 87) in white, blue, and green respectively. Below the code editor is a 'Run' tool window showing the output of the script: '226 , 87', '218 , 276', and '117 , 85'.

```

11 strXY = str(x) + ' , ' + str(y)
12 cv2.putText(img, strXY, (x, y), font, .5, (0, 0, 0), 1)
13 cv2.imshow('image', img)
14 if event == cv2.EVENT_LBUTTONDOWN:
15     blue = img[y, x, 0]
16     green = img[y, x, 1]
17     red = img[y, x, 2]
18     font = cv2.FONT_HERSHEY_SIMPLEX
19     strBGR = str(blue) + ' , ' + str(green) +
20         str(red)
21     cv2.putText(img, strBGR, (x, y), font, .5,
22                 (255, 255, 255), 1)
23     cv2.imshow('image', img)
24
25 click_event()
26
27 def click_event():
28     img = np.zeros((512, 512, 3), np.uint8)
29     cv2.imshow('image', img)
30
31     cv2.setMouseCallback('image', click_event)
32
33 click_event() > if event == cv2.EVENT_RBUTTONDOWN:
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
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
307
308
309
309
310
311
312
313
314
315
315
316
317
317
318
319
319
320
321
321
322
323
323
324
325
325
326
327
327
328
329
329
330
331
331
332
333
333
334
335
335
336
337
337
338
339
339
340
341
341
342
343
343
344
345
345
346
347
347
348
349
349
350
351
351
352
353
353
354
355
355
356
357
357
358
359
359
360
361
361
362
363
363
364
365
365
366
367
367
368
369
369
370
371
371
372
373
373
374
375
375
376
377
377
378
379
379
380
381
381
382
383
383
384
385
385
386
387
387
388
389
389
390
391
391
392
393
393
394
395
395
396
397
397
398
399
399
400
401
401
402
403
403
404
405
405
406
407
407
408
409
409
410
411
411
412
413
413
414
415
415
416
417
417
418
419
419
420
421
421
422
423
423
424
425
425
426
427
427
428
429
429
430
431
431
432
433
433
434
435
435
436
437
437
438
439
439
440
441
441
442
443
443
444
445
445
446
447
447
448
449
449
450
451
451
452
453
453
454
455
455
456
457
457
458
459
459
460
461
461
462
463
463
464
465
465
466
467
467
468
469
469
470
471
471
472
473
473
474
475
475
476
477
477
478
479
479
480
481
481
482
483
483
484
485
485
486
487
487
488
489
489
490
491
491
492
493
493
494
495
495
496
497
497
498
499
499
500
501
501
502
503
503
504
505
505
506
507
507
508
509
509
510
511
511
512
513
513
514
515
515
516
517
517
518
519
519
520
521
521
522
523
523
524
525
525
526
527
527
528
529
529
530
531
531
532
533
533
534
535
535
536
537
537
538
539
539
540
541
541
542
543
543
544
545
545
546
547
547
548
549
549
550
551
551
552
553
553
554
555
555
556
557
557
558
559
559
560
561
561
562
563
563
564
565
565
566
567
567
568
569
569
570
571
571
572
573
573
574
575
575
576
577
577
578
579
579
580
581
581
582
583
583
584
585
585
586
587
587
588
589
589
590
591
591
592
593
593
594
595
595
596
597
597
598
599
599
600
601
601
602
603
603
604
605
605
606
607
607
608
609
609
610
611
611
612
613
613
614
615
615
616
617
617
618
619
619
620
621
621
622
623
623
624
625
625
626
627
627
628
629
629
630
631
631
632
633
633
634
635
635
636
637
637
638
639
639
640
641
641
642
643
643
644
645
645
646
647
647
648
649
649
650
651
651
652
653
653
654
655
655
656
657
657
658
659
659
660
661
661
662
663
663
664
665
665
666
667
667
668
669
669
670
671
671
672
673
673
674
675
675
676
677
677
678
679
679
680
681
681
682
683
683
684
685
685
686
687
687
688
689
689
690
691
691
692
693
693
694
695
695
696
697
697
698
699
699
700
701
701
702
703
703
704
705
705
706
707
707
708
709
709
710
711
711
712
713
713
714
715
715
716
717
717
718
719
719
720
721
721
722
723
723
724
725
725
726
727
727
728
729
729
730
731
731
732
733
733
734
735
735
736
737
737
738
739
739
740
741
741
742
743
743
744
745
745
746
747
747
748
749
749
750
751
751
752
753
753
754
755
755
756
757
757
758
759
759
760
761
761
762
763
763
764
765
765
766
767
767
768
769
769
770
771
771
772
773
773
774
775
775
776
777
777
778
779
779
780
781
781
782
783
783
784
785
785
786
787
787
788
789
789
790
791
791
792
793
793
794
795
795
796
797
797
798
799
799
800
801
801
802
803
803
804
805
805
806
807
807
808
809
809
810
811
811
812
813
813
814
815
815
816
817
817
818
819
819
820
821
821
822
823
823
824
825
825
826
827
827
828
829
829
830
831
831
832
833
833
834
835
835
836
837
837
838
839
839
840
841
841
842
843
843
844
845
845
846
847
847
848
849
849
850
851
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
861
862
863
863
864
865
865
866
867
867
868
869
869
870
871
871
872
873
873
874
875
875
876
877
877
878
879
879
880
881
881
882
883
883
884
885
885
886
887
887
888
889
889
890
891
891
892
893
893
894
895
895
896
897
897
898
899
899
900
901
901
902
903
903
904
905
905
906
907
907
908
909
909
910
911
911
912
913
913
914
915
915
916
917
917
918
919
919
920
921
921
922
923
923
924
925
925
926
927
927
928
929
929
930
931
931
932
933
933
934
935
935
936
937
937
938
939
939
940
941
941
942
943
943
944
945
945
946
947
947
948
949
949
950
951
951
952
953
953
954
955
955
956
957
957
958
959
959
960
961
961
962
963
963
964
965
965
966
967
967
968
969
969
970
971
971
972
973
973
974
975
975
976
977
977
978
979
979
980
981
981
982
983
983
984
985
985
986
987
987
988
989
989
990
991
991
992
993
993
994
995
995
996
997
997
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638

```

2.4.8 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 is the Project Explorer with files like OpenCVExamples, venv, and various Python scripts. The right side shows the code editor for `Mouse_even_opencv_python.py` and a preview window titled "image".

```

16     x,y = cv2.getMousePosition()
17     green = img[y, x, 1]
18     red = img[y, x, 2]
19     font = cv2.FONT_HERSHEY_SIMPLEX
20     strBGR = str(blue) + ', ' + str(green)
21     cv2.putText(img, strBGR, (x, y), font,
22                 cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255))
23     cv2.imshow('image', img)
24
25     #img = np.zeros((512, 512, 3), np.uint8)
26     #img = cv2.imread('najib.jpg')
27     #cv2.imshow('image', img)
28
29     cv2.setMouseCallback('image', click_event)
30
31     cv2.waitKey(0)
32     cv2.destroyAllWindows()

```

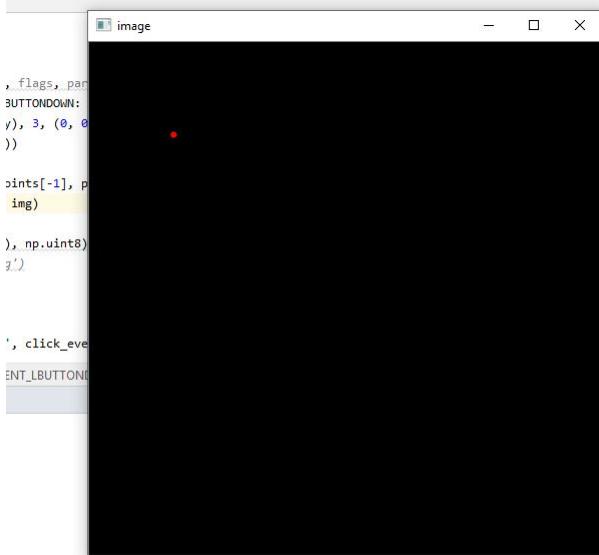
The preview window displays a portrait of a person with numerous numerical coordinates overlaid in yellow and green text, representing the pixel values at the click points. The coordinates include values like 25, 17, 232, 183, 39, 245, 31, etc.

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

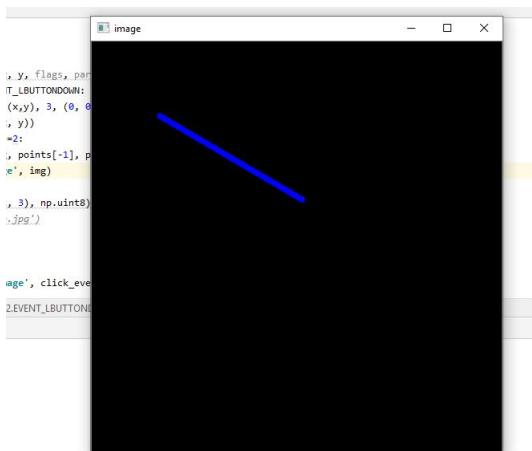
2.4.9 Event Mouse klik kiri membuat titik dan garis

```
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()
```

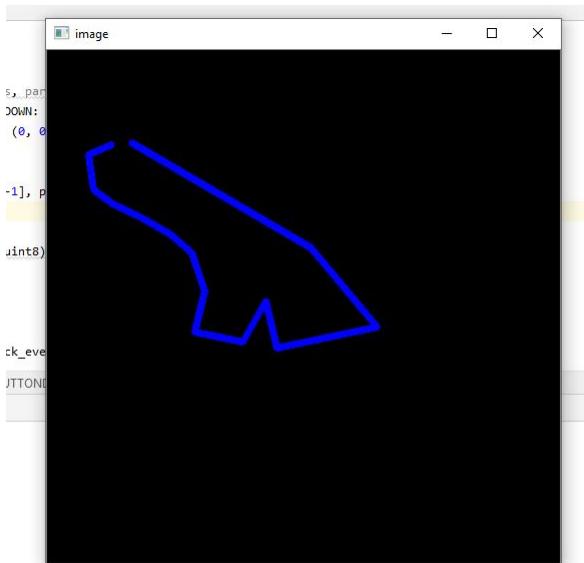
- i. Import numpy
- ii. import cv2
- iii. buat def dengan nama click event
- iv. jika mouse mengklik kiri maka akan melakukan sesuatu
- v. pada frame akan membuat sebuah titik berwarna merah sesuai lokasi mengklik frame
- vi. jika titik tersebut lebih dari sama dengan 2 maka setiap titik yang terakhir 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



Gambar 2.29 Event Mouse klik kiri membuat titik dan garis



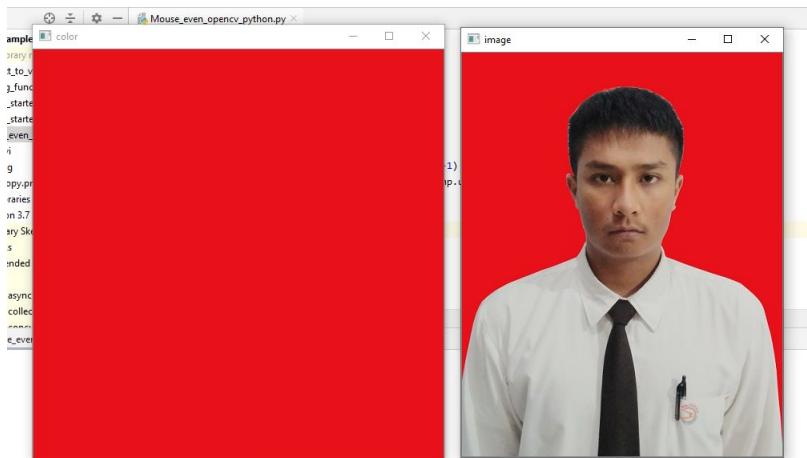
Gambar 2.30 Event Mouse klik kiri membuat titik dan garis

2.4.10 Membuat frame warna sesuai klik

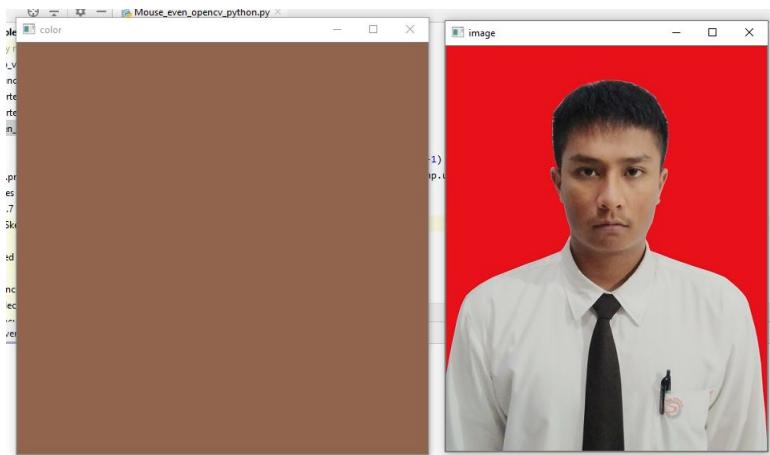
```
1 import numpy as np
2
3 import cv2
4
5 def click_event(event, x, y, flags, param):
6
7     if event == cv2.EVENT_LBUTTONDOWN:
8
9         blue = img[y, x, 0]
10
11         green = img[y, x, 1]
12
13         red = img[y, x, 2]
14
15         cv2.circle(img, (x,y), 3, (0, 0, 255), -1)
16
17     mycolorImage = 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

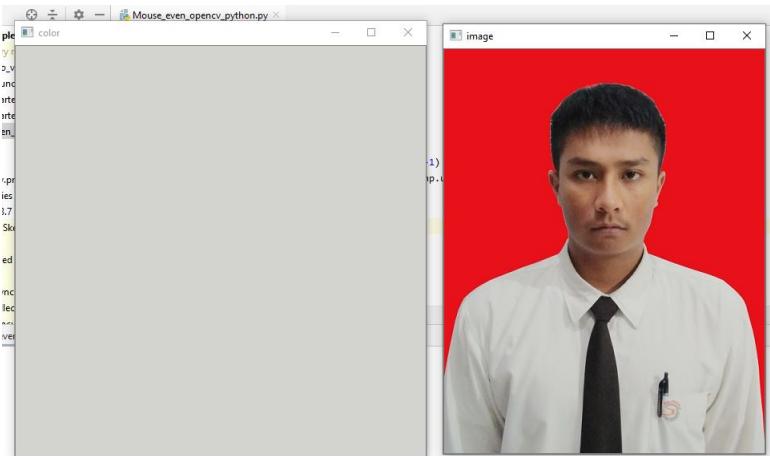
- xiv. kemudian gunakan waitKey untuk membuat frame agar tidak langsung mati atau tertutup otomatis.
- xv. destroyAllWindows digunakan untuk menutup frame.



Gambar 2.31 Membuat frame warna sesuai klik



Gambar 2.32 Membuat frame warna sesuai klik



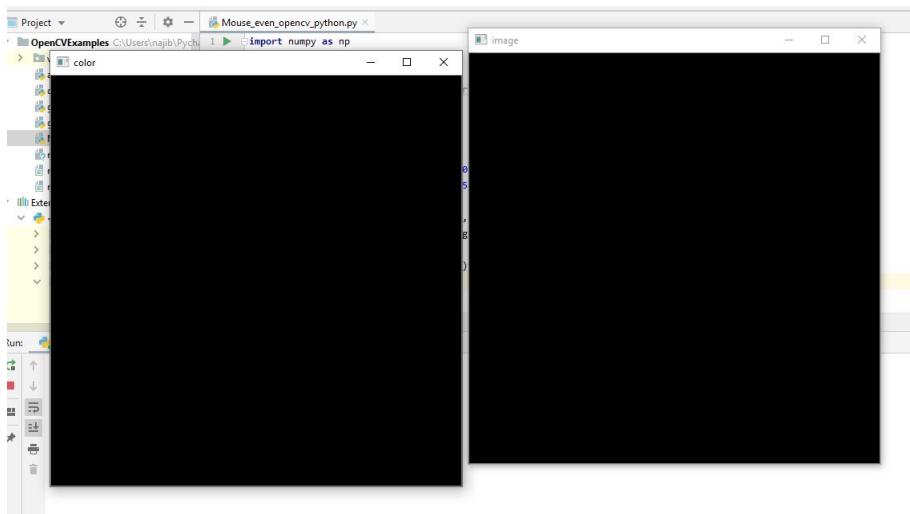
Gambar 2.33 Membuat frame warna sesuai klik

2.4.11 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

2.4.12 Berubah Ukuran 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 cv2.imshow('image', img)
18
19 cv2.waitKey(0)
20
21 cv2.destroyAllWindows()
```

i.

The screenshot shows a PyCharm IDE interface. On the left, there's a file tree for 'OpenCVExamples' containing various Python files and image files like 'messi5.jpg'. A blue oval highlights the 'Run' dropdown menu at the top of the run configuration list. The main area displays a script named 'basic_operations_on_image.py' with the following code:

```
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 b,g,r = cv2.split(img)
10 img = cv2.merge((b,g,r))
11
12 cv2.imshow('image', img)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
```

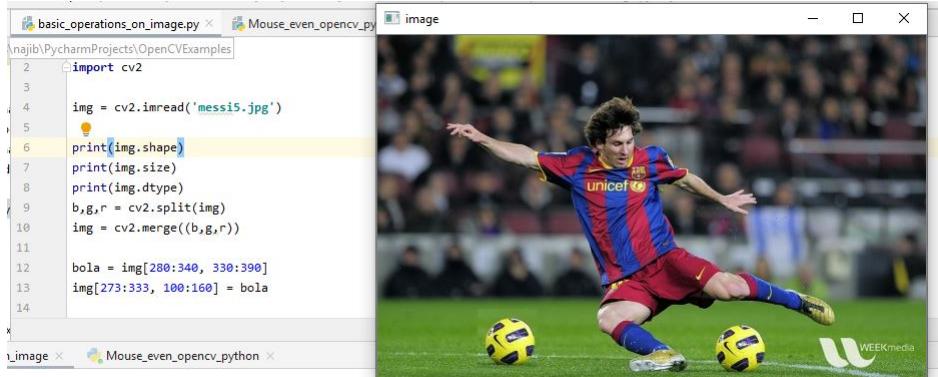
To the right of the code is a preview window titled 'image' showing a soccer player in action. Below the code editor, the 'Run' dropdown is open, showing the selected configuration 'basic_operations_on_image' which has '(342, 548, 3)' listed under its parameters.

Gambar 2.35 Membuat Frame Numpy

2.4.13 Berubah Ukuran 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.



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 an image named 'messi5.jpg' using OpenCV's cv2.imread function. It then prints the image's shape, size, and dtype. The image is split into three channels (b, g, r) and merged back together. A specific region of the image is then extracted and assigned to a variable 'bola'. The 'image' tab displays a soccer player in a red and blue jersey, performing a bicycle kick on a green field with two yellow and blue soccer balls. The image has a 'WEEKmedia' watermark in the bottom right corner.

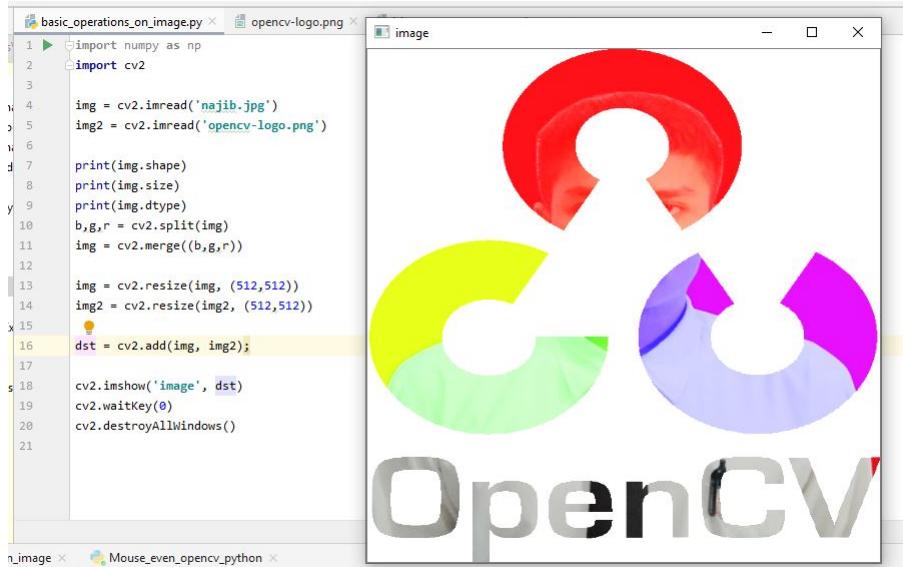
```
basic_operations_on_image.py
1 import cv2
2
3
4 img = cv2.imread('messi5.jpg')
5
6 print(img.shape)
7 print(img.size)
8 print(img.dtype)
9 b,g,r = cv2.split(img)
10 img = cv2.merge((b,g,r))
11
12 bola = img[280:340, 330:390]
13 img[273:333, 100:160] = bola
14
```

Gambar 2.36 Membuat Frame Numpy

2.4.14 Berubah Ukuran 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.



The screenshot shows a Python code editor and a window titled 'image'. The code reads two images ('najib.jpg' and 'opencv-logo.png'), prints their shapes and sizes, splits them into color channels (B, G, R), merges them back into a single image, resizes both to 512x512 pixels, adds them together, and displays the result. The resulting image is a composite of the two logos.

```

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()
21

```

Gambar 2.37 Membuat Frame Numpy

2.4.15 Berubah Ukuran Frame

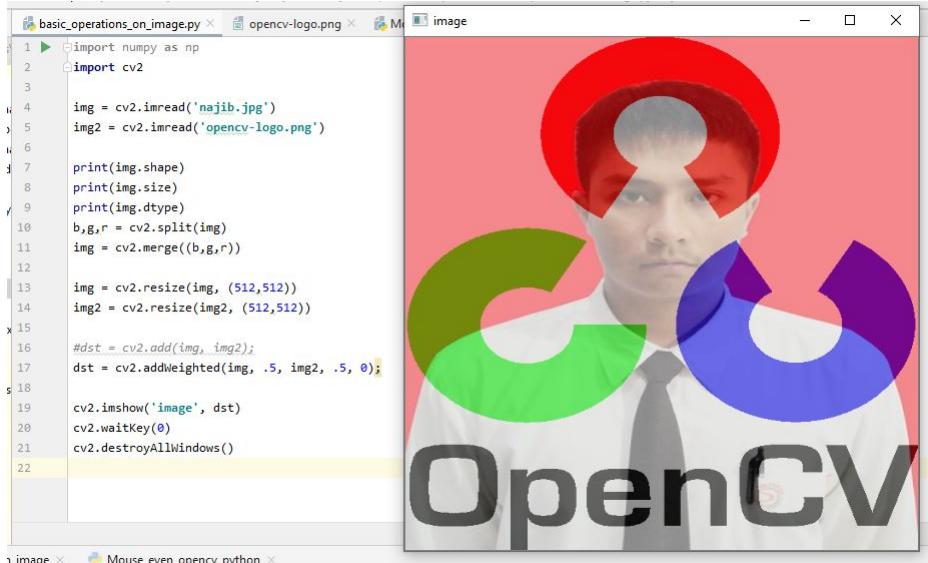
```

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 dst = cv2.addWeighted(img, .5, img2, .5, 0);
25

```

```
26 cv2.imshow('image', dst)
27
28 cv2.waitKey(0)
29
30 cv2.destroyAllWindows()
```

i.

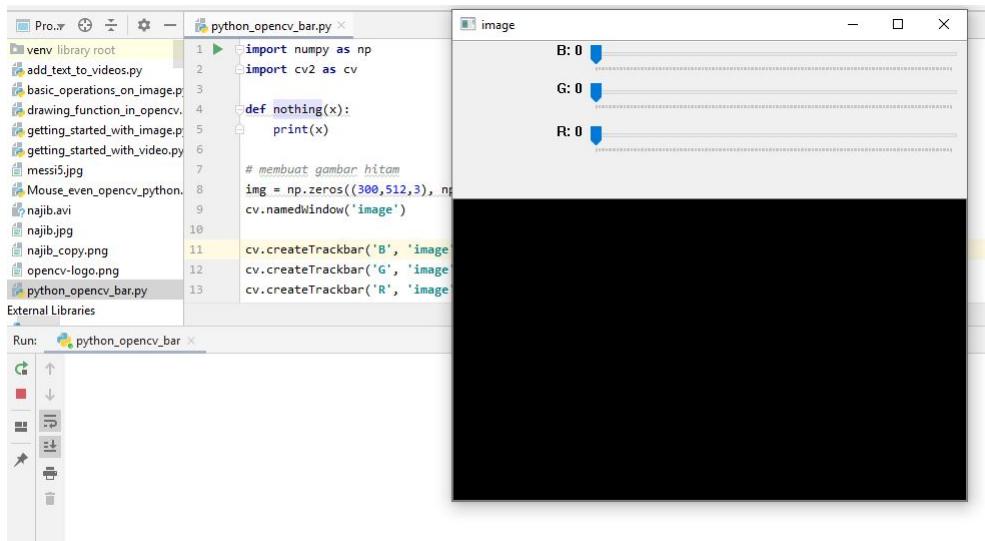


Gambar 2.38 Membuat Frame Numpy

2.4.16 Berubah Ukuran Frame

```
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 img = np.zeros((300,512,3), np.uint8)
10
11 cv.namedWindow('image')
12
13 cv.createTrackbar('B', 'image', 0, 255, nothing)
14
15 cv.createTrackbar('G', 'image', 0, 255, nothing)
16
17 cv.createTrackbar('R', 'image', 0, 255, nothing)
18
19 while(1):
20     cv.imshow('image', img)
21
22     k = cv.waitKey(1) & 0xFF
23
24     if k == 27:
25         break
26
27 cv.destroyAllWindows()
```

i.



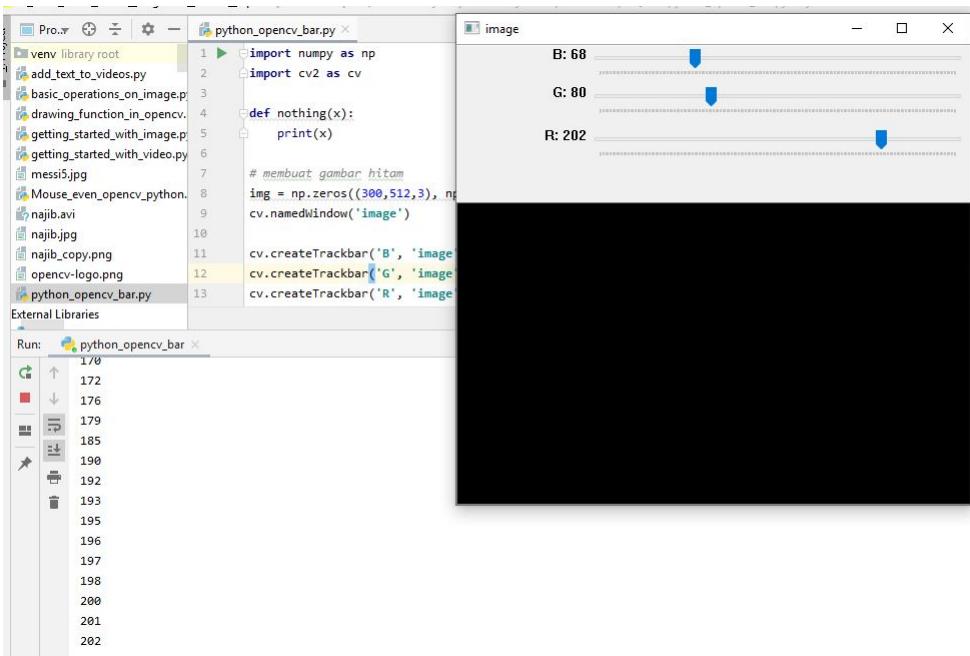
Gambar 2.39 Membuat Frame Numpy

2.4.17 Berubah Ukuran Frame

```

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 img = np.zeros((300,512,3), np.uint8)
10
11 cv.namedWindow('image')
12
13 cv.createTrackbar('B', 'image', 0, 255, nothing)
14
15 cv.createTrackbar('G', 'image', 0, 255, nothing)
16
17 cv.createTrackbar('R', 'image', 0, 255, nothing)
18
19 while(1):
20     cv.imshow('image', img)
21
22     k = cv.waitKey(1) & 0xFF
23
24     if k == 27:
25         break
26
27     b = cv.getTrackbarPos('B', 'image')

```



Gambar 2.40 Membuat Frame Numpy

```
28
29     g = cv.getTrackbarPos('G', 'image')
30
31     r = cv.getTrackbarPos('R', 'image')
32
33     img[:, :] = [b, g, r]
34
35 cv.destroyAllWindows()
```

i.

The screenshot shows the PyCharm IDE interface. On the left is the project tree with files like 'python_opencv_bar.py' selected. The main window displays the code for creating a color bar:

```

1 # membuat gambar hitam
2 img = np.zeros((300,512,3), np.uint8)
3 cv.namedWindow('image')
4
5 cv.createTrackbar('B', 'image', 0, 255, nothing)
6 cv.createTrackbar('G', 'image', 0, 255, nothing)
7 cv.createTrackbar('R', 'image', 0, 255, nothing)
8
9 while(1):
10     cv.imshow('image', img)
11     k = cv.waitKey(1) & 0xFF
12     if k == 27:
13         break
14
15     b = cv.getTrackbarPos('B', 'image')
16     g = cv.getTrackbarPos('G', 'image')
17     r = cv.getTrackbarPos('R', 'image')
18
19     img[:, :] = [b, g, r]
20
21 cv.destroyAllWindows()
22
23 while(1):
24     cv.imshow('image', img)
25
26
27

```

To the right of the code is a preview window titled 'image' showing a black frame. Above the preview are three trackbars labeled 'B: 0', 'G: 0', and 'R: 0'. Below the preview window is a toolbar with various icons.

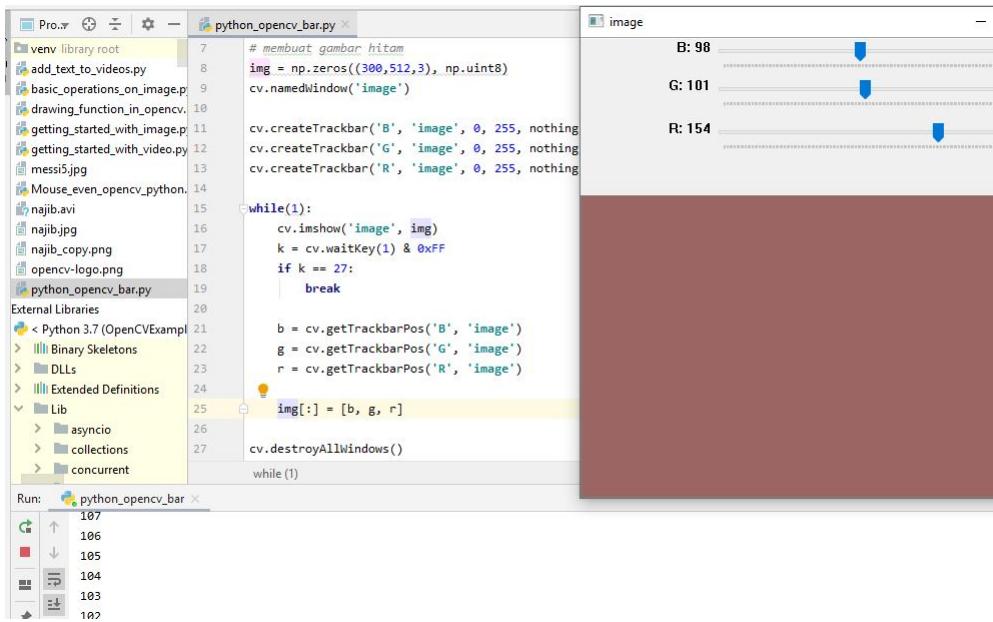
Gambar 2.41 Membuat Frame Numpy

2.4.18 Berubah Ukuran Frame

```

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 img = np.zeros((300,512,3), np.uint8)
10
11 cv.namedWindow('image')
12
13 cv.createTrackbar('B', 'image', 0, 255, nothing)
14
15 cv.createTrackbar('G', 'image', 0, 255, nothing)
16
17 cv.createTrackbar('R', 'image', 0, 255, nothing)
18
19 switch = '0 : OFF\n1 : ON'
20
21 cv.createTrackbar(switch, 'image', 0, 1, nothing)
22
23 while(1):
24     cv.imshow('image', img)

```



Gambar 2.42 Membuat Frame Numpy

```

26 k = cv.waitKey(1) & 0xFF
27
28 if k == 27:
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 s = cv.getTrackbarPos(switch, 'image')
38
39 if s == 0:
40     img[:] = 0
41 else:
42     img[:] = [b, g, r]
43
44 cv.destroyAllWindows()

```

i.

The screenshot shows the PyCharm IDE interface. On the left, the project structure displays files like 'python_opencv_bar.py' and 'najib.jpg'. The main editor window contains 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', 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:
22
23

```

To the right of the code editor is a 'Trackbar' configuration panel titled 'image' with three sliders: B: 120, G: 170, R: 66, and a switch labeled '0 : OFF...ON: 0'.

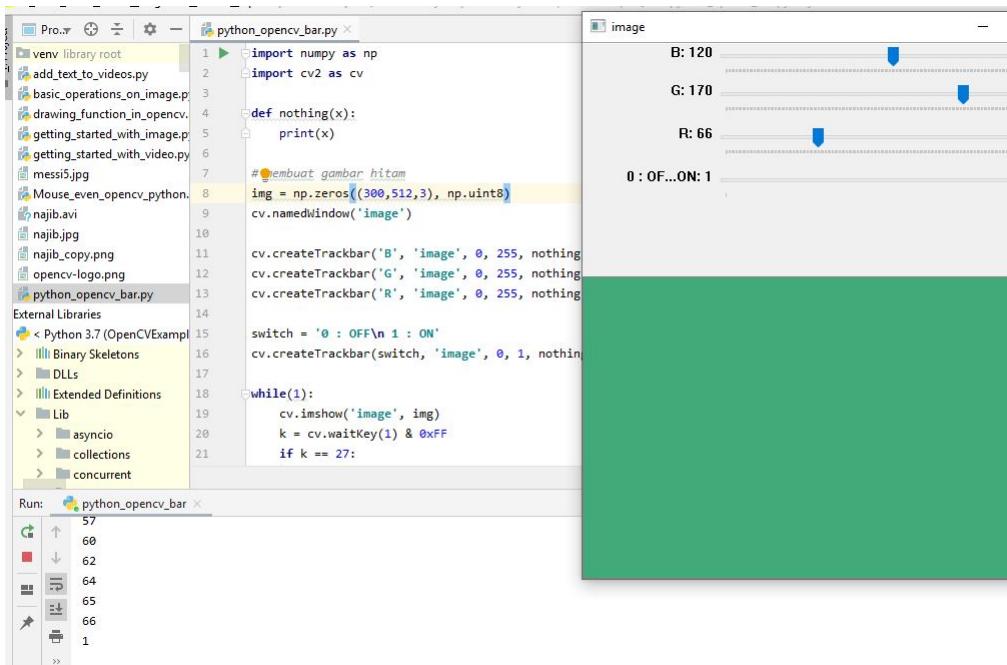
Gambar 2.43 Membuat Frame Numpy

2.4.19 Berubah Ukuran Frame

```

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 cv.namedWindow('image')
10
11 cv.createTrackbar('CP', 'image', 10, 400, nothing)
12
13 switch = '0 : OFF\n1 : ON'
14
15 cv.createTrackbar(switch, 'image', 0, 1, nothing)
16
17 while(1):
18     img = cv.imread('najib.jpg')
19
20     pos = cv.getTrackbarPos('CP', 'image')
21
22     font = cv.FONT_HERSHEY_SIMPLEX
23

```



Gambar 2.44 Membuat Frame Numpy

```

24         cv.putText(img, str(pos), (50, 150), font, 6, (0, 255, 255), 10)
25
26         k = cv.waitKey(1) & 0xFF
27
28         if k == 27:
29             break
30
31         s = cv.getTrackbarPos(switch, 'image')
32
33         if s == 0:
34             pass
35         else:
36             img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
37
38         img = cv.imshow('image', img)
39
40         cv.destroyAllWindows()

```

i.



Gambar 2.45 Membuat Frame Numpy

2.4.20 Berubah Ukuran Frame

```

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

```

i.



The screenshot shows a PyCharm interface with several windows:

- Project:** Shows files like `python_opencv_bar.py`, `najib.jpg`, and `opencv-logo.png`.
- Code Editor:** Displays the Python script `python_opencv_bar.py` with code for creating a window, trackbars, and displaying text.
- Run:** Shows the output of the script, which is a frame with a red background and yellow text "179". A trackbar at the top is labeled "CP: 179" with a slider between "0 : OFF" and "1 : ON".

```

1 import cv2 as cv
2
3 def nothing(x):
4     print(x)
5
6 # membuat gambar hitam
7 cv.namedWindow('image')
8
9 cv.createTrackbar('CP', 'image', 10, 40)
10 switch = '0 : OFF\n1 : ON'
11 cv.createTrackbar(switch, 'image', 0, 1)
12
13 while(1):
14     img = cv.imread('najib.jpg')
15     pos = cv.getTrackbarPos('CP', 'image')
16     font = cv.FONT_HERSHEY_SIMPLEX
17     cv.putText(img, str(pos), (50, 150))
18
19     k = cv.waitKey(1) & 0xFF
20     if k == 27:
21         break
22

```

Gambar 2.46 Membuat Frame Numpy

2.4.21 Berubah Ukuran Frame

```

1 import cv2
2
3 import numpy as np
4
5 def nothing(x):
6     pass
7
8 #cv2.namedWindow(" Tracking ")
9
10 while True:
11     frame = cv2.imread('PARKIR.png')
12
13     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
14
15     l_b = np.array([110, 50, 50])
16
17     u_b = np.array([130, 255, 255])
18
19     mask = cv2.inRange(hsv, l_b, u_b)
20
21     res = cv2.bitwise_and(frame, frame, mask=mask)
22
23     cv2.imshow("frame", frame)

```

The screenshot shows the PyCharm IDE interface. On the left, the project structure displays various OpenCV-related files like 'python_opencv_bar.py'. The main editor window shows the following Python code:

```

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, 40)
11
12 switch = '0 : OFF\n1 : ON'
13 cv.createTrackbar(switch, 'image', 0, 1)
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),
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. The number '179' is overlaid on the image. A status bar at the top of the window says 'CP: 179'.

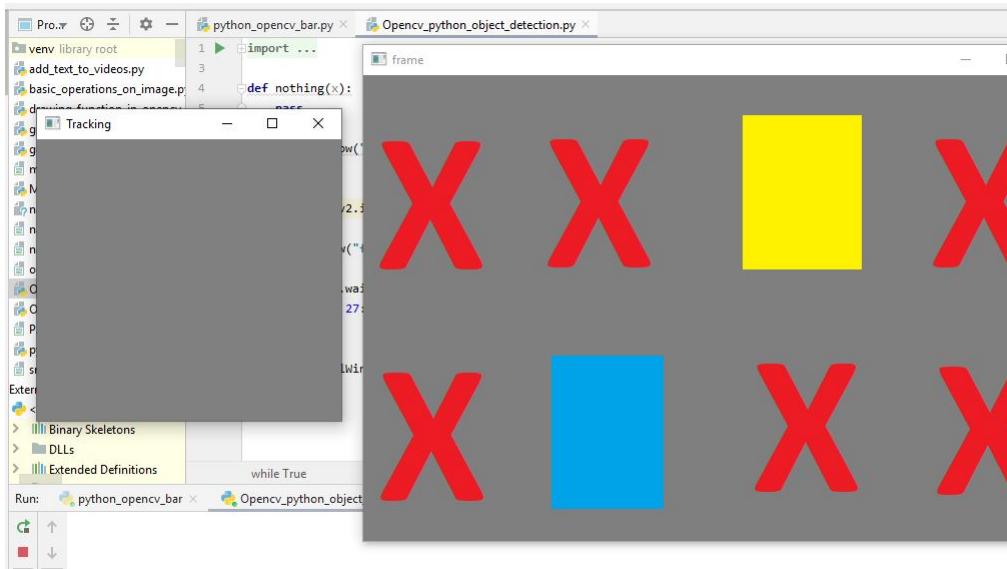
Gambar 2.47 Membuat Frame Numpy

```

24 cv2.imshow("mask", mask)
25
26 cv2.imshow("res", res)
27
28 key = cv2.waitKey(1)
29
30 if key == 27:
31     break
32
33 cv2.destroyAllWindows()
34

```

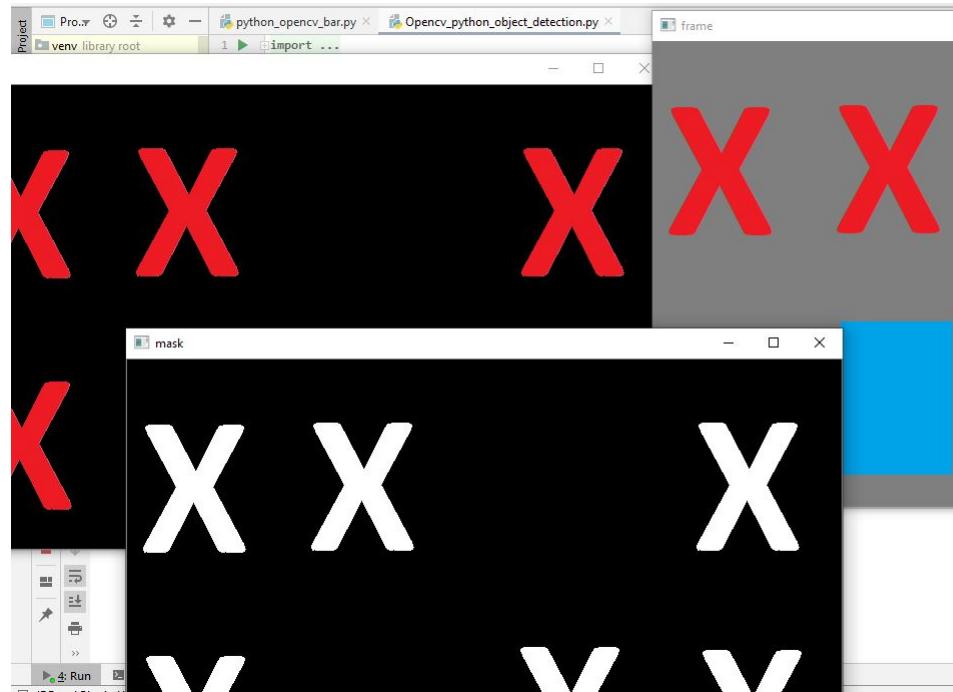
i.



Gambar 2.48 Membuat Frame Numpy

2.4.22 Berubah Ukuran Frame

```
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)
```

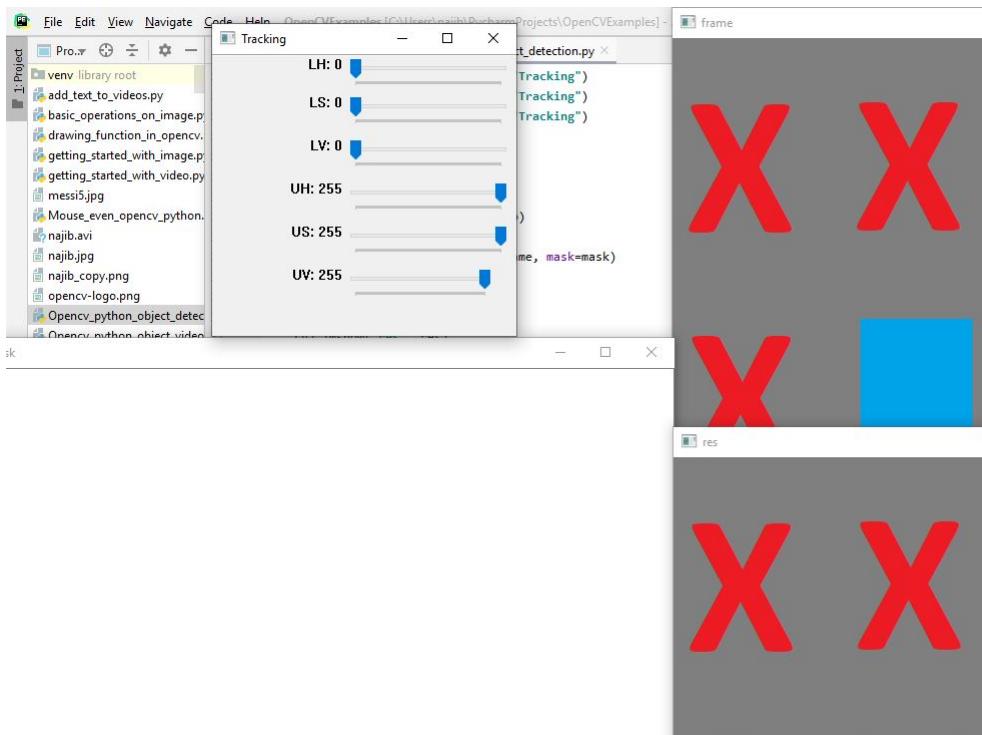


Gambar 2.49 Membuat Frame Numpy

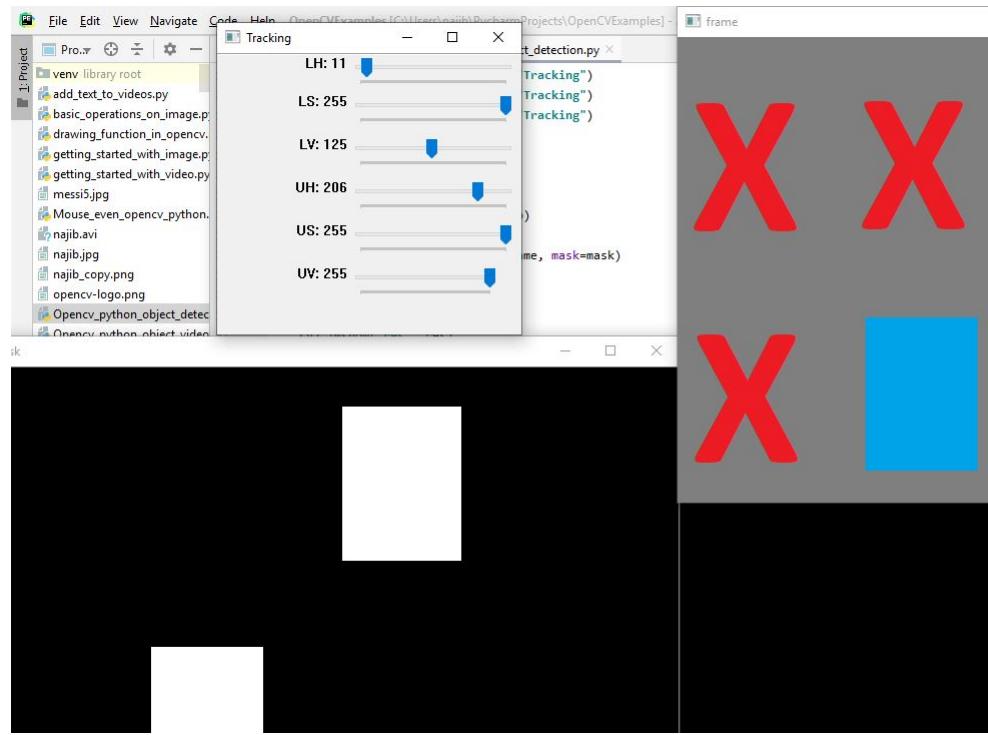
```
27 l_h = cv2.getTrackbarPos("LH", "Tracking")
28 l_s = cv2.getTrackbarPos("LS", "Tracking")
29 l_v = cv2.getTrackbarPos("LV", "Tracking")
30
31 u_h = cv2.getTrackbarPos("UH", "Tracking")
32 u_s = cv2.getTrackbarPos("US", "Tracking")
33 u_v = cv2.getTrackbarPos("UV", "Tracking")
34
35 l_b = np.array([l_h, 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.



Gambar 2.50 Membuat Frame Numpy



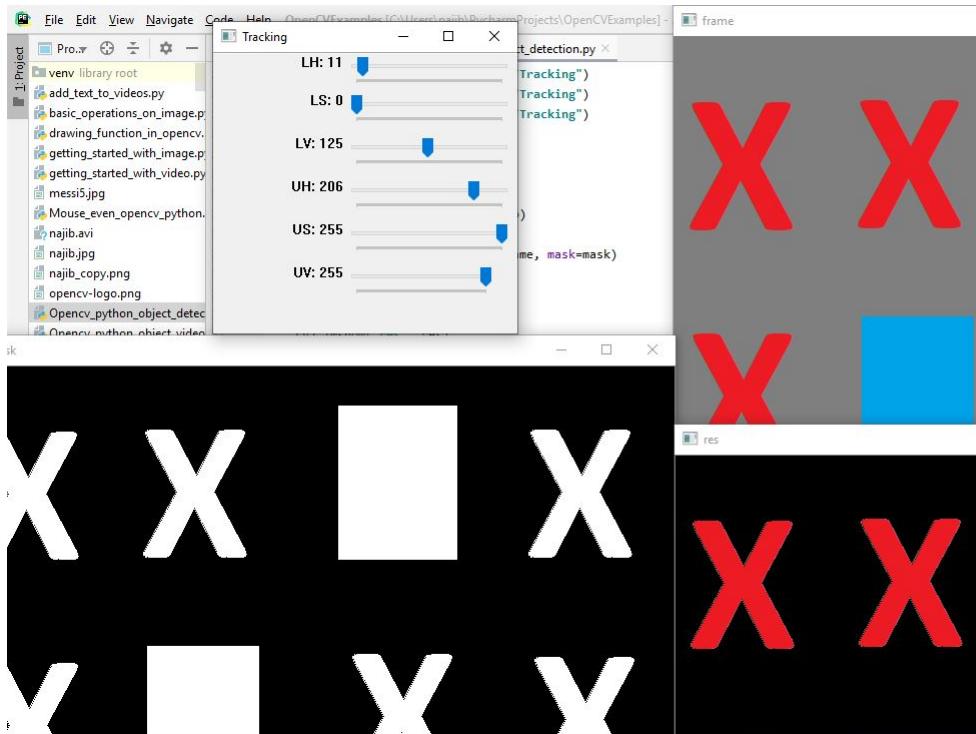
Gambar 2.51 Membuat Frame Numpy

2.4.23 Berubah Ukuran Frame

```

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

```

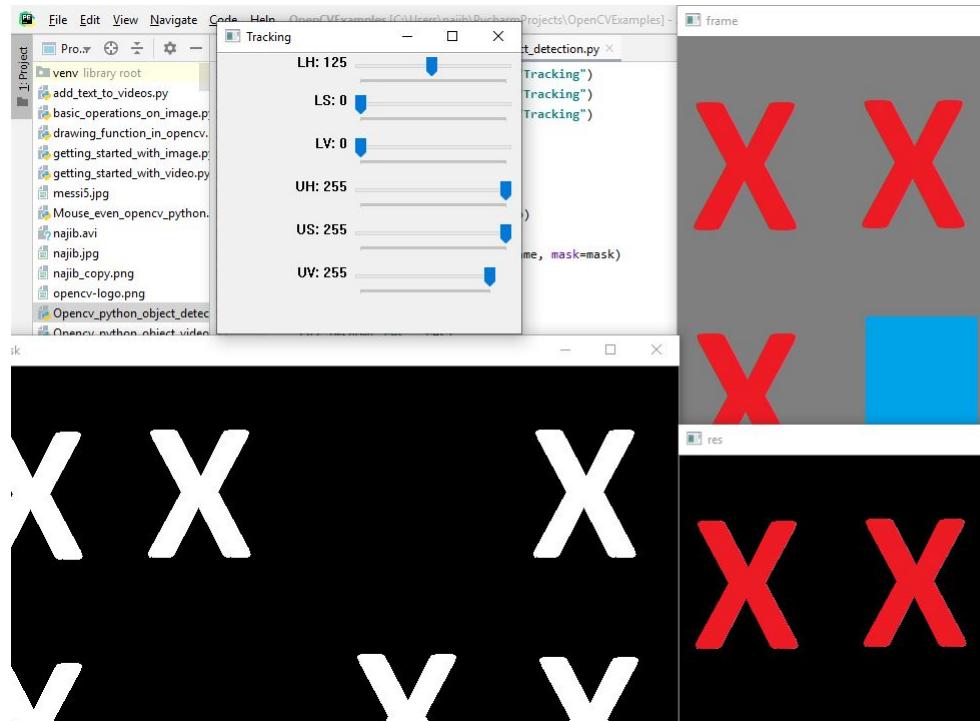


Gambar 2.52 Membuat Frame Numpy

```

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)

```



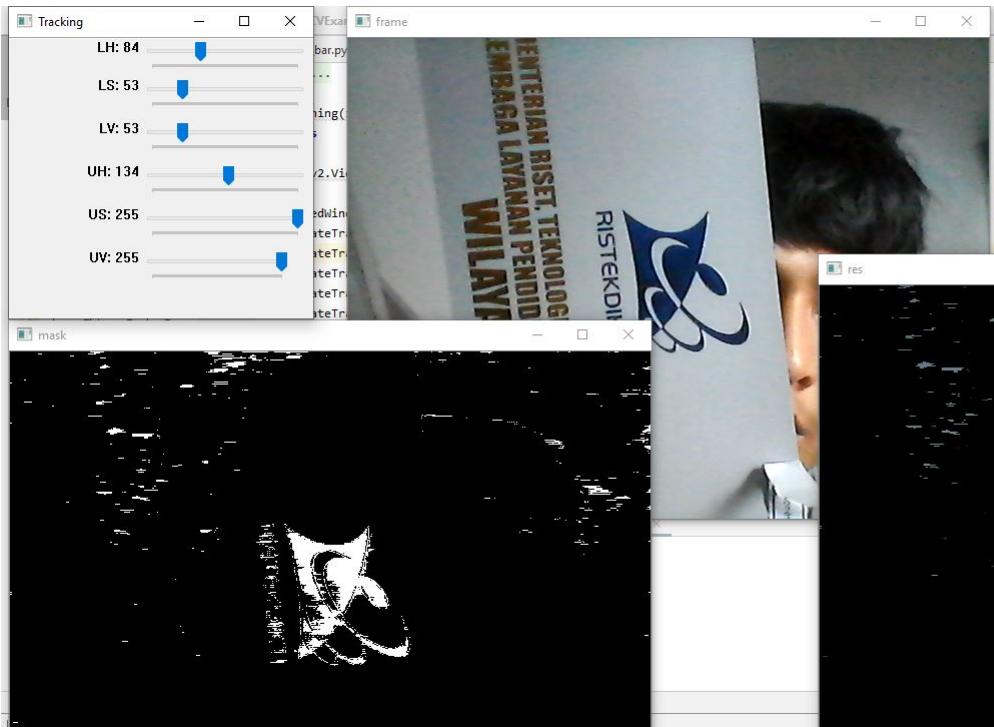
Gambar 2.53 Membuat Frame Numpy

```

42 res = cv2.bitwise_and(frame, frame, mask=mask)
43
44 cv2.imshow("frame", frame)
45 cv2.imshow("mask", mask)
46 cv2.imshow("res", res)
47
48 key = cv2.waitKey(1)
49
50 if key == 27:
51     break
52
53 cap.release()
54
55 cv2.destroyAllWindows()
56

```

i.



Gambar 2.54 Membuat Frame Numpy

