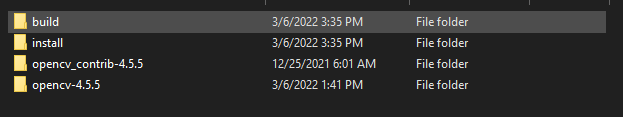
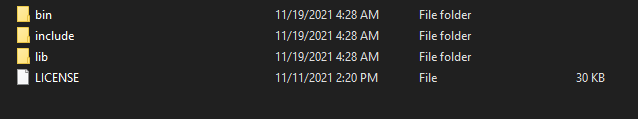
Hal pertama yang dilakukan agar dapat meng-*install library* opencv yang mendukung GPU adalah mengunduh *library* opencv pada opencv.org dan opencv contrib di link <https://github.com/opencv/opencv_contrib>, kemudian ekstrak keduanya menggunakan winrar dan satukan pada folder yang sama dan buat satu folder bernama build dan install.



Gambar 3.2.19 Folder persiapan untuk Install library opencv GPU

Langkah selanjutnya adalah meng-*install* cuda dan cudnn yang sesuai dengan versi cuda akan kita *install*, hal ini diperlukan sebagai akses untuk menggunakan GPU pada program deteksi objek. proses *installisasi* cuda cukup mudah, dimana kita hanya perlu menjalankan file hasil unduh cuda di <https://developer.nvidia.com/cuda-downloads>, untuk meng-*install* cudnn hal yang perlu kita lakukan adalah ekstraksi hasil unduh cudnn di <https://developer.nvidia.com/rdp/cudnn-archive>.



Gambar 3.2.20 Hasil Ekstaksi cudnn

Langkah selanjutnya adalah memindahkan isi dari file cudnn ini ke file cuda dengan mengikuti instruksi berikut :

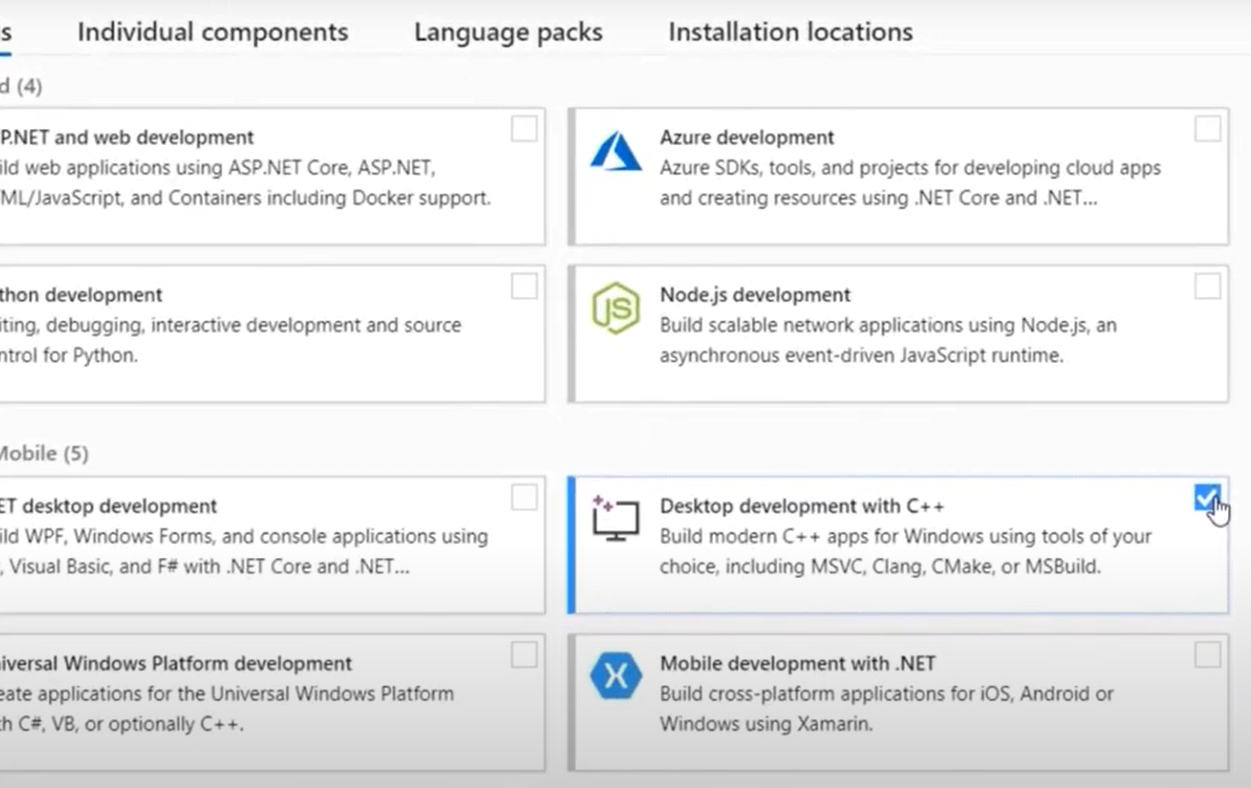
1. Pindahkan file pada<installpath>\cuda\bin\cudnn\*.dll ke C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\vx.x\bin.
2. Pindahkan file pada <installpath>\cuda\include\cudnn\*.h ke C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\vx.x\include.
3. Pindahkan file pada <installpath>\cuda\lib\x64\cudnn\*.lib ke C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\vx.x\lib\x64.

Keterangan :

Tanda bintang (\*) = semua file yang berada di dalam folder tersebut

vx.x = versi cuda yang telah di*install*

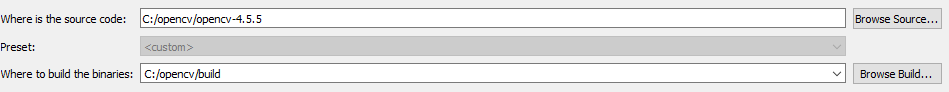
Setelah selesai meng-*install* cuda dan cudnn, Langkah selanjutnya adalah meng-*install* visual studio, disini penulis menggunakan visual studio 16 2019, saat melakukan proses *installisasi* hal yang perlu diperhatikan adalah pada menu *workloads* lakukan *enable* pada pilihan ”*Desktop development with C++”*.



Gambar 3.2.21 Tampilan menu workloads pada proses installasi visual studio 2019

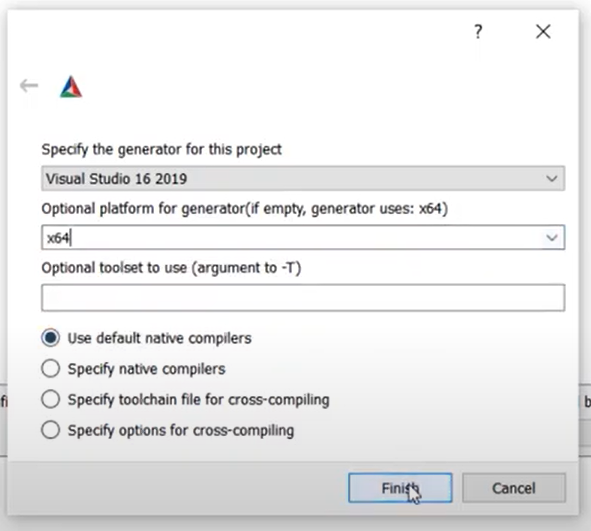
Langkah selanjutnya adalah meng-*install* cmake dan anaconda, cmake disini berperan untuk melakukan proses *install library* opencv dan anaconda sebagai *interprenter* python yang berisi *library* opencv GPU, untuk proses *install* kedua aplikasi cukup mudah dimana kita hanya perlu menjalankan file hasil unduh anaconda di <https://www.anaconda.com/products/individual> dan file hasil unduh cmake di <https://cmake.org/download/>.

Setelah proses *install* kedua aplikasi selesai, buka terlebih dahulu anaconda untuk meng-*install* *library* numpy menggunakan sintaks “pip install numpy”, Langkah selanjutnya adalah buka aplikasi cmake lalu masukkan folder opencv dan build pada gambar 3.2.18 melalui tombol *browse source* dan *browse build* pada gambar 3.2.20.



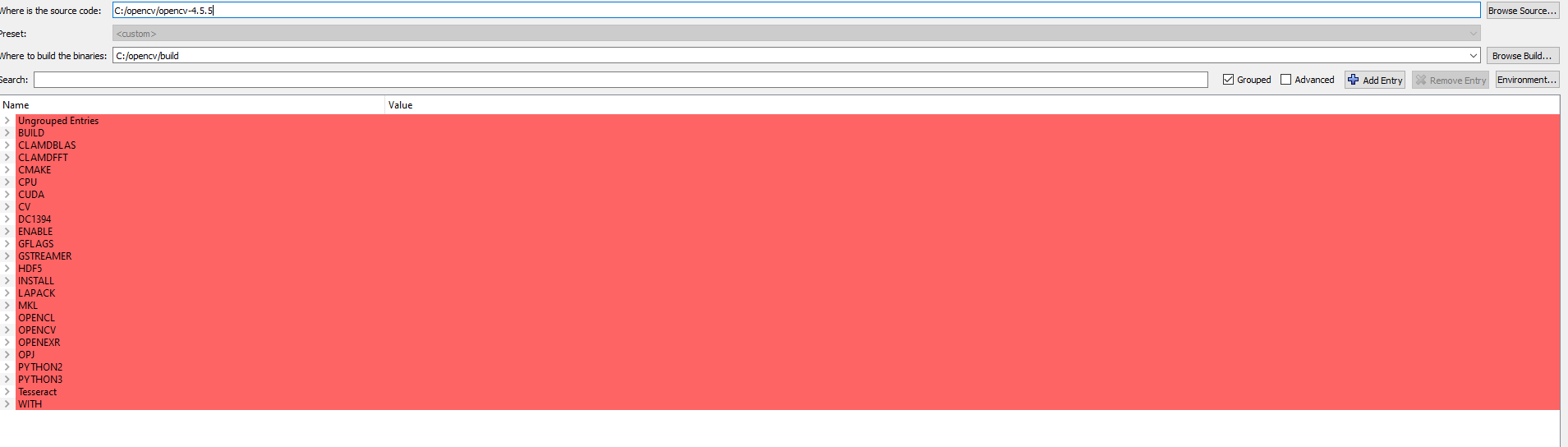
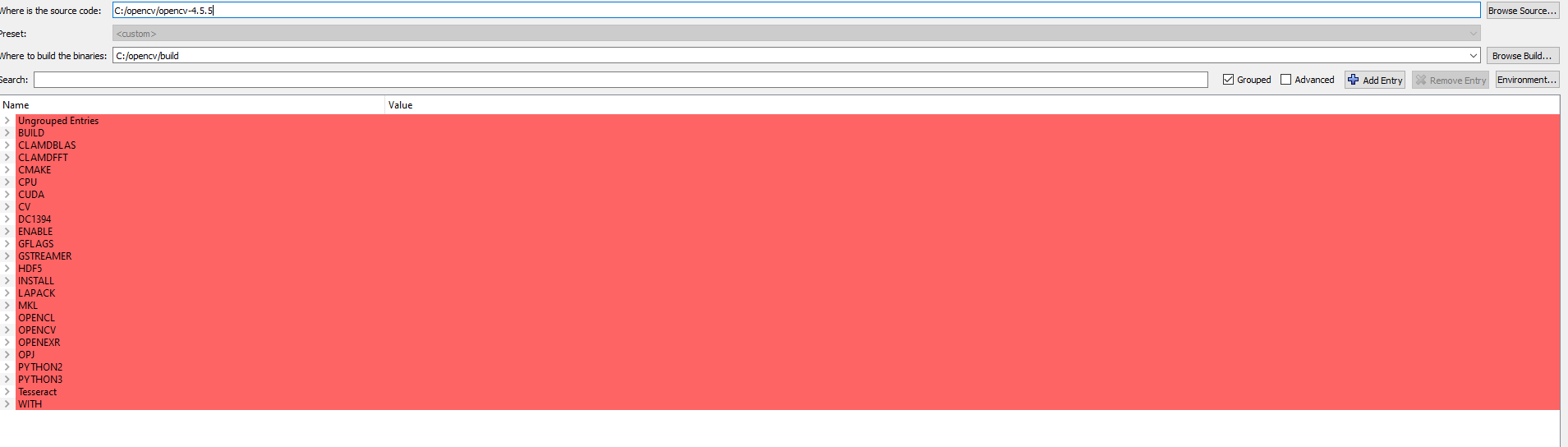
Gambar 3.2.22 Tampilan aplikasi cmake setelah dimasukkan library opencv

Kemudian klik tombol *configure* yang terletak dibawah tampilan aplikasi cmake hingga memunculkan *pop-up* dan gunakan pilihan yang sesuai seperti yang dicontohkan pada gambar 3.2.20



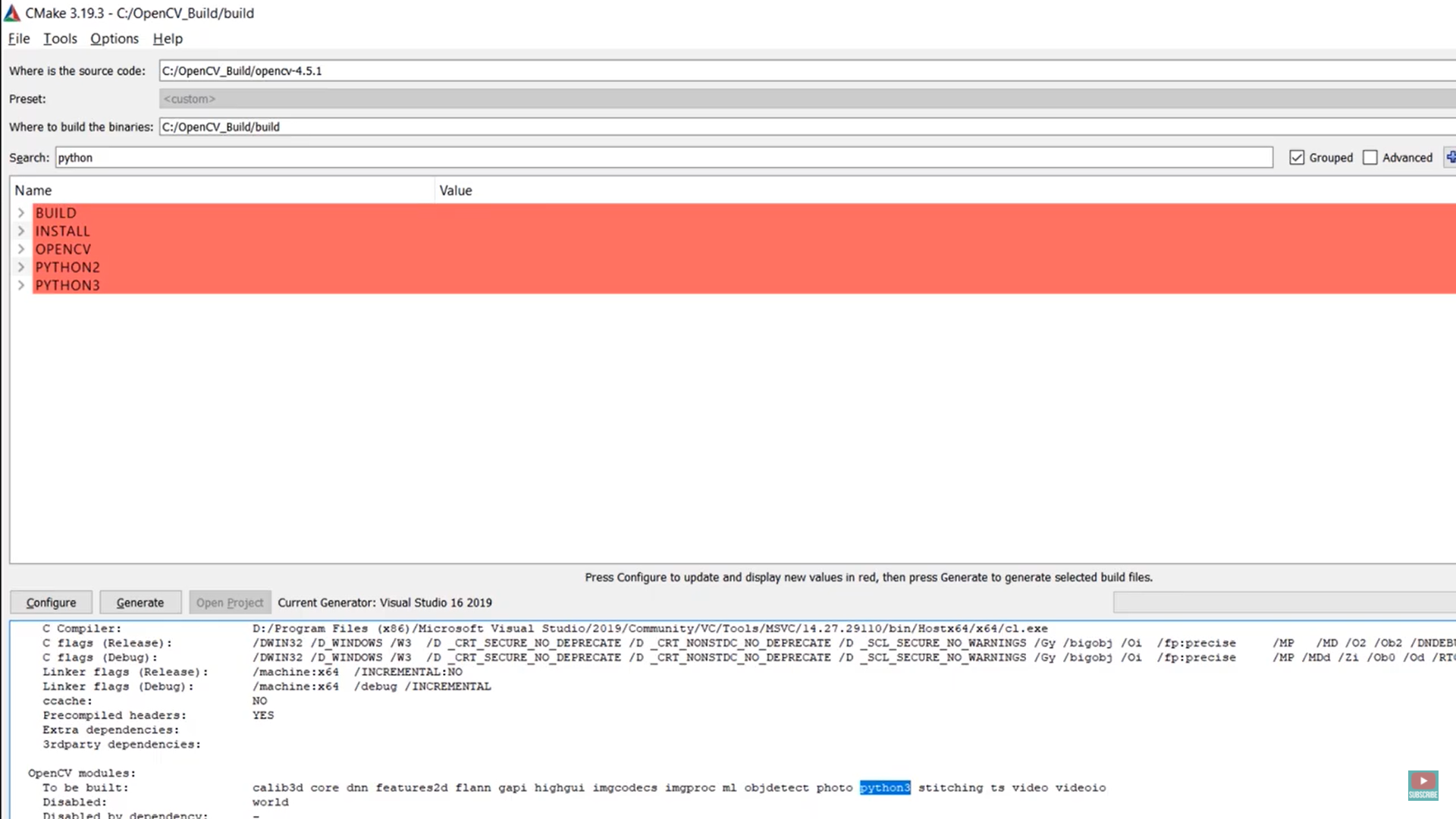
Gambar 3.2.23 *Pop-up configure* pada aplikasi cmake

Klik tombol *finish* dan tunggu proses *configure* selesai, setelah proses *configure* selesai maka akan muncul tampilan seperti gambar 3.2.22.



Gambar 3.2.24 Tampilan aplikasi setelah proses configure selesai

Langkah selanjutnya adalah mengecek apakah *library* opencv GPU telah *support* untuk python pada menu dibawah tombol *configure,* apabila *support* maka tampilan akan seperti gambar 3.2.23.



Gambar 3.2.25 Tampilan aplikasi yang menyatakan bahwa Opencv GPU dapat digunakan untuk python

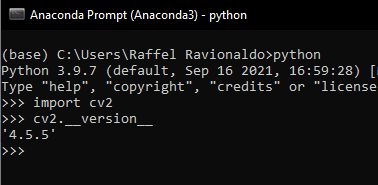
Langkah selanjutnya adalah konfigurasi agar kita bisa *generate library* opencv, caranya adalah lakukan *enable* pada bagian berikut (lakukan proses *enable* pada tampilan pada gambar 3.2.22 dan untuk mempercepat dapat menggunakan fitur *search* yang terlihat pada gambar 3.2.22) :

1. *WITH\_CUDA*
2. *BUILD\_opencv\_dnn*
3. *OPENCV\_DNN\_CUDA*
4. *ENABLE\_FAST\_MATH*
5. *BUILD\_opencv\_world*
6. *BUILD\_opencv\_python3*
7. Pada *OPENCV\_EXTRA\_MODULES\_PATH* masukkan *path* dari folder *modules* opencv-contrib

Kemudian klik *configure* sekali lagi, dan lakukan *enable* pada *CUDA\_FAST\_MATH* dan pada CUDA\_ARCH\_BIN pilih arsitektur GPU Nvidia yang anda gunakan, untuk mengetahui arsitektur GPU Nvidia yang kita gunakan pembaca dapat mencarinya pada laman berikut <https://en.wikipedia.org/wiki/CUDA> pada tabel bagian *compute capability (version).* Kemudian *search CMAKE\_INSTALL\_PREFIX* dan masukkan *path* folder install pada gambar 3.2.17, Langkah selanjutnya *search* CMAKE\_CONFIGURATION\_TYPES dan hapus *debug*. Klik *configure* sekali lagi dan kemudian klik *generate*.

Langkah selanjutnya adalah meng*install* opencv yang telah di *configure* melalui aplikasi cmake menggunakan *command prompt* pada windows dengan sintaks:  
"C:\ProgramFiles\CMake\bin\cmake.exe" --build "*path* folder build opencv, contoh : C:\OpenCV\_Build\build " --target INSTALL --config Release”

Setelah proses *installisasi* selesai, kita dapat mengecek apakah *library* opencv telah berhasil di *install* melalui anaconda prompt dengan sintaks python, import cv2, cv2.\_\_version\_\_.



Gambar 3.2.26 Tampilan untuk mengecek *library* opencv pada anaconda prompt