IDENTIFIKASI WAJAH MANUSIA MENGGUNAKAN YOLO *FRAMEWORKS* DENGAN METODE *SCALE MODIFIER* SEBAGAI *PREPROCESSING* SECARA *REAL TIME*

SKRIPSI

Disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T)

Disusun oleh:

ADAM ABDUL MALIK

NPM. 3332160050

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS SULTAN AGENG TIRTAYASA

2023

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya sebagai penulis skripsi berikut:

Judul	: Identifikasi Wajah Manusia Menggunakan Yolo *Frameworks* Dengan Metode *Scale Modifier* Sebagai *Preprocessing* Secara *Real Time.*
Nama Mahasiswa	: Adam Abdul Malik
NPM	: 3332160050
Fakultas/Jurusan	: Fakultas Teknik / Jurusan Teknik Elektro

Menyatakan dengan sesungguhnya bahwa Skripsi tersebut di atas adalah benar-benar hasil karya asli saya dan tidak memuat hasil karya orang lain, kecuali dinyatakan melalui rujukan yang benar dan dapat dipertanggung jawabkan. Apabila dikemudian hari ditemukan hal-hal yang menunjukkan bahwa sebagian atau seluruh karya ini bukan karya saya, maka saya bersedia dituntut melalui hukum yang berlaku. Saya juga bersedia menanggung segala akibat hukum yang timbul dari pernyataan yang secara sadar dan sengaja saya nyatakan melalui lembar ini.

Cilegon, Juli 2023

Adam Abdul Malik

NPM.3332160050

ii

LEMBAR PENGESAHAN

Dengan ini ditetapkan bahwa skripsi berikut:

Judul	:Identifikasi Wajah Manusia Menggunakan Yolo *Frameworks* Dengan Metode *Scale Modifier* Sebagai *Preprocessing* Secara *Real Time*.
Nama Mahasiswa	: Adam Abdul Malik
NPM	: 3332160050
Fakultas/Jurusan	: Teknik/Teknik Elektro

Telah diuji dan dipertahankan pada tanggal …. …... 2023 melalui Sidang Skripsi di Fakultas Teknik Universitas Sultan Ageng Tirtayasa Cilegon dan dinyatakan LULUS.

Dewan Penguji

		Tanda Tangan
Pembimbing I	: Rian Fahrizal, S.T., M.Eng.	.....
Pembimbing II	: Cakra Adipura Wicaksana, S.T., M.T.	.....
Penguji I	: Dr. Ing. M. Iman Santoso, M.Sc.	.....
Penguji II	: Fadil Muhammad, S.T., M.T.	.....

Mengetahui,
Ketua Jurusan

Dr. Romi Wiryadinata, S.T., M.Eng.
NIP.198307032009121006

PRAKATA

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini, Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Sultan Ageng Tirtayasa. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

(1) Kedua orang tua tercinta serta seluruh keluarga yang telah memberikan nasehat, semangat, doa, dan materi yang tak terhingga nilainya.
(2) Dr. Romi Wiryadinata, S.T., M.Eng., selaku Ketua Jurusan Teknik Elektro.
(3) Rian Fahrizal, S.T., M.Eng. dan Cakra Adipura W, S.T., M.T. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
(4) Sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Cilegon, Juli 2023

Penulis

ABSTRAK

Adam Abdul Malik

Teknik Elektro

Identifikasi Wajah Manusia Menggunakan Yolo *Frameworks* Dengan Metode *Scale Modifier* Sebagai *Preprocessing* Secara *Real Time*

Perkembangan teknologi meningkat sangat pesat di era sekarang ini, penelitian ini merancang sebuah sistem yang menggunakan metode *You only look once* (YOLO) untuk pengenalan wajah. Model YOLO harus melewati proses *training* untuk identifikasi wajah manusia. Proses *training* dilakukan di Google Colaboratory. Hasil dari proses *training* kemudian di optimalisasi sehingga akan memperkecil ukuran *file*, Setelah model YOLOv4.*weight* dibuat maka dapat membuat program untuk identifikasi wajah manusia secara *realtime* menggunakan *You only look once* (YOLO). Dalam deteksi wajah memiliki tingkat akurasi sebasar 94,193% dengan sudut kamera berada di posisi depan dengan total data wajah sebanyak 20 dan hasil FPS yang didapat rata-rata 2,8 FPS.

Kata Kunci: Identifikasi wajah manusia, *You Only Look Once* (YOLO), Python, Google Colab.

ABSTRACT

Adam Abdul Malik

Electrical Engineering

Human Face Identification Using Yolo Frameworks with Scale Modifier Method as Real Time Preprocessing

.

Technological developments have increased very rapidly in today's era, this study designed a system that uses the You only look once (YOLO) method for facial recognition. YOLO models must be trained in human facial identification. The training process is carried out in Google Colaboratory. The results of the training process are then optimized so that it will reduce the file size. After the YOLOv4.weight model is created, a program for real-time identification of human faces can be created using You only look once (YOLO). In face detection it has an accuracy rate of 94.193% with the camera angle in the front position with a total of 20 face data and the FPS results obtained on average 2,8 FPS.

Keywords: Human Face Identification, *You Only Look Once* (YOLO), Python, GoogleColab.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Selama dua puluh tahun silam, visi komputer telah menerima banyak liputan. Pelacakan objek visual adalah satu dari bidang visi komputer yang paling penting untuk objek tertentu contohnya seperti manusia, binatang, kendaraan, atau pun seperti bangunan dalam objek gambar dan hasil foto atau sebuah video. Tujuan lain dari sebuah pendeteksi objek ialah untuk mempelajari lebih lanjut model komputasi yang menyediakan informasi awal paling mendasar yang dibutuhkan oleh aplikasi visi komputer. Pengenalan objek ialah hal yang sangat penting dikarenakan proses nya banyak membuat dasar dari untuk banyak tugas visi komputer lainnya. Contonya seperti segmentasi instan, cerita gambar, objek pelacakan, dan masih banyak lagi contoh yang lainnya. [1].

Pelacakan adalah proses pelacakan dari waktu ke waktu objek yang bergerak atau beberapa objek. Tujuan pelacakan objek visual di bingkai video secara *real-time* adalah untuk mendeteksi atau menghubungkan objek target. Dalam sebuah sistem pencarian objek yang berdasarkan pada visual mampu mencari sejumlah variabel objek dalam ruang yang dinamis dan mempertahankan keaslian dari sebuah objek target itu sendiri dengan tepat tanpa memperhatikan hambatan dan beberapa gangguan visual yang lain. Penjejakan objek yang berbasis visual ini mempunyai masalah yang menarik untuk lebih dikembangkan kedepannya [2][3]. Analisis pelacakan oleh deteksi pendekatan yang mencakup deteksi oleh YOLO [4].

Satu dari beberapa kasus yang lebih banyak digunakan dalam pendeteksian dan pengidentifikasian objek ialah menggunakan algoritma *Convolutional Neural Network* (CNN). CNN adalah salah satu algoritma yang sering digunakan untuk kepentingan deteksi objek, salah satu nya ialah karena CNN didukung oleh *framework* Tensorflow buatan dari Google, tapi ternyata ada salah satu algoritma objek deteksi yang mempunyai tingkat akurasi yang lebih tinggi, dan kecepatan mengenali objek lebih baik yaitu *You Only Look Once* (YOLO) [5][6].

YOLO sendiri kali pertama ditemukan oleh Joseph Redmon pada tahun 2015 ialah sistem pendeteksi objek secara *real time* berbasis CNN (*Convolutional Neural Network*) [7]. Dalam pertemuan CVPR (*Conference on Computer Vision and Pattern Recognition*) merilis YOLOv2 dan lalu ditingkatkan akurasi dan kecepatan algoritma, dengan teknologi yang semakin berkembang selanjutnya YOLO v3 dirilis dan mempunyai performa dan kinerja yang semakin baik pada suatu pendeteksian objek [8].

YOLO memisahkan sebuah video ataupun gambar lalu dimasukan menjadi SxS *grid* [9]. Ketika titik tengah koordinat pada *ground truth (GT)* suatu objek masuk ke dalam *grid*, lalu *grid* bertugas untuk mendeteksi objek tersebut. YOLO memperkirakan *bounding box* suatu objek yang ada di bagian *grid*, kemungkinan tempat dan peluang dari semua kelas pada waktu yang sama [10].

Reformasi YOLO ialah membuat *framework* dari *region* gagasan deteksi seri R-CNN untuk menghasilkan gagasan dimana untuk melengkapi proses klasifikasi dan analisis [11]. Namun ada *overlap* dimana wilayah gagasan akan menjadi proses yang berulang. Tetapi YOLO memperkirakan bagian *bounding box* dari objek yang ada di dalam *grid*, probabilitas suatu lokasi dari semua kelas pada waktu yang sama, dengan itu kelebihan YOLO ialah *frameworks* yang bisa menangani *problem* dengan sekali proses [12]. Tujuan utama dari model YOLO ialah untuk membuat model algoritma yang bisa mengidentifikasi dan mendeteksi objek dengan cepat dan tepat tanpa mengorbankan tingkat akurasinya.

Framework ialah rancangan kerja yang digunakan untuk mempermudah para pihak pengembang perangkat lunak dalam mendesain dan mengembangkan suatu aplikasi [13]. *Framework* terdapat perintah dan fungsi utama untuk digunakan dalam merancang sebuah perangkat lunak aplikasi dan berharap aplikasi tersebut dapat dirancang dengan lebih cepat serta tersusun dan tertata dengan baik [14]. *Framework* bisa sebagai suatu komponen yang jadi dan bisa untuk digunakan dimana saja, oleh karenanya pihak pengembang tidak harus membuat ulang *script* lagi untuk desain yang sama.

Deep learning adalah suatu bagian metode *machine learning* yang membuat suatu algoritma sistem mampu untuk belajar dan mengenali dan dengan sendirinya dengan suatu data yang sudah tersedia dan pengalaman yang dialami tanpa perlu

peran manusia secara berlebihan [15]. Suatu *deep learning* sebetulnya banyak sekali *framework* yang disediakan contohnya *Tensorflow*, *Caffe2*, dan *CTNK*. Untuk dapat memakai arsitektur YOLO dapat menggunakan *framework darknet*, *darkflow*, maupun *OpenCV*.

1.2 Rumusan Masalah

Rumusan suatu masalah dalam penelitian ini adalah:

1. Bagaimana membangun suatu sistem pengidentifikasian wajah manusia dengan menggunakan YOLO *frameworks*?
2. Bagaimana proses kerja dari pendeteksian YOLO *frameworks* untuk mengidentifikasi wajah manusia dalam sudut posisi depan wajah yang melewati perekaman video berbasis *face detection*?
3. Bagaimana tingkat akurasi pendeteksian wajah manusia menggunakan YOLO *frameworks*?

1.3 Tujuan Penelitian

Tujuan permasalahan dalam skripsi ini yaitu:

1. Mengetahui proses perancangan sistem identifikasi wajah manusia dengan menggunakan YOLO *frameworks*.
2. Mengetahui proses kerja dari pendeteksian YOLO *frameworks* untuk mengidentifikasi wajah manusia dalam sudut yang di analisis yang melewati perekaman video berbasis *face detection*.
3. Mengetahui tingkat akurasi pendeteksian identifikasi wajah manusia menggunakan YOLO *frameworks*.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi akademisi, hasil penelitian ini diharapkan sangat membantu para akademisi yang ingin melakukan penelitian lebih lanjut dalam bidang sistem deteksi pada orang yang dikenali oleh sistem.
2. Bagi pendidikan, hasil penelitian ini diharapkan menambah informasi dan pengetahuan mengenai sistem deteksi menggunakan YOLO *frameworks*.

3. Bagi keberlanjutan pendidikan, hasil penelitian ini diharapkan membantu dan mendorong penelitian mengenai teknologi dibidang *computer vision*.
4. Merujuk pada perumusan masalah dan tujuan penelitian diatas manfaat dari penelitian ini ialah untuk membuat program yang dapat mendeteksi wajah manusia yang telacak melalui sistem perekaman video secara langsung.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Menggunakan YOLOv4 *framework* untuk pendeteksian wajah.
2. Identifikasi posisi depan wajah manusia melalui perekaman video.
3. Menggunakan bahasa Pemrograman Python.
4. Menggunakan laptop dengan GPU GTX 930m.
5. Sudut yang dapat ditangkap sistem perekaman video hanya bagian sisi depan wajah.
6. mengidentifikasi 20 wajah manusia.

1.6 Sistematika Penulisan

Gambaran mengenai penjelasan penelitian ini dibagi beberapa pokok permasalahan. Adapun sistematika pada penelitian ini sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang hal yang terkait dengan penelitian dan perencanaan dalam penlisan, bab ini berisi mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi mengenai tinjauan pustaka dan dasar teori, didalamnya terdapat kajian-kajian dari hasil artikel publikasi berikut dengan judul penelitian, dan landasan-landasan teori yang memfasilitasi dalam menyelesaikan penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan mengenai proses perancangan alat, cara kerja alat, perangkat dan spesifikasi alat yang digunakan dalam pembuatan alat, baik

perangkat keras ataupun perangkat lunak , serta waktu dan tempat penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan tentang penjelasan mengenai hasil dari penelitian serta analisis terhadap hasil penelitian yang dikaitkan dengan tinjauan pustaka dan metodologi penelitian.

BAB V PENUTUP

Bab ini berisikan kesimpulan dari penelitian ini dan saran untuk pengembangan penelitian yang akan datang.

BAB II
TINJAUAN PUSTAKA

2.1. Sistem *Real-Time*

Suatu sistem yang dapat mengendalikan sistem fisik dikenal sebagai sistem *real-time* [16]. Sistem *real-time* merupakan bentuk khusus dari sistem *online*. Sebuah sistem dinyatakan *real-time* jika tidak selalu mengedepankan ketepatan suatu proses, tetapi juga dengan jeda atau jarak waktu proses tersebut dilakukan. Sehingga suatu sistem *real-time* ialah sistem yang menggunakan sistem *deadline*, ialah pekerjaan yang semestinya beres dengan periode waktu yang diberikan. Pada sistem *real-time*, digunakan batasan waktu. Sistem dinyatakan gagal jika melewati batasan yang ada. Sistem *real-time* banyak diterapkan dalam berbagai macam kebutuhan kita sehari-hari contohnya sistem tersebut dimasukan di dalam alat khusus seperti di kamera telepon genggam, pemutar musik, ataupun di mobil dan pesawat [16].

Terdapat dua bentuk sistem *real-time*, yaitu sistem *hard real-time* dan sistem *soft real-time*. Pada *hard real-time*, sistem menanggung tugas yang berat akan diselesaikan secara cepat. Dalam sistem ini penyimpan kedua terbatas atau tidak terpakai, data langsung dikirim ke *memory* atau *read-only memory* (ROM) dalam waktu yang cepat. Dalam *hard real-time* sistem terjadi masalah pada sistem pembagian waktu dan tidak ditunjang oleh sistem operasi tujuan umum. Gambaran lainnya ialah *soft real-time* dimana *task* yang berat memperoleh pengetumaan yang lebih tinggi dibandingkan dengan tugas lain dan setelah tugas selesai maka tugas yang diutamakan ini akan diselesaikan. Sistem ini tidak banyak dijumpai pada bidang industri robotik. Sistem *real-time* sangat bermanfaat pada aplikasi komunikasi dan realita *virtual* [16].

Suatu *system* dikatakan *real-time* jika *system* tersebut dapat mencakup pengerjaan program atau aplikasi dengan waktu yang yang telah ditetapkan. Kelebihan suatu sistem *real-time*, yaitu:

a. Mempunyai ketentuan waktu dan mencukupi tenggang waktu.
b. Memiliki waktu yang bisa diperkirakan.

c. Terfokus pada hal yang utama saja, sehingga dapat menemukan tingkat effisiensi waktu.

Selain kelebihan, adapun beberapa kekurangan *real-time system*, yaitu:

a. Mempunyai hanya satu ara tujuan, contohnya memindahkan satu lagu dari perangkat laptop ke pemutar musik.
b. Biasanya sistem mempunyai ruang fisik yang terbatas.
c. Metode ini harus memenuhi persyaratan saat yang ditentukan dengan menggunakan algoritma.

2.2. Bahasa Pemrograman Python

Python diciptakan oleh Guido Van Rossum pertama kali di *Scitchting Mathematisch Centrum* (CWI) di Belanda pada awal tahun 1990-an. Bahasa Python termotivasi dari bahasa pemrograman ABC. Sampai saat ini, Guido masih menjadi pembuat utama untuk Python, walaupun *open source* dan bisa semua orang juga berpartisipasi dalam pengembangannya. Pada tahun 1995, Guido meneruskan pengerjaan Python di *Corporation for National Research Initiative* (CNRI) di Virginia Amerika.

Pada bulan mei tahun 2000, Guido beserta rekannya pindah ke *BeOpen.com* dan membuat tim *BeOpen* Python*Labs*. Pada bulan oktober tahun 2000, regu Python pindah ke kreasi digital dan lebih dikenal menjadi Perusahaan Zope. Di tahun 2001, dibuatlah sebuah kelompok Python yaitu Python *Software Foundation* (PSF). PSF adalah sekelompok nirlaba yang rancang istimewa agar bisa dalam semua hal yang bersangkutan dengan hak intelektual Python. Perusahaan Zope menjadi anggota pendukung dari PSF [17].

Python adalah bahasa pemrograman yang mempunyai pandagan mempunyai banyak kegunaan dengan filsafat proses yang bertujuan utama pada pemahaman kode secara cepat. Python diakui sebagai bahasa yang menyatukan keahlian dengan sistem kode yang jelas, Python pun di sponsori oleh organisasi yang besar.

Python bisa mempunyai banyak pembuatan rangka dari awal pemrograman, khususnya tetapi tidak terbatas pada pemrograman berfokus objek, bahasa yang mempunyai perintah, dan bahaya yang fungsional. Beberapa fitur yang ada pada Python adalah salah satu bahasa pemrograman dinamis yang dilengkapi pengaturan

penyimpanan otomatis. Sama kasusnya pada bahasa pemrograman dinamis lainnya, Python kebanyakan dipakai layaknya bahasa *script* walaupun pada kenyataannya pemakaian bahasa ini lebih luas pembahasan makna manfaat yang tidak dilakukan dengan menggunakan bahasa *script*. Python bisa berfungsi untuk beberapa kepentingan pengembangan *software* dan dapat berjalan di berbagai penyedia program operasi sistem [18]. Python memiliki keunggulan dibandingkan bahasa pemrograman lainnya, berikut adalah keunggulannya:

1. Python mempunyai gagasan bentuk yang baik dan mudah digunakan, kode Python dibuat lebih gampang untuk dipahami, dijalankan, dipakai secara berulang, dan dijaga. Di luar dari itu, Python lebih mendorong pemograman bertemakan suatu objek dan pemograman bisa berfungsi sebagai apapun.

2 Python mampu menambah kemampuan dan ketepatan waktu untuk penggunanya, untuk mendapatkan hasil kode yang sesuai, kode Python mempunyai lebih sedikit kode yang dibuat dibandingkan memakai pemograman bahasa yang lain misalnya Java maupun *C, C++*, dan *C#.*

3. Program yang dibuat melalui Python dapat diterapkan di kebanyakan sistem operasi seperti Unix, Windows, Mac *OS X*, dan juga perangkat komunikasi lainnya.

4. Python mempunyai banyak bantuan pustaka yang lebih di analisis oleh pihak lain, contohnya rujukan untuk pengolahan situs, pemutakhiran aplikasi visual berintegrasikan GUI, penelitian lebih lanjut dalam suatu permainan, dan lain-lain.

5. Menggunakan intruksi tertentu, kode Python bisa digabungkan dengan aplikasi yang dibuat di bahasa pemrograman lain. Seperti hal nya kode Python bisa dijalankan dari kode *C/C++*, dan tak lepas dari perkembangan .NET *framework.*

6. Python dapat digunakan secara gratis dan *open source*, walaupun dipakai untuk kebutuhan pemasaran [19].

Terdapat hal penting dalam Python yaitu penyunting naskah. Penyunting naskah ialah perangkat lunak program komputer yang menjadikan pemakainya untuk menciptakan, menyunting berkas *text* dalam rangka susunan teks. Penyunting teks dapat dipakai untuk menciptakan aplikasi dan menyunting sumber kode dari bahasa pemograman. Penyunting teks dapat digunakan dalam menyusun halaman situs dan dapat menciptakan program. Kegunaan penyunting naskah antara lain:

1. Menemukan kalimat atau bahasa dalam suatu berkas.

2. Membuat, mengeksekusi, dan untuk mencari serta mengurangi kesalahan dalam suatu aplikasi.
3. Mengelompokan data yang yang tersedia.
4. Memadukan berkas satu dan yang lain.

Python memiliki penyunting naskahnya nya sendiri yaitu IDLE. *Integrated DeveLopment Environment* (IDLE) adalah penyunting yang tersedia di Python yang dapat di pakai untuk membuat Python kode. IDLE mempunyai mode *shell* atau interaksi mode yang dapat dipakai untuk melakukan suatu tugas dengan hasil output yang bisa langsung melihat hasilnya. IDLE pun ialah asli diciptakan oleh Guido van Rossum dari bahasa Python. IDLE mempunyai kelebihan yang sangat bermanfaat bagi penggunanya, diantaranya:

1. *Editor multi-window* dengan fitur *syntax highlighting*, *autocomplete*, *smart indent*, dan lain–lain.
2. *Shell* interaksi dengan kelebihannya *syntax highlighting*.
3. Mempunyai kelebihan *debugger* agar bisa *debug listing* program [20].

2.3. *Image Processing*

Image processing atau pengolahan citra merupakan mekanisme pengolahan gambar yang merubah gambar *input* sebagai gambar yang berbeda lain supaya keluarannya mempunyai mutu yang lebih bagus dibandin mutu gambar *input*. Hal tersebut dapat berguna sebagai syarat menaikkan mutu sebuah gambar, menghapuskan kerusakan terhadap gambar, mengetahui objek, serta menjadikan satu oleh kelompok gambar lainnya. Saat ini, *image processing* adalah satu dari banyak pengetauan metode ilmiah yang bertambah cepat dan tidak bisa dipisahkan oleh bagian visi komputer [21].

Kualitas gambar masukan merupakan peran penting dalam keberhasilan komputasi gambar, karena semakin tinggi kualitas gambarnya, maka lebih mudah komputasi dijalankan. Jadi untuk meningkatkan kualitas gambar masukan, metode komputasi yang tepat digunakan ialah *image processing*. Langkah-langkah yang diperlukan seperti menghilangkan *noise*, mengoreksi pada geometris, meningkatkan tepi dan kontras, serta mengoreksi iluminasinya [21].

2.4 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah perpustakaan *machine learning* dan bersifat bebas sumber. OpenCV versi pertama diluncurkan pada tahun 1999, asal mula nya mementingkan perpustakaan dari Intel *Image Processing Library* yang lalu ketergantungan sebagainya dilenyapkan agar terbentuklah OpenCV layaknya seperti perpustakaan yang berdiri sendiri. OpenCV menunjang banyak program, bisa menunjang seperti Linux ataupun Windows, pada saat ini sudah bisa menunjang Android dan MacOSX. *Library* OpenCV amat banyak digunakan untuk pemrosesan citra karena mempunyai sangat banyak algoritma, yaitu sebesar 2500 algoritma, dimana algoritma itu dapat di pakai sebagai mengolah citra mulai dari deteksi wajah, pengenalan wajah, *object tracking*, deteksi tepi hingga *Kalman filtering*. OpenCV juga merupakan *library open source* yang digunakan untuk bahasa C++, Python, Java, dan Matlab. OpenCV mempunyai puluhan ribu orang sebagai dari sekelompok pengguna yang dapat diperkiraan lebih dari belasan juta kali diunduh [22]. OpenCV memiliki kelebihan pada *library* OpenCV diantaranya:

1. Membuat gambar data (tempat penyimpanan, menghilangkan penyimpanan, menyalin citra, pengaturan dan merubah citra).
2. Gambar/Vidio.
3. Sampling gambar dan transformasi.
4. Metode untuk AI dan *machine learning*.
5 Membuat matrikulasi dan besaran dan ada pada *routines linear algebra* (produk, pemecah masalah, nilai eigen, SVD).
6. *Basic* pemrosesan gambar seperti penyaringan, deteksi tepi, pengenalan sisi, proses seleksi hingga pengalihan, perubahan warna, membuat struktural, grafik statistik dan gambar piramida.
7. Analisa struktur.
8. Mengatur potret.
9. Mengetahui gerakan.
10. Mengidentifikasi hal.
11. Dasar antarmuka pengguna seperti citra, rekaman, papan ketik dan lainnya.
12. Penandaan gambar seperti garis, kerucut, polygon, dan gambar teks.

2.5. Deteksi Objek

Ketika manusia melihat suatu objek, kemampuan kita untuk dengan cepat mengenali gambar dalam citra, menentukan posisi objek tersebut, dan memahami situasi yang sedang berlangsung adalah salah satu aspek yang sangat kuat dari kemampuan visual kita. Basis visual yang dimiliki oleh orang yang terampil dan tepat dalam pengamatan ini memungkinkan kita untuk menjalankan tugas yang kompleks, bahkan dalam situasi yang sulit sekalipun. [23]. Dengan adanya algoritma deteksi objek yang canggih, komputer dapat digunakan dalam berbagai aplikasi, termasuk pengawasan keamanan, kendaraan otonom, pengolahan medis, pengenalan wajah, dan banyak lagi. Kemampuan ini dapat meningkatkan efisiensi dan akurasi dalam berbagai tugas, memungkinkan komputer untuk bekerja dengan citra dan video dalam konteks yang lebih mendalam dan bermakna.

Proses ini sering melibatkan teknik-teknik seperti ekstraksi fitur, pemodelan statistik, penggunaan algoritma pembelajaran mesin, atau jaringan saraf tiruan dalam kasus algoritma deteksi objek yang telah dibahas sebelumnya. Tujuan utamanya adalah untuk mengidentifikasi dan menentukan lokasi serta karakteristik objek dengan akurasi tinggi dalam citra atau video.

2.6. Tensorflow *Lite*

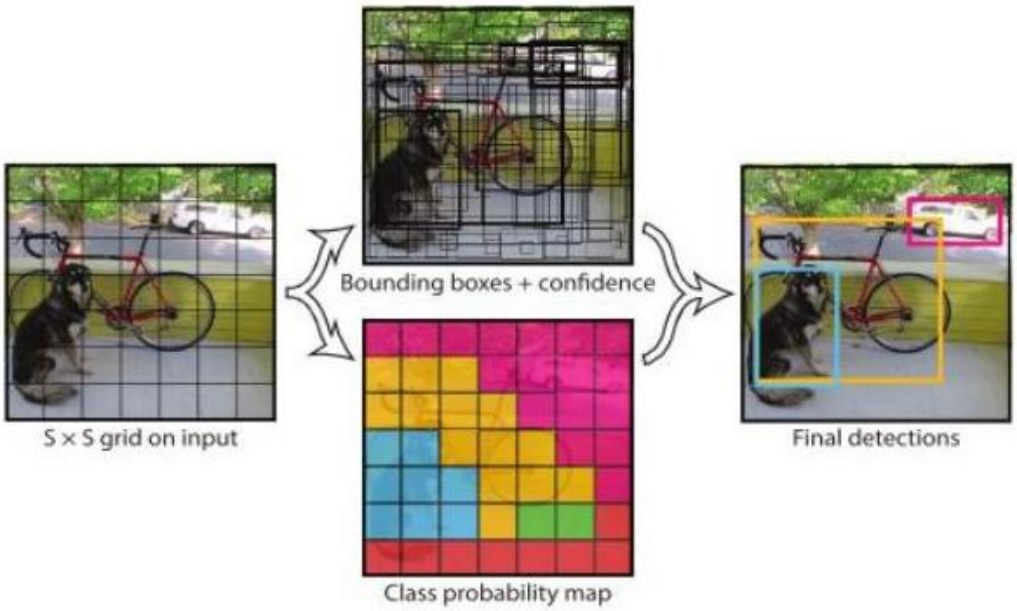
TensorFlow Lite memang merupakan antarmuka yang digunakan untuk mengekspresikan algoritma pembelajaran mesin dan untuk menjalankan perintah dengan menggunakan informasi yang dimiliki tentang objek atau target yang dikenali. Ini memungkinkan model pembelajaran mesin yang telah dihasilkan sebelumnya untuk dijalankan pada perangkat dengan sumber daya terbatas seperti perangkat seluler atau mikrokontroler [24] Selain itu, TensorFlow, versi utama dari framework TensorFlow, memiliki kemampuan untuk melakukan pelatihan model pembelajaran mesin menggunakan CPU dan GPU. Penggunaan GPU dalam pelatihan model dapat menghasilkan waktu pelatihan yang lebih cepat dibandingkan dengan CPU, karena GPU memiliki arsitektur yang sangat cocok untuk tugas-tugas yang melibatkan operasi matriks paralel yang umum dalam pembelajaran mesin. Hal ini memungkinkan para peneliti dan praktisi untuk melatih

model yang lebih besar dan kompleks dalam waktu yang lebih singkat, meningkatkan produktivitas dalam pengembangan model pembelajaran mesin..

2.7. *You Only Look Once* (YOLO)

YOLO adalah sebuah algoritma yang dikembangkan khusus untuk mendeteksi objek secara real-time dalam citra atau video. Pendekatan yang digunakan oleh YOLO berbeda dengan pendekatan lain yang memerlukan beberapa tahapan, seperti mengidentifikasi objek pertama dan kemudian melakukan perbaikan. Sebaliknya, YOLO menggabungkan tahapan-tahapan tersebut menjadi satu proses yang dilakukan secara bersamaan [6].

Dalam YOLO, citra input dibagi menjadi grid berukuran S×S. Dalam kasus yang Anda sebutkan, nilai S adalah 7, dan citra input berukuran 448 x 448 pixel. Setiap sel dalam grid bertanggung jawab untuk memprediksi beberapa bounding box (kotak pembatas) yang mungkin mengandung objek. Umumnya, ada 2 bounding box yang diprediksi dalam satu sel grid. Oleh karena itu, untuk setiap sel, kita memiliki 2*B bounding box, di mana B adalah jumlah bounding box yang diprediksi dalam satu sel biasanya 2 [25]. Proses kerja YOLO ditampilkan pada Gambar 2.1 berikut:



Gambar 2.1 Proses Kerja YOLO [25]

Berdasarkan Gambar 2.1 Dalam YOLO, normalisasi atribut pada bounding box diperlukan untuk memastikan bahwa semua nilai berada dalam rentang yang seragam, yaitu antara 0 hingga 1. Ini membantu dalam pelatihan dan pemrosesan selanjutnya Normalisasi ini penting karena membuat representasi bounding box seragam dan independen terhadap resolusi gambar asli, sehingga memudahkan proses pelatihan dan inferensi. Dengan normalisasi ini, model YOLO dapat memprediksi bounding box dengan koordinat dan ukuran yang dapat diinterpretasikan dengan baik dalam berbagai ukuran citra [11].

2.8. *Deep Learning*

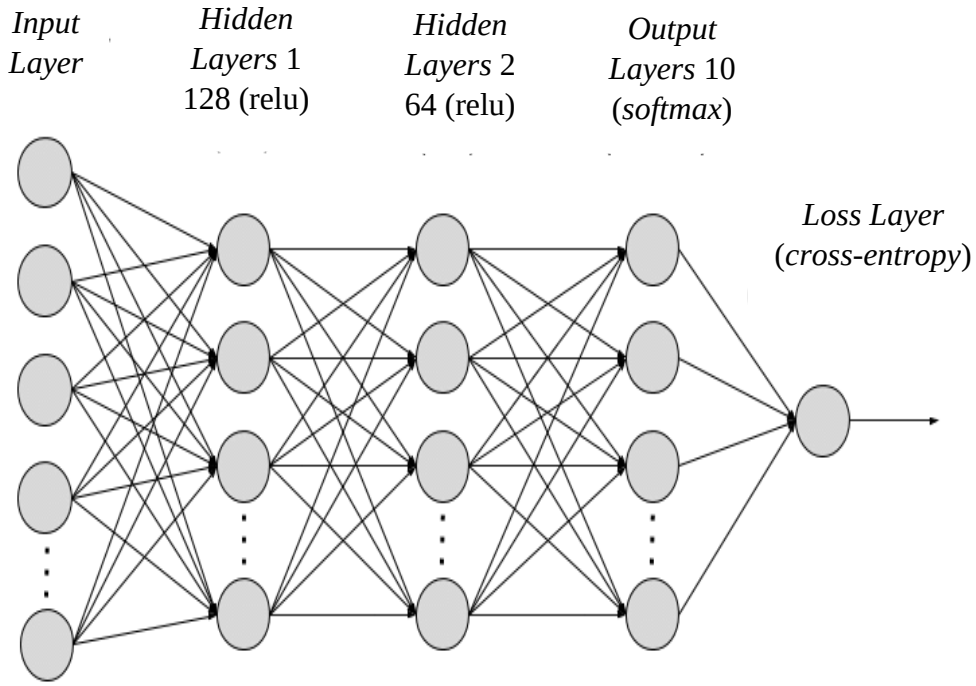
Teknologi deep learning memungkinkan komputer untuk memahami data yang lebih kompleks, yang sering kali sulit diproses dengan metode tradisional. Ini membuka peluang untuk berbagai aplikasi kecerdasan buatan yang dapat membantu dalam menjalankan tugas-tugas yang sebelumnya membutuhkan kecerdasan manusia. Beberapa contoh aplikasi deep learning yang umum digunakan termasuk:

1. Pengenalan Gambar: Deep learning digunakan untuk mengenali objek dalam gambar, seperti dalam sistem pengenalan wajah, deteksi objek, dan klasifikasi gambar.
2. Penerjemahan Bahasa: Model deep learning digunakan dalam sistem penerjemahan bahasa otomatis, yang dapat menerjemahkan teks dari satu bahasa ke bahasa lainnya.
3. Pemrosesan Suara: Deep learning digunakan dalam aplikasi pemrosesan suara, seperti pengenalan ucapan, transkripsi otomatis, dan asisten suara.
4. Mendeteksi Pola: Deep learning dapat digunakan untuk mendeteksi pola kompleks dalam data, seperti prediksi harga saham, analisis sentimen dalam teks, dan pemrosesan sinyal medis.
5. Otomatisasi Tugas: Model deep learning dapat digunakan untuk otomatisasi tugas-tugas yang repetitif, seperti penggantian teks dalam gambar atau mengkategorikan email spam.

Dengan kemampuan deep learning untuk mengenali pola yang rumit dalam data, teknologi ini telah menjadi sangat berharga dalam berbagai bidang, termasuk teknologi medis, otomotif, penerbangan, dan banyak lagi. Ini terus mendorong

perkembangan aplikasi kecerdasan buatan yang canggih dan memengaruhi berbagai aspek kehidupan [29].

Seperti otak manusia, deep learning juga menggunakan proses pembelajaran untuk mengatur bobot dan parameter di antara neuron-neuron dalam jaringan. Selama pelatihan, jaringan neural "menggunakan" data pelatihan untuk mengoptimalkan bobot-bobot ini sehingga dapat melakukan tugas tertentu, seperti pengenalan pola atau klasifikasi. Jaringan neuron buatan adalah algoritma *deep learning* yang menggunakan simpul ini untuk memecahkan masalah kompleks. Berikut adalah komponen jaringan *deep learning* pada Gambar 2.2.



Gambar 2.2 Komponen Jaringan *Deep Learning* [29]

Berdasarkan Gambar 2.2 dapat dilihat komponen jaringan *deep learning* memiliki beberapa lapisan diantaranya ialah Jaringan neural deep learning memiliki banyak lapisan tersembunyi, yang dapat bervariasi dalam jumlah tergantung pada arsitektur jaringan. Jumlah lapisan tersembunyi yang dalam atau kompleks dapat memungkinkan jaringan untuk menganalisis masalah dari berbagai sudut pandang, memungkinkan pembelajaran fitur yang lebih abstrak dan pemecahan masalah yang lebih rumit. Namun, penggunaan lapisan tersembunyi yang lebih dalam juga dapat meningkatkan kompleksitas dan kebutuhan komputasi dalam pelatihan jaringan.

Oleh karena itu, desain arsitektur jaringan deep learning harus mempertimbangkan trade-off antara kedalaman dan kinerja

Namun, dalam tugas klasifikasi yang lebih kompleks dengan banyak kategori yang berbeda misalnya, klasifikasi gambar hewan menjadi berbagai jenis hewan, jumlah simpul di lapisan output akan sesuai dengan jumlah kategori tersebut. Setiap simpul dalam lapisan output mewakili probabilitas atau skor kategori tertentu. Dalam kasus ini, model deep learning mencoba untuk menghasilkan prediksi yang lebih rinci dengan memberikan skor untuk setiap kategori yang mungkin, dan kategori dengan skor tertinggi akan menjadi prediksi akhir.

2.9. *Scale Modifier*

Scale modifier bertujuan untuk mengurangi atau mengecilkan ukuran dimensi dari gambar yang dipilih. Biasanya gambar yang di pangkas memiliki ukuran yang besar oleh karna itu diperlukan *scale modifier* untuk memunculkan ukuran gambar yang lebih kecil [26]. Efek menggunakan *scale modifier* adalah akan memperoleh tingkat akurasi yang lebih baik karna dapat mendeteksi gambar pada kondisi yang kecil sehingga YOLO dapat dengan lebih mudah mendeteksi objek yang sudah kita tentukan atau *training* terlebih dahulu datanya.

2.10. Optimasi *Model*

Optimasi model merupakan sebuah cara yang bertujuan mengurangi ukuran *file*, mengurangi latensi, dan mengkompabilitasi *akselerator*. Pada dasarnya, optimasi model bekerja dengan mengurangi ketepatan dari angka yang digunakan untuk merepresentasikan parameter model [27]. Efek yang terjadi saat mengoptimasi model ialah model ukurannya menjadi lebih sama dan cara untuk memperoleh hasil yang terbaik.

2.11. Google Colaboratory

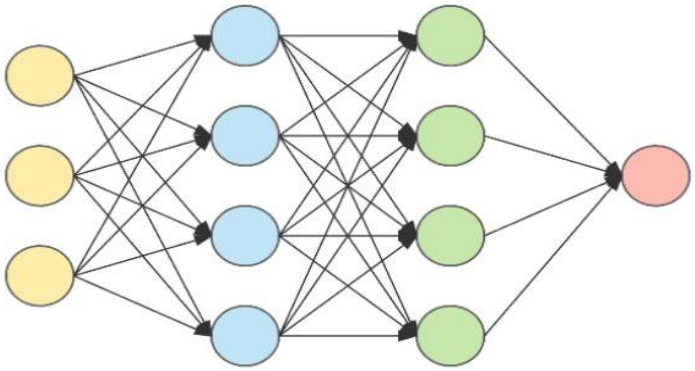
Google Colaboratory adalah sebuah proyek yang bertujuan untuk menyebarkan pendidikan dan penelitian *machine learning* [28]. Buku catatan kolaboratoris didasarkan pada Jupyter dan berfungsi sebagai objek Google

Dokumen dapat dibagikan dan pengguna dapat berkolaborasi di buku catatan yang sama. Colaboratory menyediakan *runtime* Python 2 dan 3 yang telah dikonfigurasi sebelumnya dengan pustaka *machine learning* dan *artificial intelligence* yang penting, seperti TensorFlow, Matplotlib, dan Keras. Mesin virtual di bawah *runtime* (VM) dinonaktifkan setelah jangka waktu tertentu, dan semua data dan konfigurasi pengguna hilang. Namun, *notebook* tersebut dipertahankan, dan juga memungkinkan untuk mentransfer *file* dari *hard disk* VM ke akun Google Drive pengguna [29].

Google Colaboratory mempunyai 4 virtual GPU yaitu Nvidia Tesla K80s, T4s, P4s, dan P100s. Hanya GPU Nvidia Tesla K80s yang dapat digunakan secara gratis, sisanya harus menggunakan Google Colaboratory yang versi berbayar. Dalam penggunakan Google Colaboratory yang gratis disediakan juga *memory* GPU sebesar 12 GB, RAM sebesar 25 GB, dan *termporary disk* sebesar 107 GB.

2.12. *Artificial Neural Network*

Artificial Neural Network bekerja dengan mengalirkan informasi dari *layer* masukan melalui *layer* tersembunyi ke layer keluaran. Hubungan antara neuron-neuron di seluruh jaringan memiliki bobot yang disesuaikan selama pelatihan jaringan untuk mengoptimalkan hasil prediksi [30]. Contoh penggambaran model *artificial neural network* dapat dilihat pada Gambar 2.3.



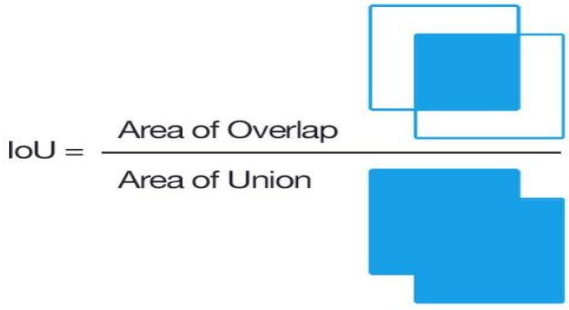
Gambar 2.3 Model Dasar *Artificial Neural Network* [31]

Berdasarkan Gambar 2.3 dapat diihat model dasar jaringan syarat tiruan memiliki lapisan *input* biasanya dalam bentuk vektor multidimensi ke lapisan *input*

yang akan mendistribusikannya ke lapisan tersembunyi, lapisan tersembunyi kemudian akan membuat keputusan dari lapisan sebelumnya dan mempertimbangkan bagaimana perubahan statistik di dalam dirinya dan meningkatkan hasil akhir, dan ini disebut sebagai lapisan *output*. Kemudian ada Jaringan Syaraf Tiruan (JST) yang neuron di dalam *layer*-nya tersusun menjadi 3 dimensi ialah *Convolutional Neural Network* (CNN) [31]. CNN telah mencapai kesuksesan besar dalam sejumlah tugas pengolahan citra, termasuk keberhasilan dalam kompetisi seperti ImageNet, di mana CNN telah menjadi arsitektur utama yang digunakan dalam klasifikasi gambar. Namun, kegunaan CNN tidak terbatas pada citra saja; mereka juga telah diterapkan dalam berbagai bidang pemrosesan data spasial, termasuk pemrosesan sinyal suara dan pemrosesan video.

2.13. *Intersection Over Union*

Intersection Over Union (IoU) adalah metrik yang penting dalam mengevaluasi kinerja model deteksi objek, dan sering digunakan dalam pelatihan model dengan mengukur sejauh mana prediksi bounding box mendekati ground-truth. Nilai IoU yang diharapkan dapat bervariasi tergantung pada tugas dan datasetnya, tetapi seringkali ada batasan IoU minimum yang didefinisikan sebagai ambang batas untuk menganggap deteksi berhasil atau tidak berhasil[32]. Ilustrasi persamaan IoU dapat dilihat pada Gambar 2.4.



Gambar 2.4 *Intersection Over Union* [32]

Berdasarkan Gambar 2.4 dapat dilihat persamaan *Intersection over Union (IoU)* adalah titik temu atas dan *union* hanyalah sebuah rasio. Di pembilang menghitung area *overlap* antara kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar. Penyebutnya adalah area *union*, atau lebih

sederhananya, area yang dicakup oleh kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar. Menghitung melalui membagi area atas antara kotak pembatas dengan area *union* menghasilkan nilai akhir.

2.14. *Confusion Matrix*

Confusion matrix adalah tabel yang menggambarkan empat kondisi yang dapat muncul saat sebuah model klasifikasi memproses data[33]. *Confusion Matrix* dan metrik yang dihitung dari tabel ini sangat berguna dalam mengevaluasi sejauh mana model klasifikasi mampu melakukan tugasnya dan mengidentifikasi kekuatan serta kelemahan dari model tersebut Isi dari tabel *confusion matrix* ada 4, yaitu:

a. True Positive (TP): Ini adalah kondisi di mana model mengklasifikasikan data sebagai positif (ya/True), dan kenyataannya juga positif (ya/True). Ini mengindikasikan bahwa model dengan benar mengidentifikasi contoh positif.
b. True Negative (TN): Ini adalah kondisi di mana model mengklasifikasikan data sebagai negatif (tidak/False), dan kenyataannya juga negatif (tidak/False). Ini menunjukkan bahwa model dengan benar mengidentifikasi contoh negatif.
c. False Positive (FP): Ini adalah kondisi di mana model mengklasifikasikan data sebagai positif (ya/True), tetapi kenyataannya negatif (tidak/False). Ini juga dikenal sebagai kesalahan Type I atau "false alarm."
d. False Negative (FN): Ini adalah kondisi di mana model mengklasifikasikan data sebagai negatif (tidak/False), tetapi kenyataannya positif (ya/True). Ini juga dikenal sebagai kesalahan Type II atau "miss.

Berikut Persamaan yang akan digunakan berdasarkan data dari *confusion matrix* adalah akurasi, presisi, *recall* dan f-*score*.

1. Nilai akurasi mengukur sejauh mana model benar dalam klasifikasi secara keseluruhan. Nilai akurasi adalah metrik yang berguna untuk menilai seberapa baik model Anda dalam konteks umum. Namun, perlu diingat bahwa akurasi mungkin tidak selalu menjadi metrik yang paling informatif jika Anda memiliki ketidakseimbangan kelas dalam dataset. Dalam kasus tersebut, metrik lain seperti

presisi, recall, dan F1-Score dapat memberikan pandangan yang lebih baik tentang kinerja model Akurasi dapat dicari menggunakan Persamaan (2.1).

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \times 100\% \quad (2.1)$$

Berdasarkan Persamaan 2.1 nilai akurasi dapat dihitung dengan mencari nilai *true positive*, *true negative*, *false positive*, dan *false negative* antara kelas dan hasil prediksi dari modelnya.

2. *Precision* adalah mengukur sejauh mana model benar dalam mengklasifikasikan positif dan dihitung. Presisi mengukur tingkat keakuratan prediksi positif. Presisi dapat dihitung menggunakan Persamaan (2.2) berikut.

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.2)$$

Berdasarkan Persamaan 2.2 nilai presisi dapat dihitung menggunakan hasil nilai dari *true positive* yang diberikan oleh model dibagi dengan hasil nilai dari *true positive* ditambah dengan hasil nilai *false positive*.

3. F-*Score* adalah rata-rata harmonik dari presisi dan recall dan digunakan untuk mengukur keseimbangan antara keduanya Mencari nilai F-*Score* dapat diketahui melalui persamaan (2.3) berikut.

$$\text{F} - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (2.3)$$

Berdasarkan Persamaan 2.3 nilai F-*Score* didapatkan melalui perkalian nilai *precision* dan *recall* yang dikali 2 dan dibagi dengan nilai *precision* ditambah dengan hasil nilai dari *recall*.

4. *Recall* mengukur sejauh mana model dapat mendeteksi semua contoh positif. Recall dapat dicari nilainya menggunakan dengan Persamaan (2.4).

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.4}$$

Berdasarkan Persamaan 2.4 nilai recall didapatkan melalui nilai true positive dibagi dengan nilai true positive lalu dijumlah dengan nilai hasil false negative.

2.15. *Computer Vision*

Computer vision adalah bagian dari ilmu komputer yang membahas bagaimana sebuah komputer dapat melihat seperti manusia, oleh karena sangat erat kaitannya dengan penglihatan, pencahayaan menjadi faktor yang juga penting dalam hal ini [34]. Pencahayaan adalah salah satu faktor penting dalam computer vision karena dapat memengaruhi kualitas citra yang dianalisis.

Dalam pengembangan computer vision, teknik dan algoritma yang kompleks digunakan untuk mendeteksi objek, melacak gerakan, mengenali wajah, melakukan segmentasi gambar, dan banyak tugas visual lainnya. Aplikasi dari computer vision sangat luas, mulai dari penginderaan mesin untuk otomatisasi pabrik hingga kendaraan otonom, pengawasan keamanan, pengenalan karakter optik, hingga pengolahan medis.

Dengan teknologi *computer vision*, kita dapat memantau pergerakan benda dalam waktu nyata dan mengestimasi kecepatannya. Ini bisa digunakan dalam berbagai konteks, termasuk lalu lintas jalan raya, pengukuran kecepatan dalam perlombaan olahraga, pengawasan pergerakan logistik di pabrik, dan banyak aplikasi lainnya.

Selain itu, computer vision juga telah membantu dalam pengembangan teknologi kendaraan otonom yang dapat melihat dan merespons dunia di sekitarnya, serta dalam berbagai bidang lainnya yang memanfaatkan data visual untuk pengambilan keputusan. Dengan kemajuan teknologi dalam computer vision, aplikasi yang lebih canggih terus dikembangkan untuk memecahkan berbagai masalah dunia nyata

2.16. Kajian Pustaka

Dalam penulisan ilmiah, penting untuk memberikan rujukan yang tepat pada penelitian sebelumnya. Ini dapat dilakukan melalui penulisan kutipan dan bibliografi yang merinci semua sumber yang digunakan. Dengan cara ini, penelitian baru dapat dihubungkan dengan kerangka pengetahuan yang sudah ada dan memberikan kontribusi yang berarti bagi bidang tersebut. Referensi berikut menjelaskan sistem deteksi objek wajah manusia. pengenalan individu melalui identifikasi wajah telah dilakukan dengan akuisisi video, anotasi label kelas, dan pembuatan model dengan metode YOLOv5. Model terbaik untuk pelatihan pengenalan wajah 70 mahasiswa diperoleh mAP 0,5 sebesar 99,4% dengan pengaturan parameter epoch = 250, *batch size* = 16, dan *learning rate* = 0,01. Hasil akurasi rata-rata terbaik adalah 99,88% yang ditunjukkan dengan pengujian pada video yang berisi satu siswa dengan tingkat kecerahan normal [35].

Penelitian yang lain menggunakan metode *You only look once* (YOLO) untuk pengenalan wajah, yang memiliki kecepatan mendeteksi dan akurasi yang tinggi. *You only look once* (YOLO) dalam deteksi wajah memiliki tingkat akurasi 100% dengan sudut kamera berada di posisi depan, kanan, dan kiri terhadap wajah, Untuk pengenalan dan identifikasi wajah dengan berbagai sudut pandang memiliki akurasi 100%. sedangkan pengujian pengenalan dengan jarak 30-100 cm memiliki akurasi 100% Dan jarak 5-20 cm dan memiliki akurasi 100% [36].

Selanjutnya ada penelitian tentang memanfaatkan kamera CCTV yang ada di setiap lantai untuk mendeteksi keberadaan orang yang ingin ditemui. menggunakan CNN (Convolution Neural Network) untuk mengolah data, metode yang digunakan adalah *object detection*, menggunakan algoritma YOLO (*You Only Look Once*) versi 4 dan OpenCV untuk mendeteksi manusia yang terlihat dan mengenalinya. Data yang digunakan berupa data foto, untuk setiap orang diambil 30 buah foto menghasilkan akurasi 80% [37].

Penelitian selanjutnya menggunakan *Convolutional Neural Network* (CNN) proses pendeteksian citra wajah digunakan Algoritma YOLO. Pendeteksian wajah dilakukan terhadap *input* berupa video hasil perekaman data dari *webcam* yang memuat ekspresi wajah, data ekspresi wajah dilatih dengan menggunakan CNN menghasilkan akurasi 85% dan akurasi *convolutional neural network* 46% [38].

Kemudian masih menggunakan metode *You Only Look Once* (YOLO) sebagai dalam pembuatan aplikasi deteksi jumlah orang pada area indoor ntuk mendukung pelaksanaan PPKM dengan metode YOLO dalam proses deteksi jumlah orang dengan jumlah data yang diuji 15 orang mendapatkan hasil rata-rata akurasi 86,7% [39].

Penelitian lainnya yang menggunakan metode *You Only Look Once* (YOLO) sebagai dalam perancangan alat identifikasi wajah dengan algoritma *You Only Look Once* (YOLO) untuk presensi mahasiswa. Prediksi wajah pada manajemen absensi akan dikirimkan ke sebuah *website monitoring* presensi mahasiswa. Hasil pengujian didapatkan rata-rata akurasi 0,9793 dengan memperhatikan parameter berupa pencahayaan dan *real time* mengirimkan ke *website* [40].

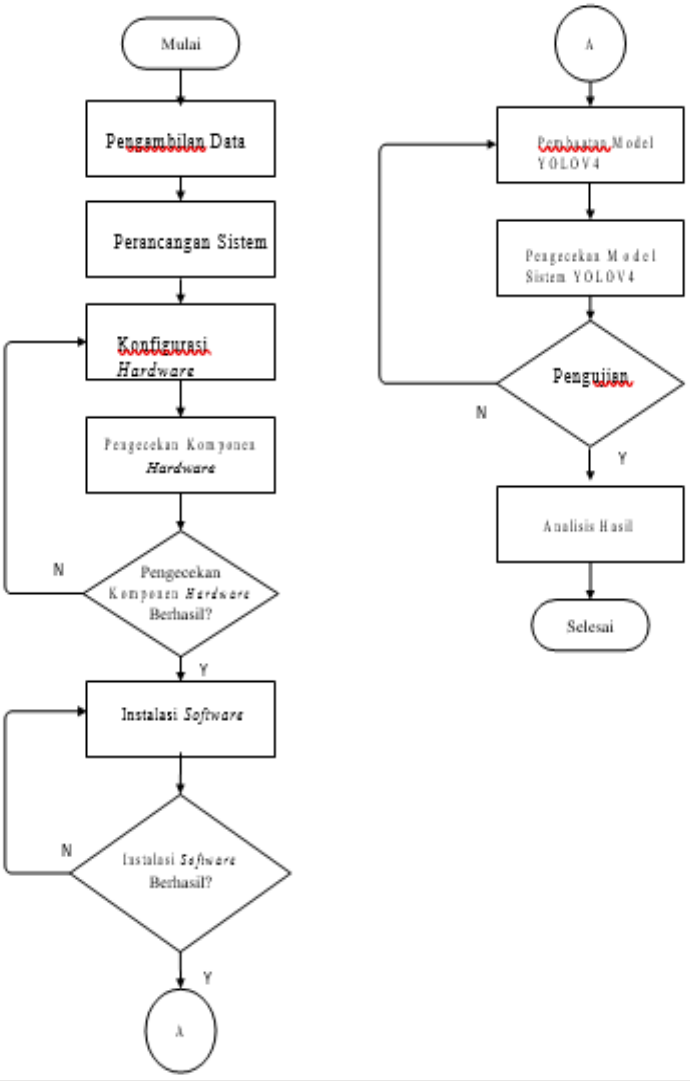
Penelitian selanjutnya yang menggunakan metode *You Only Look Once* (YOLO) adalah sistem pemantauan aktivitas keseharian lansia berbasis deteksi objek menggunakan algoritma YOLO. Data yang didapatkan akan dikirimkan ke *smartphone caregiver* dan keluarga via Telegram dalam bentuk pesan dan notifikasi. Dengan sistem tersebut, *user* dapat mengetahui aktivitas lansia dilokasi melalui aplikasi Telegram saat *user* di luar jangkauan lansia dalam bentuk pesan rutin berupa data aktivitas lansia. Hasil pengujian didapatkan rata-rata akurasi hasil Presisi 100%, *Recall* 100%, F1 *Score* 100%, *Average* IoU 87.26%, *Average Loss* 6.41%, mAP 100% serta akurasi yang dihasilkan mencapai 100% dengan parameter model yang digunakan adalah Rasio 90%:10%, *Batchsize* 64, *Learning rate* 0.008 dan *Max Batches* 4000 [8].

Penelitian yang lain menggunakan metode *You Only Look Once* (YOLO) adalah sistem absensi yang dapat dijalankan dengan fitur *Global Positioning System* (GPS) untuk secara otomatis mengecek lokasi pemilik wajah. Dari 20 data penilaian yang dilakukan sistem, sistem manajemen absensi dengan fitur pengenalan wajah dan GPS menggunakan YOLO pada *platform* Android menghasilkan akurasi sebesar 0.93435 dan terendah masih dalam range 93%, sedangkan nilai rata-rata akurasi adalah 93.26% [6].

BAB III
METEDOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini merancang sistem *real-time* untuk identifikasi wajah dengan menggunakan metode YOLO. Data yang digunakan dalam penelitian ini berupa gambar wajah manusia dengan sudut pandang yaitu dari sisi depan wajah. Gambar tersebut diambil dari hasil video rekaman menggunakan kamera ponsel dan webcam dengan bantuan laptop. Pengambilan data dan pengujian berlokasi di berbagai tempat pengambilan video. Adapun alur penelitian untuk identifikasi wajah seperti Gambar 3.1 berikut:



Gambar 3.1 Flowchat Penelitian

Berdasarkan *flowchart* tahapan penelitian pada Gambar 3.1, maka dapat diketahui bahwa tahapan-tahapan yang digunakan untuk menyelesaikan penelitian tersebut adalah sebagai berikut:

1. Melakukan pengambilan data dengan menggunakan kamera telefon seluler yang berupa video gambar wajah manusia. Data wajah yang diambil ialah khusus dari sisi depan saja.
2. Perancangan sistem, perancangan sistem ini dilakukan untuk merancang sistem yang ingin dibuat supaya sistemnya nanti dapat terbentuk sesuai dengan konsep penelitian. Perancangan sistem yang terbentuk merupakan gambaran dari sebuah mekanisme sistem yang ingin dibuat nantinya
3. Melakukan konfigurasi pada laptop seperti memasang perangkat lunak atau *software* dan *library* yang dibutuhkan agar dapat melakukan pendeteksian wajah manusia.
4. Memasukkan data yang sudah diambil melalui kamera belakang seluler ke model yang telah dibuat.
5. Melakukan pengujian terhadap program dan model YOLO yang telah dibuat lalu didalamnya mendapatkan hasil Mendapatkan hasil dari deteksi berupa bentuk *confusion matrix*. *Confusion matrix* berupa tabel matriks 3x3 sesuai jumlah jenis data yang di dalamnya berisi jumlah *true positive*, *true negative*, *false positive*, dan *false negative*.
6. Hasil yang diperoleh berupa pendeteksian wajah manusia, setelah itu dianalisis.
7. Setelah data pengujian didapat dan dianalisis selanjutnya melakukan kesimpulan dan menentukan bagus tidaknya akurasi sistem bekerja untuk proses sistem identifikasi wajah manusia, jika program masih belum sesuai dengan hasil yang diharapkan akan kembali ke *point* dua untuk melakukan penambahan data yang sudah ada.
8. Menulis laporan penelitian setelah semua sistem program dan pengujian data identifikasi wajah manusia berjalan dengan baik dan tepat.

3.2. Menentukan model wajah manusia

Pengambilan data wajah menggunakan kamera telefon seluler dengan bentuk formatnya video. Video yang telah didapat, kemudian diambil gambar yang hanya terdapat wajah sesuai dengan kondisi yang telah ditentukan. Dalam satu video, setiap orang diambil hanya dari sisi depannya saja sehingga pada saat *screenshoot* diambil gambar menghasilkan gambar dengan posisi depan saja. Gambar 3.2 berikut merupakan beberapa gambar data yang digunakan.

Gambar 3.2 Sampel data yang digunakan

Berdasarkan Gambar 3.2, Gambar tersebut telah dilakukan serangkaian proses pengolahan gambar seperti *labelling*, augmentasi, dan serangkaian proses lainnya agar bisa diproses dengan menggunakan bahasa pemrograman, hanya gambar dari posisi depan yang dijadikan data dan menentukan wajah manusia untuk melakukan identifikasi wajah. Pada penelitian ini melakukan identifikasi wajah pada 20 orang.

3.3. Alat dan bahan

Instrumen yang digunakan dalam perancangan dan penelitian ini meliputi perangkat keras atau *hardware* dan perangkat lunak atau *software* yang dapat

mendukung dan mempermudah kinerja dalam proses pengerjaan penelitian. Tabel 3.1 berikut merupakan perangkat yang digunakan.

Tabel 3.1 Instrumen Penelitian

No		Instrumen	Aplikasi
1	*Hardware*	Asus A455L, RAM 8GB, Processor Core i3 5030 iu, GPU NVIDIA Geforce 930 2GB.	Berfungsi untuk mengolah gambar, memproses gambar dan menjalankan program identifikasi wajah.
		Webcam	Sebagai input gambar ke laptop.
2	*Software*	Python IDLE	Membuat *listing* program lalu menyusun dan mengunggah program pada Laptop.

Berdasarkan pada Tabel 3.1, terdapat *hardware* dan *software* yang diperlukan dalam penelitian ini. Pada *hardware*, terdapat 2 komponen utama dalam menjalankan sistemnya. Sedangkan pada *software*, terdapat 1 aplikasi yang memiliki peran sangat penting.

3.4. Menentukan Media

Media bertujuan untuk pengambilan data latih untuk melakukan pengujiannya. Pada penelitian ini media yang digunakan adalah kamera telepon seluler. Penelitian ini menggunakan adalah kamera telepon seluler agar lebih mudah mendapatkan data wajah orang yang dipilih untuk membuat data latihnya dan menganalisis hasilnya.

3.5. Konfigurasi Laptop

Konfigurasi Laptop bertujuan supaya dapat melakukan *preprocessing data training* dan menjalankan program untuk melakukan identifikasi wajah. Berikut ini beberapa tahapan sebelum gambar dapat melalui proses *training.*

3.5.1. Mengambil Gambar Wajah Dari Hasil Video Rekaman.

Data latih telah diambil sebelumnya berupa video orang yang di dapat dari hasil merekam langsung dari telefon seluler. Setelah data latih didapat, kemudian diambil gambar wajahnya saja dan disimpan.

3.5.2. Mengumpulkan Data Latih

Data latih digunakan untuk membangun model YOLO dari wajah manusia. Data latih didapat dari pengambilan video wajah manusia secara pribadi. Data latih berupa video nantinya akan diambil gambar yang ada manusianya saja. Pengambilan data latih dilakukan pada siang atau pun malam hari dalam kondisi pencahayaan yang terang dan minim cahaya.

3.5.3. Menentukan model YOLO *frameworks*

YOLO *frameworks* memiliki banyak model, sehingga harus memilih model yang sesuai dengan *hardware* supaya mendapatkan hasil yang maksimal dan sesuai dengan yang diinginkan. Pada penelitian ini menggunakan model YOLOv4. Model tersebut dipilih karna memiliki akurasi yang paling baik di *hardware* ini dibandingkan dengan model YOLO yang lain.

3.5.4. *Preprocessing Training* Data

Pengolahan data latih bertujuan untuk membangun model YOLOv4. Data latih berupa gambar yang sudah diambil sebelumnya kemudian diproses sehingga menjadi model yang memiliki *extention*.*weights*. Adapun tahap-tahap mengolah data latih berikut ini.

3.5.5. Melakukan Normalisasi Gambar

Gambar yang telah didapat dari hasil video rekaman orang yang diambil langsung menggunakan kamera telepon seluler dengan posisi bagian depan wajah, sisi kiri wajah,dan sisi kanan wajah. Normalisasi gambar dilakukan untuk menyamakan rasio menjadi ukuran gambar. Ukuran rasio gambar akan berpengaruh pada proses *training*, semakin kecil ukuran filenya maka akan semakin cepat proses *training* dan akan mempersingkat waktu untuk *upload* data *training* ke google drive. Ukuran rasio gambar yang digunakan adalah 768x1024 *pixel*.

3.5.6. Melakukan *Scale Modifier*

*Scale modifie*r bertujuan untuk memperkecil atau memperbesar ukuran rasio gambar, berikut dibawah ini adalah ukuran Gambar yang belum melewati proses scale modifier dapat dilihat pada gambar 3.3.

768x1024

Gambar 3.3. Gambar sebelum Proses *Scale Modifier*

Berdasarkan Gambar 3.3 sebelum Gambar dilakukan proses scale modifier gambar tersebut mempunyai rasio gambar dengan ukuran normal 768x1024 *pixel*. dan dapat dilihat pada Gambar 3.4 yang sudah melewati tahap *scale modifier*.

768x1024

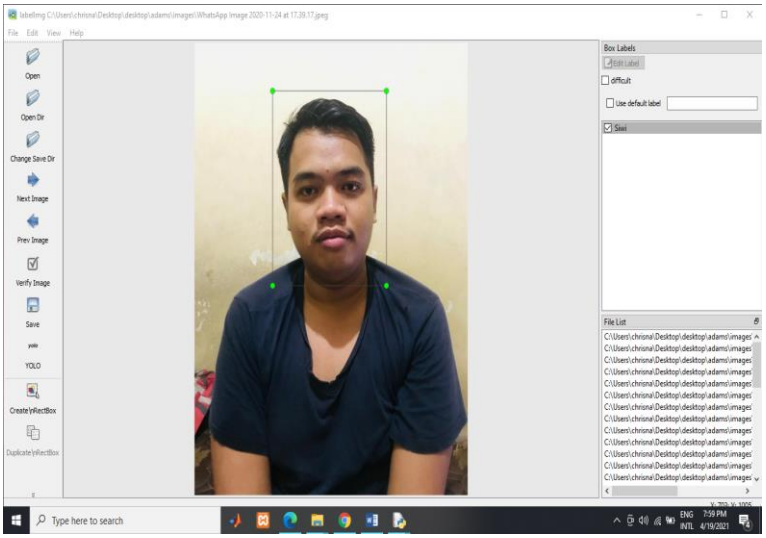
Gambar 3.4. Gambar sesudah Proses *Scale Modifier*

Berdasarkan Gambar 3.4 setelah dilakukan proses *scale modifier* dan normalisasi gambar tersebut berubah menjadi ukuran 768x1024 *pixel* dengan ukuran gambar yang lebih kecil dari sebelumnya. Tujuan dari *scale modifier* sendiri adalah supaya dapat mendeteksi wajah pada jarak yang jauh atau dekat dari *webcam*

dengan akurasi yang sama baiknya. kemudian dilakukan proses *scale modifier* sehingga akan menghasilkan ukuran gambar yang mempunyai rasio yang sama.

3.5.7. Melakukan Pelabelan Gambar/Citra

Proses selanjutnya adalah melakukan pelabelan gambar/citra atau *labeling*. *Labeling* gambar menggunakan aplikasi *LabelImg* kemudian simpan dengan format YOLO supaya hasilnya berformat .txt.. Proses pelabelan gambar menggunakan aplikasi *LabelImg* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Proses Pelabelan Gambar

Berdasarkan Gambar 3.5 dapat dilihat bahwa proses anotasi gambar adalah dengan membuat kotak pada wajah manusia, kemudian menamai kotak sesuai dengan identitas manusia tersebut dan setelah itu disimpan. Pada proses ini akan menghasilkan *file* dengan format "namafilewajah.txt" dan "classes.txt". Pada *file* "namafilewajah.txt" berisi koordinat posisi wajah pada gambar dan pada *file* "classes.txt" berisi identitas dari wajah manusia itu sendiri yang telah diberi nama atau identitas.

3.5.8. Konfigurasi Model YOLOv4

Melakukan konfigurasi *classes, batch, subdivisions, dan filters* pada model YOLOv4.cfg. Konfigurasi tersebut disesuaikan dengan banyaknya jumlah wajah yang ingin diidentifikasi. pada *line batch* diganti dengan *batch*=64,

subsdivisions=16, *max_batches* = (banyaknya wajah yang ingin diidentifikasikan)* (2000), *steps*=(80% dari *max_batches*), (90% dari *max_batches*), *filters* = (banyaknya wajah yang ingin diidentifikasikan + 5)*3.

3.5.9 . Konfigurasi *File* Program

Proses konfigurasi *file* Program `obj.names` dan `obj.data` merupakan bagian yang penting untuk melakukan *training*. Pada *file* `obj.names` merupakan *file* yang ada dalam proses *training* untuk menentukan *classes* berisi nama wajah yang ingin diidentifikasikan dan `obj.data` merupakan *file* yang ada dalam proses *training* untuk menentukan *classes* berisi jumlah nama wajah yang ingin diidentifikasikan.

3.5.10. *Training* Data dengan Google Colab

Training data dilakukan setelah langkah *preprocessing training* datanya selesai. Pada proses *training* data ini, prosesnya dilakukan di Google Colab karena terdapat super GPU (*Graphics Processing Unit*), sehingga bisa mempercepat prosesnya. Google Colab yang digunakan ialah Google Colab Pro yang terbatas dimana mempunyai keuntungan diantaranya dapat mengakses ke GPU (*Graphics Processing Unit*) dan TPU (*Tensor Processing Unit*) yang lebih cepat, memiliki lebih banyak memori RAM dan *disk*, dan mendapatkan waktu proses yang lebih lama sehingga lebih jarang memutuskan sambungan. Terdapat beberapa tahapan dalam melaksanakan proses *training* data dengan menggunakan Google Colab, tahapannya yaitu:

1. Masuk ke *website* https://colab.research.google.com/.
2. *Login* melalui akun Google, jika belum mempunyai akun Google, maka buat akun terlebih dahulu.
3. *Upload file* dengan format .ipynb ke Google Colab.
4. Setelah *upload file* berhasil, maka muncul beberapa *listing code* yang dieksekusi diantaranya yaitu *mount drive*, ekstrak *file* model, update sistem terbaru dan *install library* yang dibutuhkan. Langkah eksekusi tersebut wajib dilakukan sebelum *training* data dimulai.

5. *Mount drive* with colab. Hal ini dilakukan agar *file* nya tersimpan ke direktori *drive*.
6. *Upload file* data latih dan modelnya *file* data latih yang dimaksud ialah *file* hasil dari *preprocessing training* data.
7. *Update* sistem dan *install library* yang dibutuhkan. Hal ini wajib dilakukan untuk mendukung proses jalannya *training* data.
8. *Training* data dilakukan.
9. *Backup* berkala. Hal ini dilakukan karena saat *training* data, ada kemungkinan gagal dan jika itu terjadi, maka *file* nya menghilang. *Backup* juga dilakukan dengan antisipasi koneksi yang tiba-tiba bisa terputus.

Lalu setelah melalui proses *training data* akan menghasilkan model YOLOv4.weight. Proses *training* dilakukan di google colab, karena pada google colab terdapat super GPU yang akan mempercepat proses *training data*. *Output* yang dihasilkan setelah *training* adalah *file* yang berektensi .weight.

3.6. Menentukan Kinerja Alat

Menentukan kinerja alat ini dimaksudkan untuk mengetaui performa dari komponen utama yang digunakan. Terdapat 2 tujuan utama yang penting diantaranya yaitu pengaturan *Frame Per second* (FPS) atau *frame rate* dan pemakaian CPU pada sistem ini. Pengaturan FPS merupakan pengaturan yang dilakukan untuk mengetahui berapa nilai FPS yang didapat dengan menggunakan komponen utama laptop itu sendiri. FPS perlu diketahui karena bisa menjadi acuan bahwa komponen yang digunakan apakah sudah sesuai atau belum dan berjalan baik atau tidak. Cara untuk mengetahui FPS yang didapat pada laptop yang diuji, dibuat program Python dan dijalankan programnya, programnya dapat dilihat pada Lampiran A-.

Kemudian untuk mengetahui CPU *usage* pada komponen laptop itu sendiri yang menggunakan sistem operasi windows, gunakan perintah menekan tombol *Ctrl + Shift + Esc* untuk membuka *Task Manager*. Klik *tab Processes* untuk melihat secara detail melihat aplikasi apa saja yang sedang dijalankan khususnya ketika membuka atau sedang menjalankan programnya, dengan seperti itu kita dapat melihat penggunaan CPU dan penggunaan memorinya.

3.7 Tempat dan Waktu Penelitian

Penelitian dilakukan di Fakultas Teknik Universitas Sultan Ageng Tirtayasa. Waktu pengerjaan Skripsi ini berlangsung dari bulan Maret 2022.

BAB IV
PEMBAHASAN DAN HASIL

Hasil pembahasan dan pengujian yang dilakukan dijelaskan di bab ini. Tujuan dari bab ini adalah untuk mengetahui tingkat keberhasilan terhadap perancangan sistem yang telah dirancang sebelumnya. Pengujian yang dilakukan meliputi pengujian tingkat akurasi deteksi wajah dari sisi depan, intensitas cahaya dan pengujian tingkat akurasi identifikasi wajah.

4.1. Hasil Pengambilan Data Latih

Pengambilan data latih menggunakan kamera telepon seluler berbentuk video. Video yang telah didapat kemudian diambil gambar yang hanya terdapat wajahnya saja, gambar tersebutlah yang menjadi data latih. Tabel 4.1 berikut yang berisi jumlah gambar yang menjadi data latih.

Tabel 4.1 Jumlah Data Latih

Daftar Wajah	Jumlah gambar
Adam abdul malik	30
Aditya Fitrais Nugroho	30
Ananda Saksena Siwi	30
Bapak Sulis	30
Dian Aini	30
Dila Septiyani	30
Fariz Alfarizi	30
Galih Prasetya Ningtyas	30
Gefira Aina	30
Geovanny Valerian	30
Hanif Anggit Wicaksono	30
Ibu Susilawati	30
Icah Nuraisyah	30
Ika Ervina	30
Layin Hafidzah Nurrahmah	30
Muhamad Aulia Muhibubidin	30
Nugroho Anis Rahmanto	30
Nur Aisyah	30
Pandu Akbar Maulana	30
Rifaldi Tryawan	30
Total	600

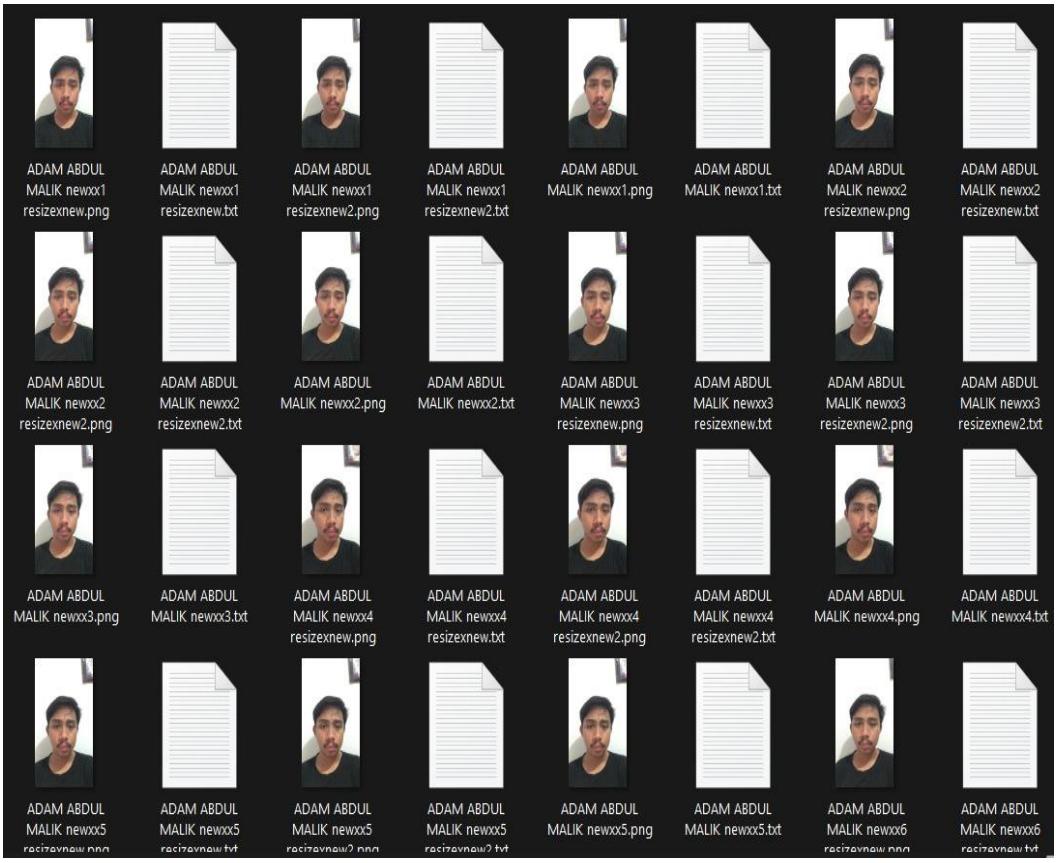
Berdasarkan data pada Tabel 4.1 dapat dilihat total data latih yang diambil dalam penelitian ini adalah sebanyak 600 gambar dengan total nama data manusia yang diuji sebanyak 20 orang. Masing-masing data manusia dibuat sama agar mendapatkan data yang seimbang dalam pengolahan data dan pengujian identifikasi wajah manusia.

4.2. Hasil Pelabelan Objek

Pelabelan objek adalah proses pembuatan label pada gambar dengan cara memberikan kotak pembatas atau yang sering disering ditemukan istilahnya adalah *bounding box* beserta nama kelas pada objek. Tahap pelabelan objek dilakukan ketika data yang berupa video deteksi objek berupa pengenalan wajah manusia atau *human face recognition*, diambil data wajahnya saja melalui *screenshot* diaplikasi pemutar video. Hasil transformasi video menjadi gambar dapat dilihat pada Gambar 4.1 berikut ini:

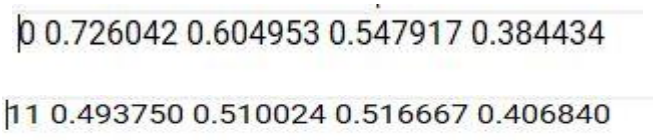
Gambar 4.1 Hasil Transformasi Video Menjadi Gambar

Berdasarkan Gambar 4.1 Objek yang dilabel dari sisi muka tampak depan sesuai dari hasil video rekaman. Proses pelabelan ini menggunakan YOLO frameworks. Hasil dari pelabelan tersebut adalah data yang terdapat informasi letak kotak pembatas dan labelnya dalam bentuk `.txt`. Pada `.txt` file terdapat baris *file* yang memiliki format <*object-class*> <*x_center*> <*y_center*> <*width*> <*height*>, dimana pada <*object-class*> merupakan bilangan bulat yang menyatakan kelas objek, <*x_center*> dan <*y_center*> adalah koordinat pusat persergi kotak pembatas, <width> dan <height> adalah nilai *float* relatif terhadap dimensi gambar. Berikut merupakan hasil gambar yang sudah diberikan label dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Dataset Telah diberikan Label

Berdasarkan Gambar 4.2 dapat dilihat hasil masing-masing gambar yang sudah diberikan label, mempunyai nomor teks pelabelan itu sendiri. Ini sangat berbeda ketika gambar hanya baru diambil data gambarnya saja. Hasil dari program `.txt` *file* dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 Hasil dari Program

Berdasarkan Gambar 4.3 dapat dilihat hasil dari program `.txt` *file* gambar yang sudah mendapatkan label. Di dalam `.txt` *file* berisi informasi lebih lanjut mengenai identitas gambar berupa ukuran dimensi gambar dan nilai koordinat dari gambar.

4.3. Hasil Training data

Ketika tahap pelabelan sudah dilaksanakan, cara berikutnya ialah proses *training*. Langkah ini dibuat agar komputer dapat terlatih dengan citra yang diolah dengan data yang ada kemudian mendapatkan hasil struktur dan ciri khas masing-masing *classes* untuk material pengolahan komputer untuk mendapatkan hasil suatu tujuan atau probabilitas. Dalam tahap berikut memakai pre-*trained weights* YOLOv4 dengan memakai teknik *transfer learning*. *Transfer learning* ialah suatu cara yang dipakai suatu rangka yang telah melewati tahapan proses *training before* atau *pre-trained model* yang bisa diimplementasikan untuk pengenalan data baru dengan tidak melakukan *training* data dari awal. Konfigurasi *transfer learning* pada Darknet memakai data *file, cfg file*, dan *pre-trained weigths*. File data terdapat letak citra yang implementasikan untuk data latih dan percobaan. File CFG teerdapat kerangka struktur yang diimplementasikan untuk *training*, dan *pre-trained weights* terdapat model *weight* yang telah melewati proses *training* pada jaringan YOLO.

Tabel 4.2 Konfigurasi pada Darknet

Jenis Konfigurasi	**Keterangan**
Load Model	Darknet
Load Weight	YOLOv4
OPENCN	1
GPU	1
CUDNN	1

Berdasarkan pada Tabel 4.2 Proses *training* memakai Darknet-53 sebagai *load model* dan YOLOv4 sebagai *load weight* dengan pengaturan pada Tabel 4.2. Nilai *batch* berpengaruh pada total citra yang dikerjakan sebelum *network weight* melakukan perubahan. *Subdivision* berguna menjalankan separuh kecil ukuran *batch* bersamaan dengan GPU. *Max_batch* merupakan batas perulangan proses kerja *training* yang diperoleh melalui perbendingan. Suatu perulangan telah memperoleh angka 10000, lalu proses *training* otomatis berhenti. Nilai *max_batches* didapatkan dari Persamaan 4.1.

$$max\_batches = jumlahclass.2000 \quad (4.1)$$

Nilai step didapatkan pada Persamaan 4.2 berikut ini.

$$steps = (80\%max\_batches), (90\%max\_batches) \quad (4.2)$$

Height dan *Weight* merupakan dimensi gambar masukan yang dilatih. *Classes* merupakan jumlah kelas yang dideteksi. Nilai *filter* didapatkan pada Persamaan 4.3 dibawah ini.

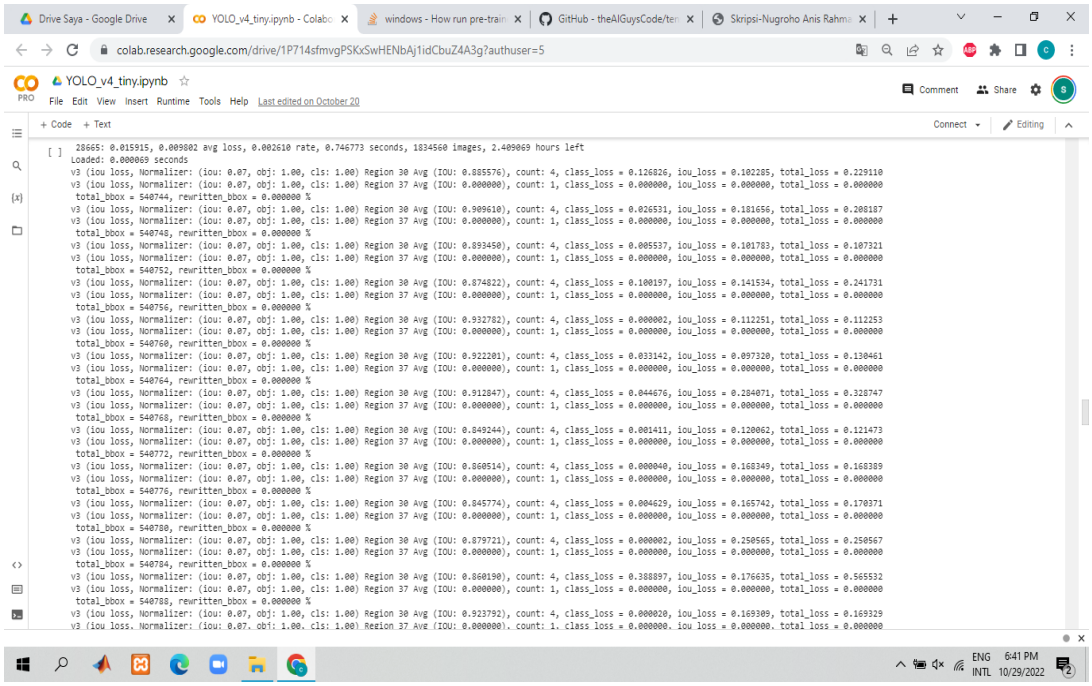
$$filter = (jumlahclass + 2)x\ 3 \quad (4.3)$$

Tabel 4. 3 Konfigurasi pada *weights* YOLOv4

Jenis Konfigurasi	**Keterangan**
Batch	64
Subdivisions	64
Width	608
Height	608
max_batches	40000
Steps	32000, 36000
Classes	20
Filters	75

Berdasarkan setelah proses pada Tabel 4.2 dan Tabel 4.3 semuanya sudah selesai mengkonfigurasi dan semua *file* yang diperlukan untuk *training* data tercukupi, langkah selanjutnya ialah melakukan proses *training* data yang sudah dibuat menggunakan google colab. Setelah data *training* selesai dibuat, tahapan berikutnya adalah pengujian mendeteksi objek dari sisi depan yang sudah

ditentukan. Setelah objek data benar maka langkah berikutnya melihat hasil di *computer vision*. Jika data yang ada sudah tedeteksi cocok dengan wajah orang yang sudah melewati proses *training* dan diuji dengan data yang baru akurat, maka proses bisa dikatakan berhasil. Hasil *training* data latih dapat dilihat pada Gambar 4.4 berikut.

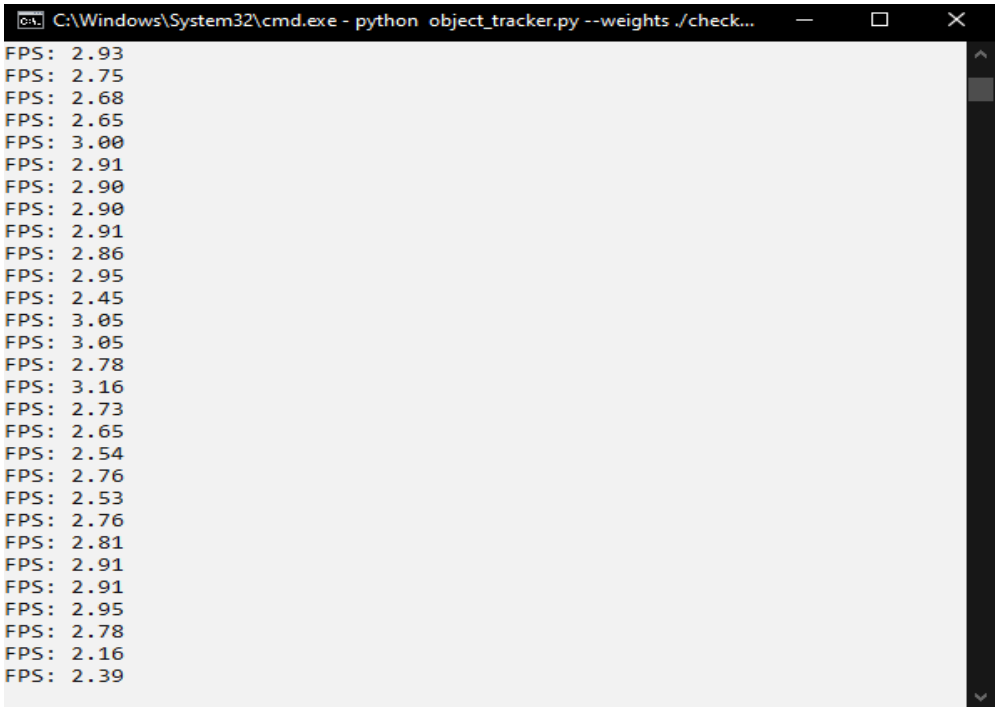


Gambar 4.4 Proses *Training* Data Latih

Berdasarkan Gambar 4.4 dapat dilihat diatas bisa disimpulan bahwa proses *training* pada data yang di latih menggunakan google colab dengan dukungan nvidia dari laptop mendapatkan total *average loss* sekitar 0,0026. Semakin kecil *average loss* yang didapat dari proses *training* data latih maka data yang akan diuji pun akan semakin baik dan lebih akurat sistem dalam mengenali proses identifikasi wajahnya.

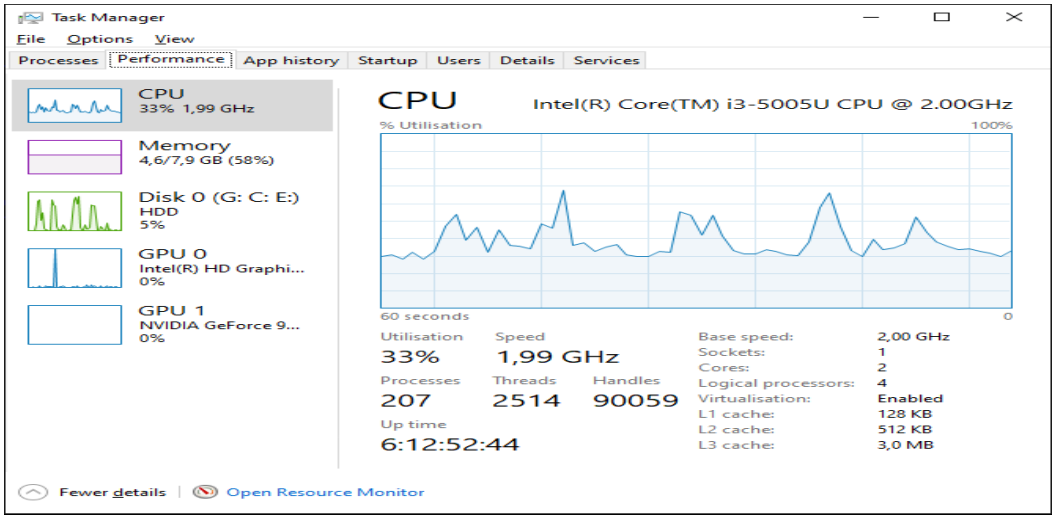
4.4 Hasil Kinerja Alat

Kinerja alat disini menunjukkan kinerja yang dihasilkan oleh komponen utama laptop dimana terdapat *Frame per Second (*FPS) yang didapat saat sistemnya dijalankan dan CPU *usage* pada saat tidak membuka apa-apa, membuka program, dan saat programnya dijalankan. Berikut merupakan FPS yang didapat saat sistemnya dijalankan.



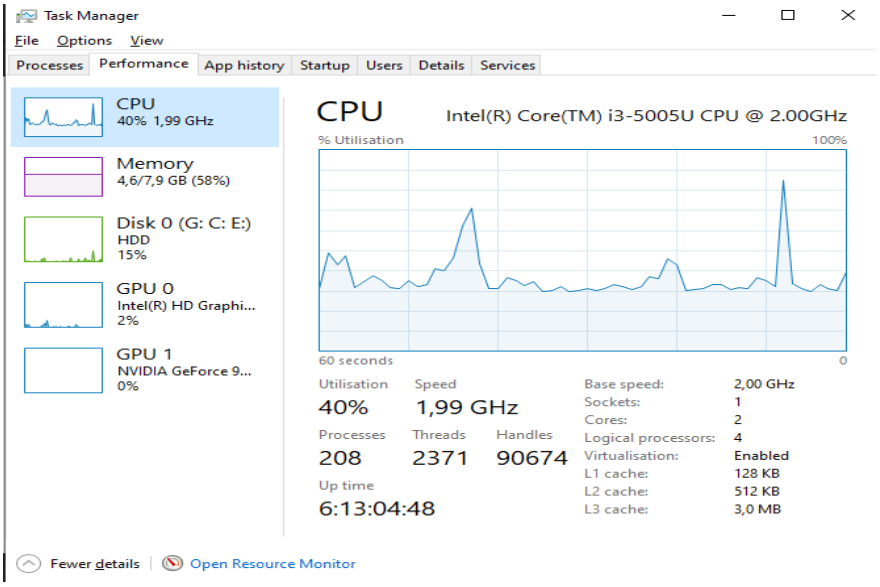
Gambar 4.5 FPS saat Program dijalankan

Berdasarkan Gambar 4.5, menunjukkan FPS yang ditangkap oleh sistem rata-rata sebasar 2,1 hingga 3,16 FPS, dimana FPS sebesar itu didapatkan saat program atau sistem dijalankan. Jika dibandingkan dengan menggunakan YOLO versi v3 ataupun YOLOv4 *Tiny* , FPS yang didapat lebih besar, tetapi hasil akurasinya tidak lebih baik jika dibandingkan dengan versi YOLOv4 biasa. Lalu untuk pemakaian CPU, berikut merupakan CPU *usage* saat tidak membuka aplikasi atau program apapun pada Gambar 4.6.



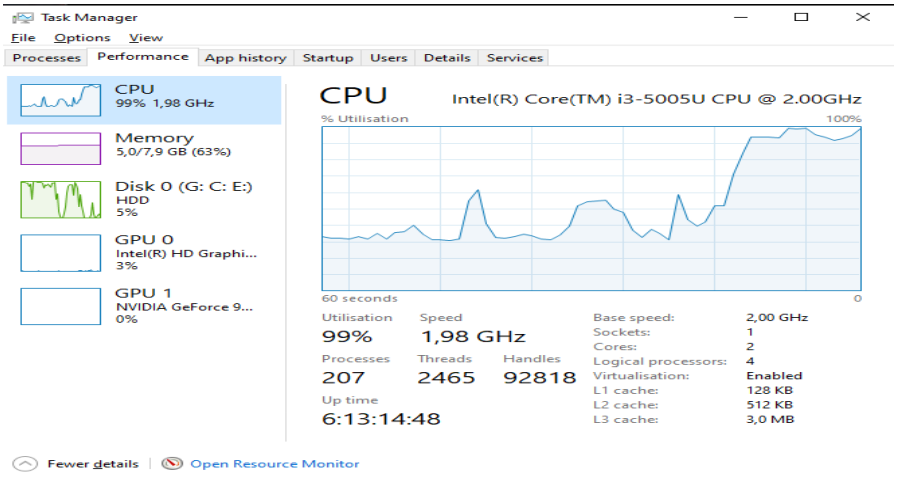
Gambar 4.6 CPU saat tidak membuka Program

Berdasarkan Gambar 4.6, dapat diketahui bahwa CPU yang digunakan memiliki 2 *core*. Saat tidak membuka aplikasi apapun, CPU *usage* yang terpakai sebesar 33% dan memori yang terpakai sebesar 4,6 GB dari 7,9 GB. CPU *usage* saat membuka program dapat dilihat seperti pada Gambar 4.7 berikut.



Gambar 4.7 CPU saat membuka program

Berdasarkan Gambar 4.7, saat membuka programnya CPU *usage* yang terpakai naik menjadi 40%, ini proses yang wajar ketika program phyton dijalankan , sedangkan memori yang terpakai terlihat sama saja 4,6 GB dari 7,9 GB. Hasil dari CPU *usage* saat programnya dijalankan dapat dilihat seperti pada Gambar 4.8 berikut.



Gambar 4.8 CPU saat Program dijalankan

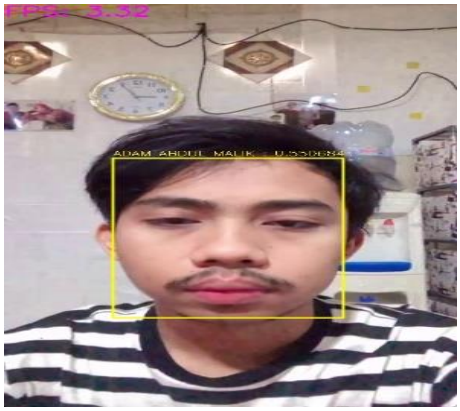
Berdasarkan Gambar 4.8 saat programnya dijalankan, CPU *usage* yang terpakai mencapai 99% dan memori yang terpakai naik mencapai 5,0 GB dari 7,9 GB. Dapat disimpulkan bahwa penggunaan CPU pada komponen laptop semakin meningkat seiring banyak dijalankannya program. Penggunaan CPU tertinggi mencapai 100%, sedangkan memori yang terpakai tetap yaitu sebesar 5,0 GB saat program sedang dijalankan.

4.5. Pengujian Deteksi Identifikasi Wajah

Pengujian deteksi identifikasi wajah disini meliputi identifikasi wajah bagian depan, identifikasi berdasarkan intensitas cahaya dan di waktu minim cahaya, yaitu bertujuan untuk mengetahui apakah sistem sudah dapat mengenali wajah dan menghitung tingkat akurasi dari sistem YOLOv4 *frameworks* yang telah dibuat. Banyak faktor yang dibutuhkan dalam keberhasilan pengujian ini, yaitu dari jumlah *dataset* yang telah dimiliki, jumlah *step* yang didapat, pencahayaan, serta resolusi yang dihasilkan dari kamera. Pengujiannya mendapatkan hasil seperti dijelaskan pada poin-poin berikut.

4.5.1. Hasil Pengujian Sisi Depan

Pengujian wajah dalam proses kerjanya ada aspek yang diterapkan untuk langkah *input* data untuk menemukan nilai akurasi deteksi wajah manusia. Arah kamera ialah hal yang mengharuskan untuk melakukan pengujian skripsi ini, berikut dapat dilihat pada Gambar 4.9 beberapa contoh hasil dari pengujian data dari sisi depan.

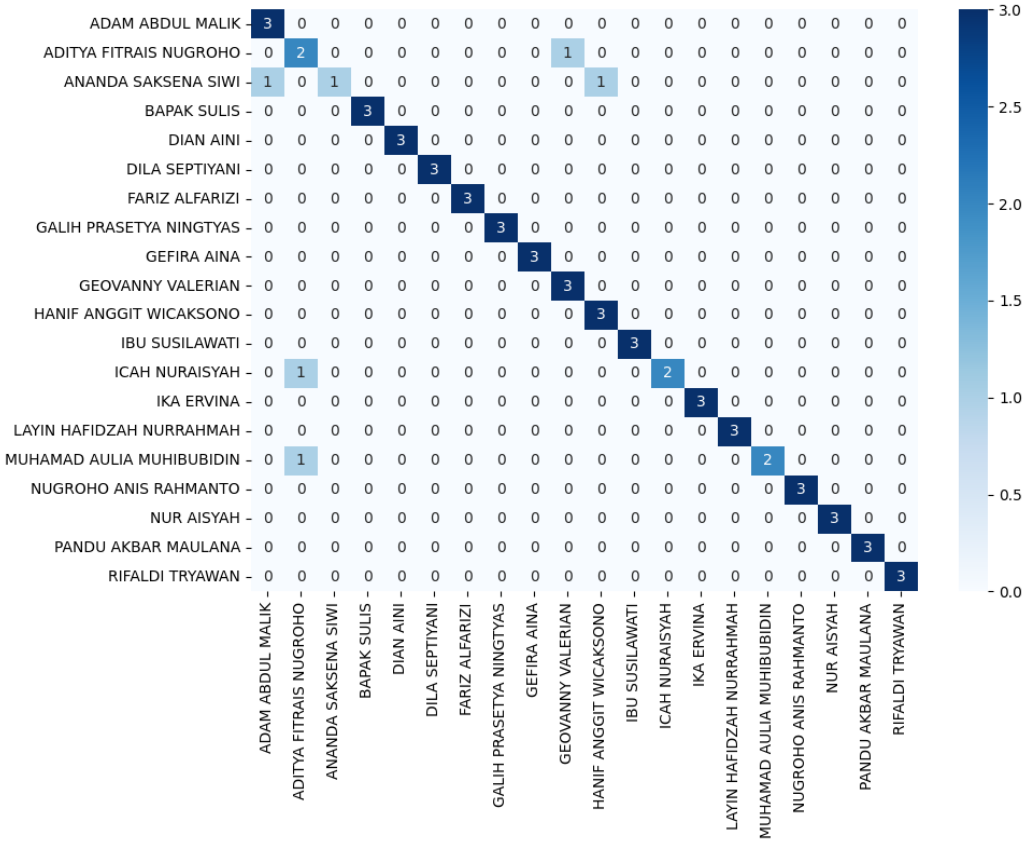


Gambar 4. 9 Hasil Pengujian pada Sisi Depan

Berdasarkan Gambar 4.9 dapat dilihat beberapa contoh proses pengujian hasil tingkat akurasi identifikasi wajah ada hal yang harus diperhatikan Langkah kerja untuk mengambil hasil data, dengan dari awal mengambil data untuk orang yang ingin diuji yang telah melawati tahap *training* untuk sekarang dicoba dalam proses pengujian. Pengujian ini dilakukan masing-masing 3 kali pengujian untuk mendapatkan data yang seimbang.

Objek yang diuji yaitu sebanyak 20 wajah manusia, dari pengujian yang diambil berdasarkan gambar 4.9 terlihat suatu hasil identikasi atau suatu pengenalan dalam dari *bounding box* gambar pada sisi depan kamera memperlihatkan hasil yang tepat dan akurat sistem berhasil mengenali nama wajah pertama yang diuji dalam pengujian identifikasi wajah yang diinginkan, untuk hasil semua pengujian identifikasi wajah dapat dilihat dilampiran B-1.

Dari hasil pengujian yang didapat dari sisi depan berdasarkan Gambar 4.9 menunjukkan teridentifikasinya data wajah yang diuji dalam bounding box, karena posisi wajah tepat menghadap kedepan. Dari tiap arah pengujian pengambilan gambar memiliki nilai data yang beragam karena ada beberapa faktor. Berikut adalah hasil akhir rata-rata persentase pengujian pengenalan sisi depan wajah manusia dapat ditampilkan pada Tabel 4.4 berikut ini.



Tabel 4.4 *Confusion Matrix* pada seluruh pengujian posisi depan

Berdasarkan Tabel 4.4 bisa diatikan diatas merupakan tabel *confusion matrix* pada seluruh pengujian identifikasi wajah pada posisi depan , dari tabel tersebut ada 20 objek wajah manusia yang diuji masing-masing 3 kali dengan total 60 kali proses uji. Dari data diatas bisa didapatkan nilai *true positive*, *false positive*, *false negative*, dan *true negative* dengan tabel dibawah ini. Hasil nilai pengujian bisa dilihat pada Tabel 4.5 berikut.

Tabel 4.5 Hasil seluruh pengujian identifikasi wajah pada posisi depan

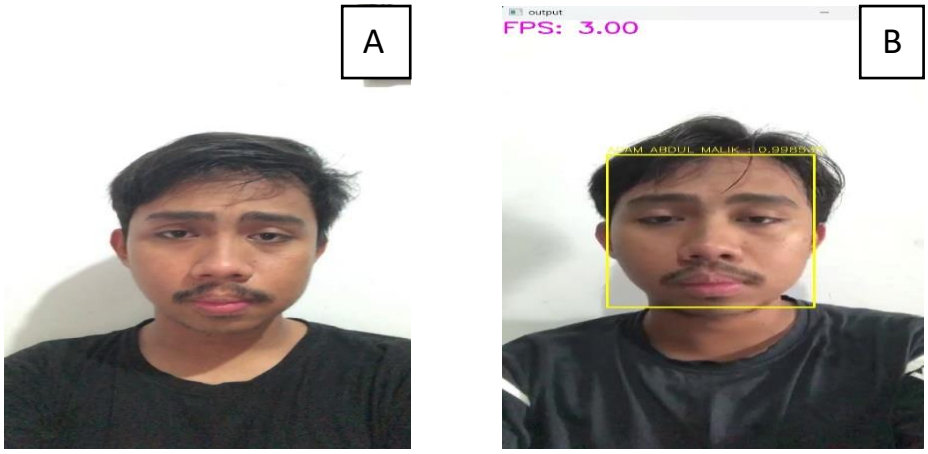
Data Wajah	Jumlah	TP	FP	FN	TN
Adam Abdul Malik	3	3	0	1	52
Aditya Fitrais Nugroho	3	2	1	2	53
Ananda Saksena Siwi	3	1	2	0	54
Bapak Sulis	3	3	0	0	52
Dian Aini	3	3	0	0	52
Dila Septiyani	3	3	0	0	52
Fariz Alfarizi	3	3	0	0	52

Galih Prasetya Ningtyas	3	3	0	0	52
Gefira Aina	3	3	0	0	52
Geovanny Valerian	3	3	0	1	52
Hanif Anggit Wicaksono	3	3	0	1	52
Ibu Susilawati	3	3	0	0	52
Icah Nuraisyah	3	2	1	0	53
Ika Ervina	3	3	0	0	52
Layin Hafidzah Nurrahmah	3	3	0	0	52
Muhamad Aulia Muhibubidin	3	2	1	0	53
Nugroho Anis Rahmanto	3	3	0	0	52
Nur Aisyah	3	3	0	0	52
Pandu Akbar Maulana	3	3	0	0	52
Rifaldi Tryawan	3	3	0	0	52
Total	60	55	5	5	1045

Dengan mengimplementasikan Persamaan 2.1, 2.2, 2.3, 2.4, dan data dari Tabel 4.5 diatas didapatkan nilai *recall* sebesar 0,91, nilai *precision* sebesar 0,91, nilai F-*Score* sebesar 0,90 dan akurasi sebesar 94,193%. Dengan total seluruh pengujian sebanyak 60 kali, nilai *true positive* sebesar 55, *false positive* dan *false negative* sebesar 5, dan total dari *true negative* sebesar 1045. Nilai *true negative* masing-masing pengujian didapat dari total seluruh pengujian mencari nilai *true positive* yang diuji dikurangi nilai total *true positive* seluruh pengujian 20 identifikasi wajah manusia dan hasilnya akan mendapatkan total *true negative* dari pengujian tersebut.

4.5.2 Hasil Pengujian Kedua berdasarkan Intensitas Cahaya yang sama

Pengujian kedua ini dilakukan dengan menggunakan intensitas cahaya yang sama bisa didapatkan dari berapa nilai akurasi *system* dalam proses identifikasi wajah manusia. Hasil pengujian kedua dapat dilihat pada gambar 4.10.



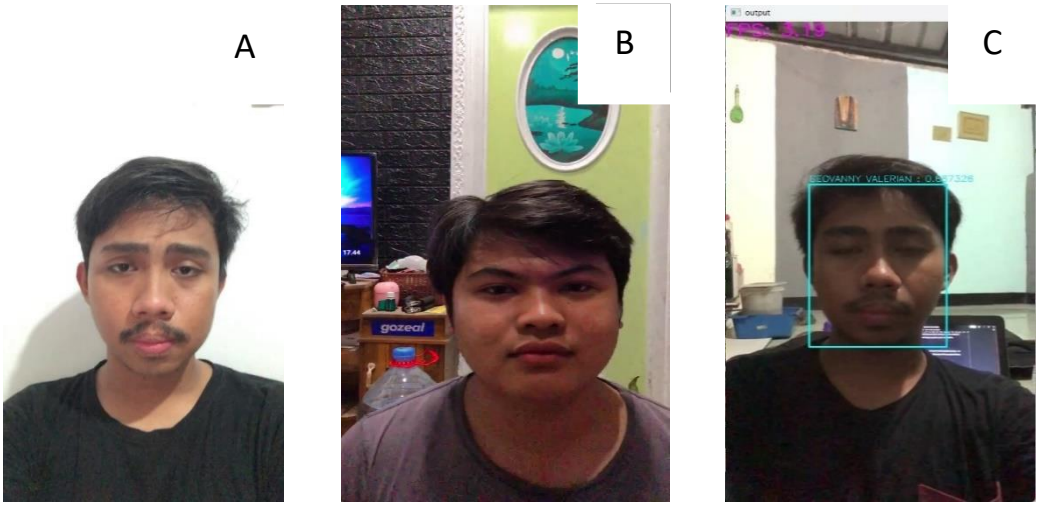
Gambar 4.10 Hasil Perbandingan Pengujian Kedua

A) Gambar dari Data Latih B) Gambar Hasil Pengujian

Berdasarkan pada Gambar 4.10 Hasil Perbandingan Pengujian kedua dapat dilihat Gambar A pada data latih Adam abdul malik memiliki intensitas cahaya sekitar 42 lux. Pada Gambar B data Gambar yang diuji memiliki intensitas cahaya sekitar 40 lux. Dari hasil pengujian dapat dilihat bahwa sistem berhasil mendeteksi wajah dengan baik. Ini dikarenakan sistem dapat mengenali data latih yang sudah melewati proses *training* data dengan baik, ketika dalam pengujian mengidentifikasi wajah sistem dapat mengenali objek wajah dengan benar karena data latih dan data uji tidak terlalu berbeda jauh besaran intensitas cahayanya.

4.5.3 Hasil Pengujian Ketiga dengan Intensitas Cahaya yang berbeda

Pengujian ketiga ini dilaksanakan dengan menerapkan besaran nilai cahaya yang beda dari data latih yang ada. Pengujian ini dilaksanakan guna mencari nilai akurasi *system* dalam proses identifikasi wajah manusia. Dalam percobaan kali ini menggunakan dua data yang berbeda untuk pengujiannya. Hasil pengujian ketiga dengan intensitas cahaya yang berbeda dapat dilihat pada gambar 4.11 berikut.

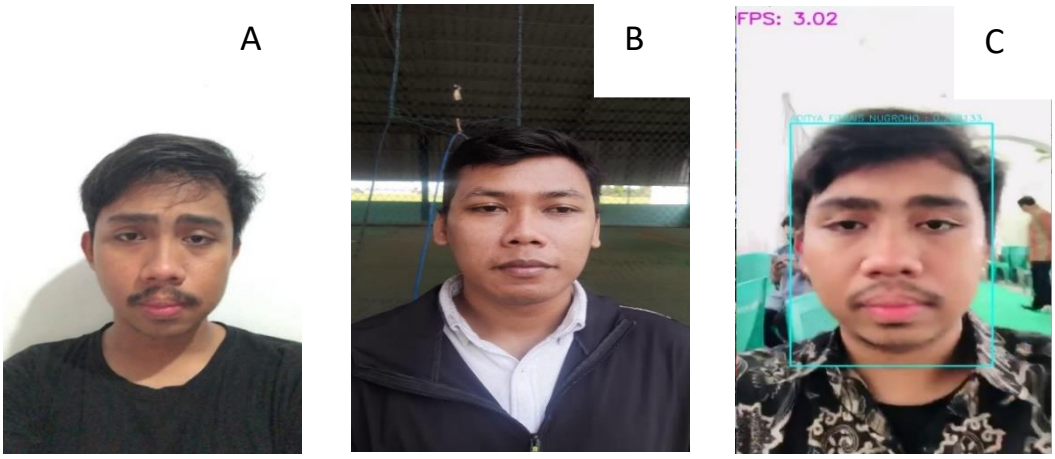


Gambar 4.11 Hasil Perbandingan Pengujian Ketiga
A) Gambar Data Latih Adam, B) Gambar Data Latih Geovanny, C) Gambar Hasil Pengujian Intensitas Cahaya Rendah

Berdasarkan Gambar 4.11 hasil perbandingan pengujian ketiga dapat dijelaskan pada Gambar A data latih Adam dengan intensitas cahaya sekitar 42 lux. Pada Gambar B merupakan Gambar data latih Geovanny dengan intensitas cahaya yang rendah sekitar 23 lux. Dapat disimpulkan melalui Gambar C yaitu citra hasil uji bahwa intensitas cahaya yang rendah tidak dapat mendeteksi wajah dengan baik sehingga seharusnya nama deteksi yang keluar Adam abdul malik menjadi Geovanny valerian. Ini terjadi karena data latih Geovanny mempunyai banyak data latih dalam keaadaan besaran cahaya yang minim, sehingga *system* gagal mengenali dengan baik dan akurat.

4.5.4 Hasil Pengujian Keempat berdasarkan Intensitas Cahaya Tinggi

Pengujian keempat ini dijalankan melalui system menggunakan besatan nilai cahaya yang lebih tinggi guna mendapatkan nilai akurasi *system* mengenali dalam proses identifikasi wajah manusia. Hasil pengujian keempat dapat dilihat pada gambar 4.12 berikut.

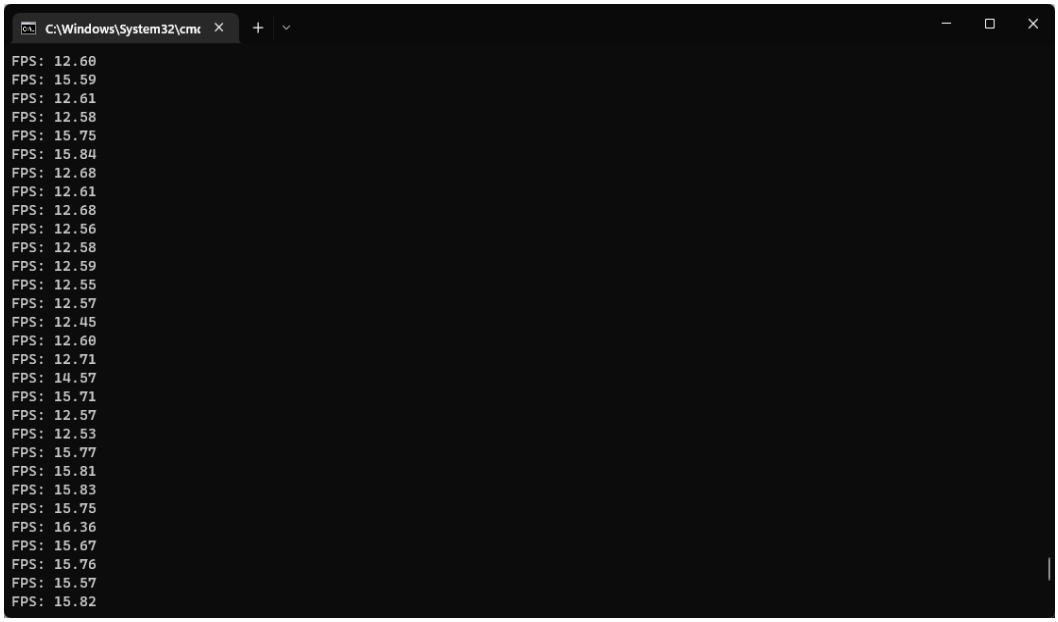


Gambar 4.12 Hasil Perbandingan Pengujian Keempat
A) Gambar dari Data Latih Adam B) Gambar Data Latih Adit
C) Gambar Hasil Pengujian Intensitas Cahaya tinggi

Berdasarkan Gambar 4.12 hasil perbandingan pengujian keempat dapat dilihat Gambar A data latih Adam mempunyai intensitas cahaya sekitar 42 lux. Pada Gambar B merupakan Gambar data latih Adit dengan intensitas cahaya sebesar 960 lux, dan pada Gambar C merupakan Gambar hasil pengujian dengan intensitas Cahaya yang sangat tinggi sebesar 4500 lux. Dapat dilihat bahwa intensitas cahaya yang terlalu tinggi pun kurang baik dalam mendeteksi khususnya wajah secara akurat, dikarenakan sistem lebih bagus mendeteksi data uji yang memiliki intensitas cahaya yang cukup dan mendekati dengan data latihnya yang telah melalui proses *training* data terlebih dahulu.

4.6 Hasil Perbandingan *Running* Program

Hasil *running* dari program yang dipakai di perangkat berbeda bertujuan untuk membandingkan hasil FPS yang di dapat di perangkat lain. Hasil dari perbandingannya dapat dilihat pada gambar 4.13 sebagai berikut.



Gambar 4.13 Hasil Running Program di Perangkat Lain

Berdasarkan pada gambar 4.13 dari hasil percobaan yang telah dilakukan rata-rata FPS yang didapat saat pengujian adalah 12 sampai dengan 15,8 FPS. Perangkat yang dipakai untuk membandingkan hasil *running* program ini spesifikasinya adalah Intel Core i5-10300H Processor (2.50 GHz, up to 4.50 GHz with Turbo boost, 4 cores, 8 thread) VGA NVIDIA GTX1650 with 4GB DDR6, VRAM memory 8GB DDR4 3200mhz. Perbedaan hasil kecepatan FPS dikarenakan perangkat yang di pakai memiliki spesifikasi yang terbatas.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan mengenai identifikasi wajah manusia menggunakan YOLO *frameworks* dengan metode *scale modifier* sebagai *preprocessing* secara *real time* dapat ditarik kesimpulan sebagai berikut:

1. Mampu membangun sistem identifikasi 20 jenis wajah manusia melalui yolo v4 *frameworks*.
2. Tingkat akurasi untuk mengidentifikasi 20 wajah manusia dari masing-masing tiga kali pengujian disetiap sisi yang telah dilakukan mendapatkan hasil akurasi rata-rata 94,19%, nilai F-*Score* sebesar 0,90%, nilai *precision* sebesar 0,91, dan hasil nilai *recall* sebesar 0,91.
3. Proses identifikasi wajah manusia sangat berpengaruh terhadap intensitas cahaya yang didapat.
4. Tujuan deteksi wajah ini lebih bisa digunakan dan dikembangkan kedepannya untuk absensi wajah, keamanan rumah, dan lain lain. Memilih YOLOv4 untuk identifikasi wajah manusia di perangkat terbatas masih bisa menggunakan *frameworks* dari YOLO yang di konversi ke versi yang lebih rendah dikarenakan perangkat dengan spesifikasi yang terbatas.

5.2 Saran

Adapun saran untuk pengembangan penelitian selanjutnya antara lain:

1. Pada penelitian ini menggunakan data latih sebanyak 600 gambar, untuk menghasilkan keakuratan yang lebih baik lagi dibutukan data latih yang lebih banyak lagi dan memiliki jumlah data yang sama disetiap jenis wajah yang di analisis dan seimbang dari tingkat intensitas cahayanya.
2. Untuk memaksimalkan kerja perangkat laptop dengan spesifikasi yang terbatas maka diperlukan *overclock* ke 2,4 GHz dan memberikan pendingin supaya tidak terjadi overheat. Efek dari overclock adalah bertambahnya FPS yang dijalankan. Dan untuk kedepannya menggunakan perangkat keras dengan spesifikasi yang lebih tinggi seperti GPU NVIDIA versi terbaru agar

dapat menggunakan versi YOLO terbaru sehingga proses pendeteksian bisa menjadi lebih baik dan mendapatkan FPS yang stabil.

3. Pada penelitian ini dapat dilakukan identifikasi wajah manusia pada malam hari namun masih belum optimal, karena perangkat laptop terbatas dan pengetahuan yang terbatas pada penelitian ini. Untuk kedepannya berharap agar penelitian selanjutnya dapat melakukan identifikasi wajah manusia dengan lebih baik lagi dan yang lebih akurat dalam pengidentifikasian objek wajah pada malam hari.

DAFTAR PUSTAKA

[1] Marpaung, F., F. Aulia, R. C. Nabila, "Computer Vision dan Pengolahan Citra Digital," Surabaya: *Pustaka Aksara,* 2022.

[2] Lauw, K. O., L. W. Santoso, R. Intan, "Identifikasi Jenis Anjing Berdasarkan Gambar Menggunakan Convolutional Neural Network Berbasis Android," *Jurnal Infra*, vol. 8, no. 2, pp. 37–43, 2020.

[3] Yuliati, A., C. Machbub, P. H. Rusmin, "Kajian Singkat Tentang: Pengendalian Dan Penjejakan Objek Berbasis Visual," *Al-Jazari Journal of Mechanical Engineering*, vol. 3, no. 1, pp. 1–9, 2018.

[4] Cheng, R., "A survey: Comparison between Convolutional Neural Network and YOLO in Image Identification," *Journal of Physics: Conference Series*, vol. 1453, no. 1, 2020.

[5] Patil, A., M. Rane, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," *Smart Innovation System Technology*, vol. 195, pp. 21–30, 2021.

[6] Hartiwi, Y., E. Rasywir, Y. Pratama, P. A. Jusia, "Sistem Manajemen Absensi dengan Fitur Pengenalan Wajah dan GPS Menggunakan YOLO pada Platform Android," *Jurnal Media Informatika Budidarma*, vol. 4, pp. 1235–1242, 2020.

[7] Shinde, S., A. Kothari, V. Gupta, "YOLO based Human Action Recognition and Localization," *Procedia Compuer Science*, vol. 133, no. 2018, pp. 831–838, 2018.

[8] Addin, M., C. Setianingsih, T. W. Purboyo, "Sistem Pemantauan Aktivitas Keseharian Lansia Berbasis Deteksi Objek Menggunakan Algoritma YOLO Monitoring System Of Elderly Daily Activities Algorithm," *eProceeding of Engineering*, vol. 10, no. 1, pp. 836–843, 2023.

[9] Mileanasari, F., F. Anisa, M. S. Abdillah, N. Setyawan, "Monitoring of Physical Distance for Covid-19," *Prosiding SENTRA (Seminar Teknologi dan Rekayasa)*, pp. 33–38, 2021.

[10] Baihaqi, K. A., Y. Cahyana, "Application of Convolution Neural Network Algorithm for Rice Type Detection Using Yolo v3," *Jurnal SYSTEMATICS*

vol. 3, no. 2, pp. 272–280, 2021.

[11] Cao, C. Y., J. C. Zheng, Y. Q. Huang, J. Liu, C. F. Yang, “Investigation of a Promoted You Only Look Once Algorithm and Its Application in Traffic Flow Monitoring,” *Applied Science*, vol. 9, no. 17, pp. 1–14, 2019.

[12] Putra, M. H., Z. M. Yussof, K. C. Lim, S. I. Salim, “Convolutional Neural Network for Person and Car Detection Using YOLO Framework,” *Journal of Telecommunication Electronic and Computer Engineering*, vol. 10, no. 1–7, pp. 67–71, 2018.

[13] Malahella, A. H., I. Arwani, Tibyani, “Pemanfaatan Framework React Native dalam Pengembangan Aplikasi Pemesanan Minuman Kopi pada Kedai Bycoffee,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 9, pp. 3178–3184, 2020.

[14] Arslan, R. C., M. P. Walther, C. S. Tata, “formr: A Study Framework Allowing for Automated Feedback Generation and Complex Longitudinal Experience-Sampling Studies Using R,” *Behaviour Research Methods*, vol. 52, no. 1, pp. 376–387, 2020.

[15] Aini, Q., N. Lutfiani, H. Kusumah, and M. S. Zahran, “Deteksi dan Pengenalan Objek Dengan Model Machine Learning : Model Yolo,” *Jurnal SENSI*, vol. 6, no. 2, pp. 192–199, 2021.

[16] Jatmiko, W., P. Mursanto, G. Jati, "Real Time Operating System (Rtos) Teori dan Aplikasi," Depok: *Fakultas Ilmu Komputer Universitas Indonesia.* 2015.

[17] Bimodwi, “Sejarah dan Manfaat Bahasa Pemrograman Python,” 2022. https://idmetafora.com/news/read/691/Sejarah-Dan-Manfaat-Bahasa-Pemrograman-Python.html (diakses 18 Juli 2023).

[18] Kurniawan, "Input dan Output Pada Bahasa Pemograman Python," *Jurnal Dasar Pemograman Python STMIK*, pp. 1–7, 2018.

[19] Nurohman, P., “Kenapa Kamu Harus Memilih Mempelajari Bahasa Pemrograman Python,” 2016. https://codepolitan.com/blog/kenapa-kamu-harus-memilih-bahasa-pemograman-python-57cdd334db9c2-18512 (diakses 18 Juli 2023).

[20] Andre, “Tutorial Belajar Python Part 5: Cara Menjalankan Python dari IDLE,” 2018. https://www.duniailkom.com/tutorial-belajar-python-cara-

menjalankan-python-dari-idle/ (diakses Jul. 18, 2023).

[21] Munir, R., "Pengantar Pengolahan Citra," *Pengolah. Citra Digit.*, no. Bagian 1, pp. 1–10, 2004, [Online]. Available: http://rosni-gj.staff.gunadarma.ac.id/Downloads/files/15431/pendahuluan.pdf

[22] Sidharta, H. A., "Introduction to Open CV," 2017. https://binus.ac.id/malang/2017/10/introduction-to-open-cv/

[23] Syaikhoni, A., A. Ariyadi, "Deteksi Objek dengan Tensorflow Object Detection API," 2018. https://mti.binus.ac.id/2018/12/26/deteksi-objek-dengan-tensorflow-object-detection-api/ (diakses Jul. 18, 2023).

[24] Pratiwi, B. M., N. Q. Nada, "Penerapan Model Machine Learning dalam Menentukan Rekomendasi Objek Wisata Provinsi Jawa Tengah," *Sci. Eng. Natl. Semin.*, vol. 7, no. 7, 2022.

[25] Redmon, J., S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection Joseph," *Univ. of Washington, Allen Inst. AI, Faceb. AI Res.*, 2018.

[26] Akhtar, H., "Perbedaan Adaptasi, Modifikasi, dan Konstruksi Skala," 2017. https://www.semestapsikometrika.com/2017/09/perbedaan-adaptasi-modifikasi-dan.html (diakses Jul. 18, 2023).

[27] Admin, "Model Optimisasi," *Univ. Negeri Yogyakarta*, pp. 1–12, 2016.

[28] Admin, "Colaboratory: Frequently Asked Questions," *Google Colaboratory*. 2018.

[29] Carneiro, T., R. V. M. Da Nobrega, T. Nepomuceno, G. Bin Bian, V. H. C. De Albuquerque, P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018.

[30] Saxena, A., "An Introduction to Convolutional Neural Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 943–947, 2022.

[31] Menegaz, M., "Understanding YOLO," *Hacker Noon*, 2018. https://hackernoon.com/understanding-yolo-f5a74bbc7967 (diakses Jul. 18, 2023).

[32] Rosebrock, A., "Intersection over Union (IoU) for object detection," 2016. https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-

object-detection/ (diakses Jul. 18, 2023).

[33] Admin, "Simple Guide to Confusion Matrix Terminology," *Data School* 2014. https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

[34] Hadi, A. P., "COMPUTER VISION ? PENGERTIAN, CONTOH, DAN APLIKASINYA," 2023. https://komputer-grafis d3.stekom. ac.id/informasi/baca/COMPUTER-VISION-PENGERTIAN-CONTOH DAN APLIKASINYA/5dbdf66075ce5e47cb39584288bc8b7dc3333b1b (diakses Jul. 28, 2023).

[35] Hidayatulloh, M. S., "Sistem Pengenalan Wajah Menggunakan Metode Yolo (You Only Look Once)," *Tugas Akhir Fakultas Teknologi dan Informatika Universitas Dinamika*, 2021.

[36] Ubaidillah, A., A. F. Ibadillah, M. M. F. Nur, "Deteksi Jumlah Pengunjung Dan Penggunaan Masker Dengan Menggunakan Metode YOLO Dan Haar Cascade Classifier," *Journal Zetroem*, vol. 5, no. 1, pp. 10–18, 2023.

[37] Widjaja, P. A., R. Theo, K. Liem, "Penggunaan YOLOv4 Untuk Menentukan Lokasi Dosen Dan Mahasiswa Dengan Menggunakan CCTV," *Infinity*, vol. 2, no. 1, pp. 2–5, 2022.

[38] Lubis, C., N. J. Perdana, "SISTEM PENDETEKSIAN DAN PENGENALAN EKSPRESI WAJAH DENGAN ALGORITMA YOLO DAN CONVOLUTIONAL NEURAL NETWORK," 2020. [Online]. Available:http://repository.radenintan.ac.id/11375/1/PERPUS PUSAT.pdf%0Ahttp://business-law.binus.ac.id/2015/10/08/pariwisata-syariah/%0Ahttps://www.ptonline.com/articles/how-to-get-better-mfi-results%0Ahttps://journal.uir.ac.id/index.php/kiat/article/view/8839

[39] Adinata, Y., K. Gunadi, I. Sugiarto, "Aplikasi Deteksi Jumlah Orang pada Area Indoor Untuk Mendukung Pelaksanaan PPKM dengan Metode YOLO," *J. Infra*, vol. 10, no. 1, pp. 135–141, 2022, [Online]. Available: https://publication.petra.ac.id/index.php/teknikinformatika/article/view/12031%0Ahttps://publication.petra.ac.id/index.pp/teknikinformatika/article/viewFile/12031/10566

[40] Salamah, I., M. R. A. Said, S. Soim, "Perancangan Alat Identifikasi Wajah

Dengan Algoritma You Only Look Once (YOLO) Untuk Presensi Mahasiswa," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1492, 2022.

LAMPIRAN

Lampiran A

Listing Code Deteksi wajah

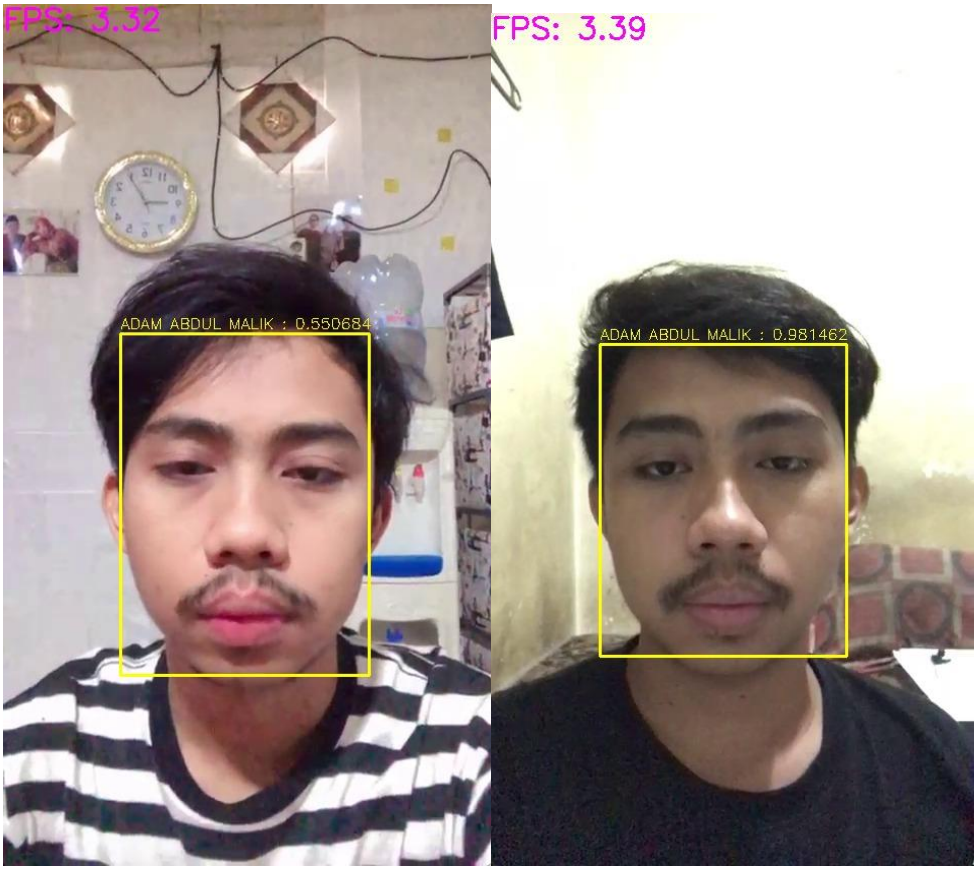
```
 import cv2
 import time

 CONFIDENCE_THRESHOLD = 0.2
 NMS_THRESHOLD = 0.4
 COLORS = [(0, 255, 255), (255, 255, 0), (0, 255, 0), (255, 0, 0)]

 class_names = []
 with open ("classes.txt", "r") as f:
 class_names = [cname.strip() for cname in f.readlines()]

cap = cv2.VideoCapture("ADAMFIX.mp4")

net = cv2.dnn.readNet("yolov4.weights", "yolov4.cfg")
#net.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
#net.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA_FP32)

model = cv2.dnn_DetectionModel(net)
model.setInputParams(size=(608, 608), scale=1/255, swapRB=True)
while cv2.waitKey(1) < 1:
 (grabbed, frame) = cap.read()
 if not grabbed:
  exit()

 start = time.time()
 classes, scores, boxes = model.detect(frame,
CONFIDENCE_THRESHOLD, NMS_THRESHOLD)
 end = time.time()

 for (classid, score, box) in zip(classes, scores, boxes):
  color = COLORS[int(classid) % len(COLORS)]
  label = "%s : %f" % (class_names[classid[0]], score)
  cv2.rectangle(frame, box, color, 2)
  cv2.putText(frame, label, (box[0], box[1]-5),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 1)

 fps = "FPS: %.2f " % (1 / (end - start))
```

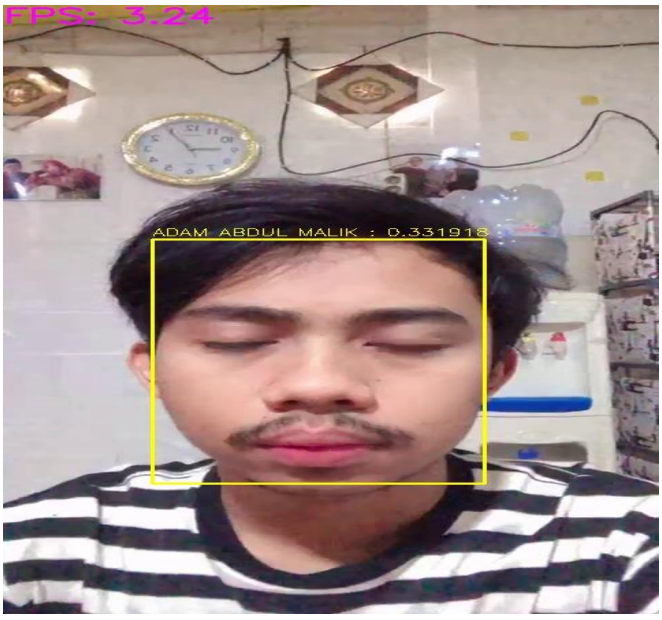
```
 cv2.putText(frame, fps, (0, 25), cv2.FONT_HERSHEY_SIMPLEX, 1,
(255, 0, 255), 2)
    cv2.imshow("output", frame)
```

Lampiran B

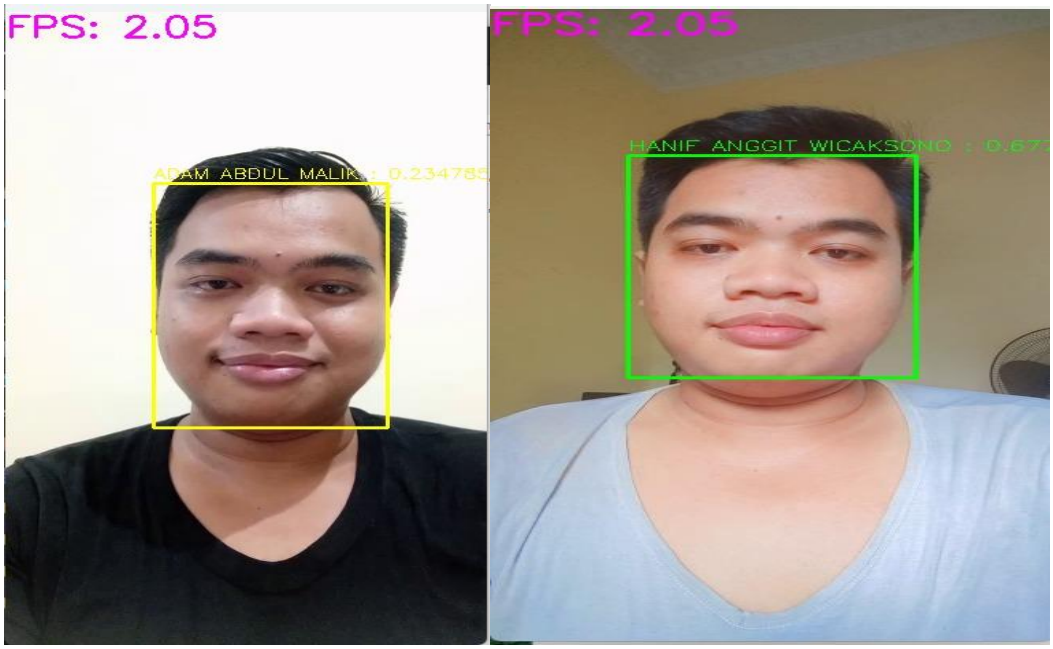
Hasil identifikasi wajah bagian sisi depan

ADAM ABDUL MALIK

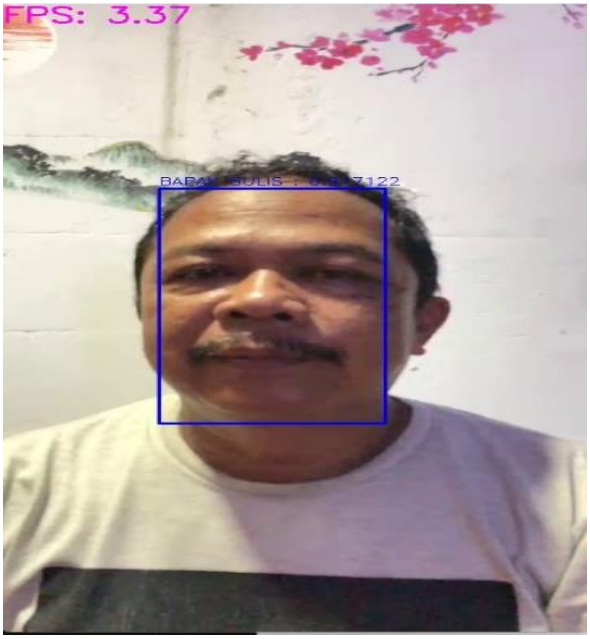




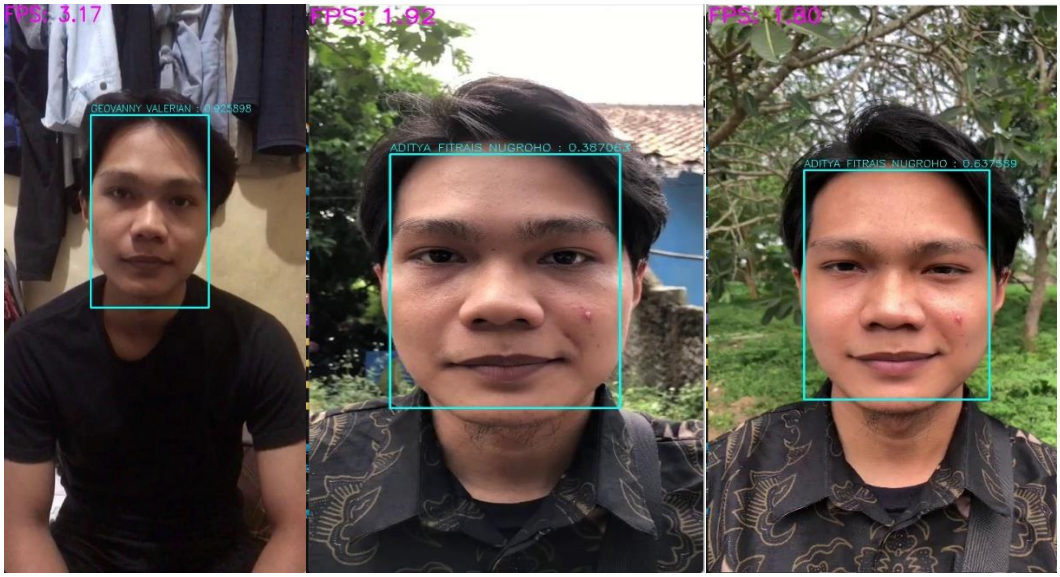
ANANDA SAKSENA SIWI



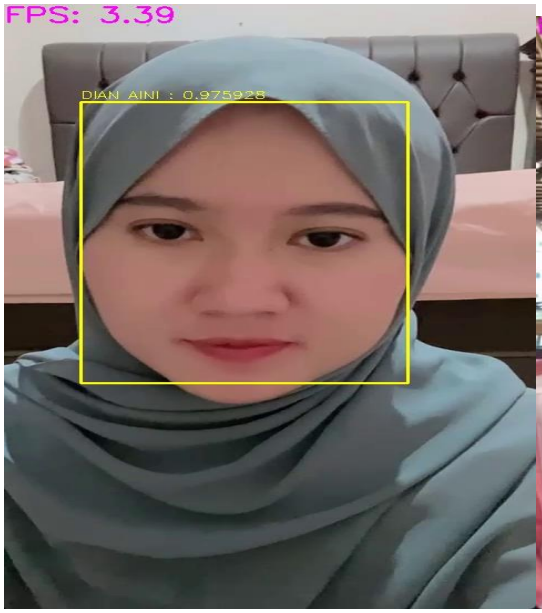
BAPAK SULIS

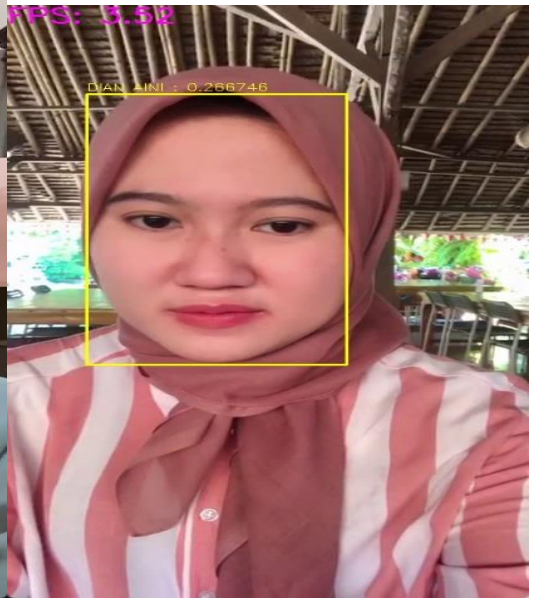


BIDIN


FPS: 3.17
GEOVANNY VALERIAN : 0.925898
FPS: 1.92
ADITYA FITRAIS NUGROHO : 0.387063
FPS: 1.80
ADITYA FITRAIS NUGROHO : 0.637389

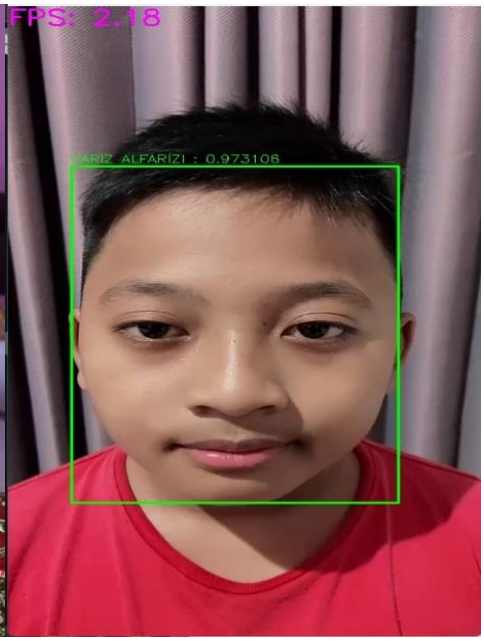
DIAN AINI

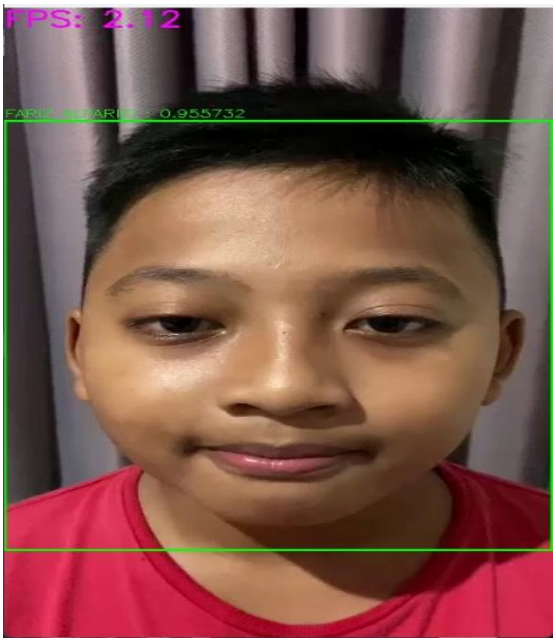




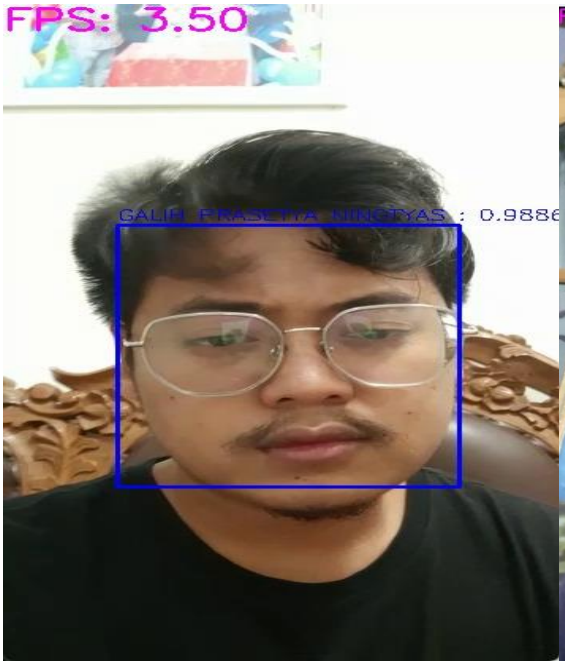
DILA SEPTIYANI

FARIZ ALFARIZI





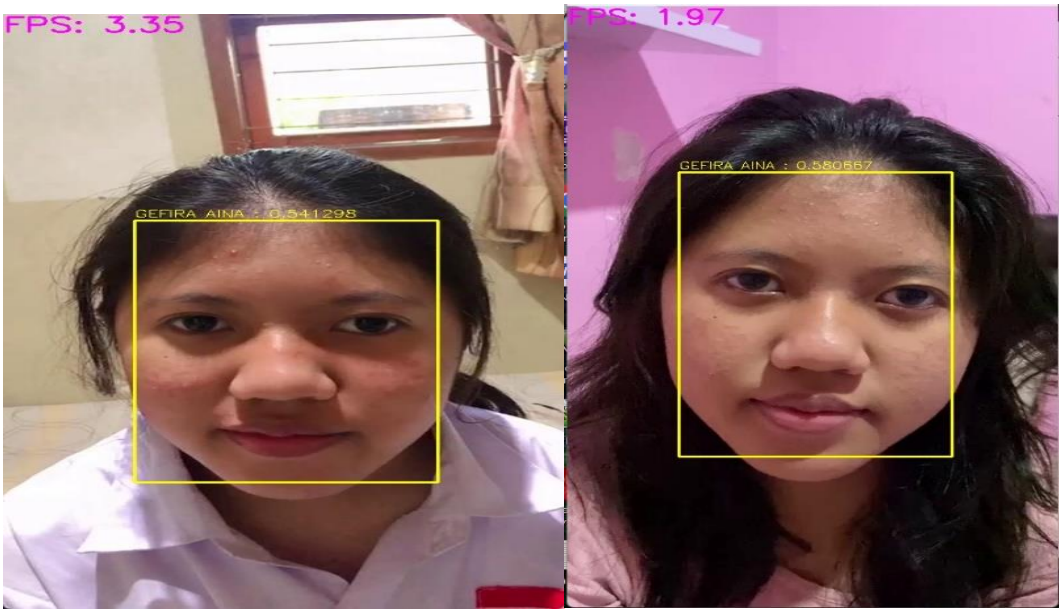
GALIH PRASETYA

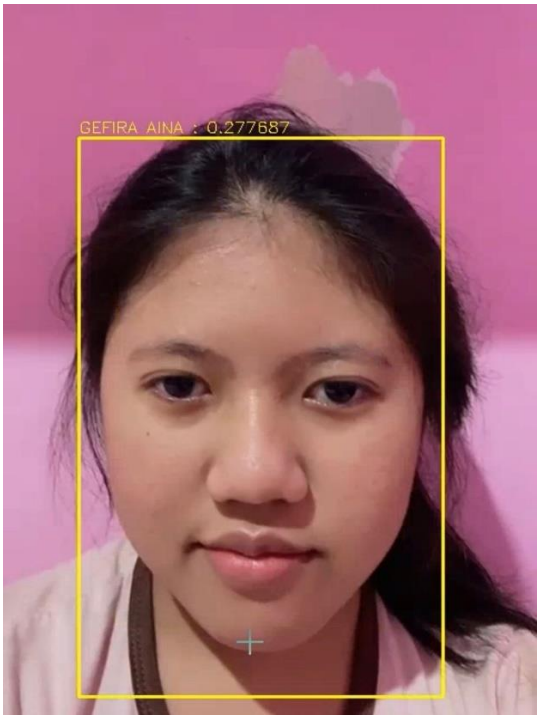

FPS: 3.50


FPS: 1.83


FPS: 1.95

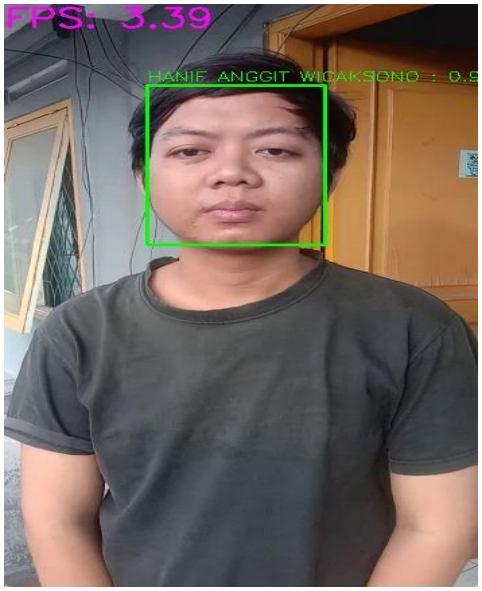
GEFIRA AINA



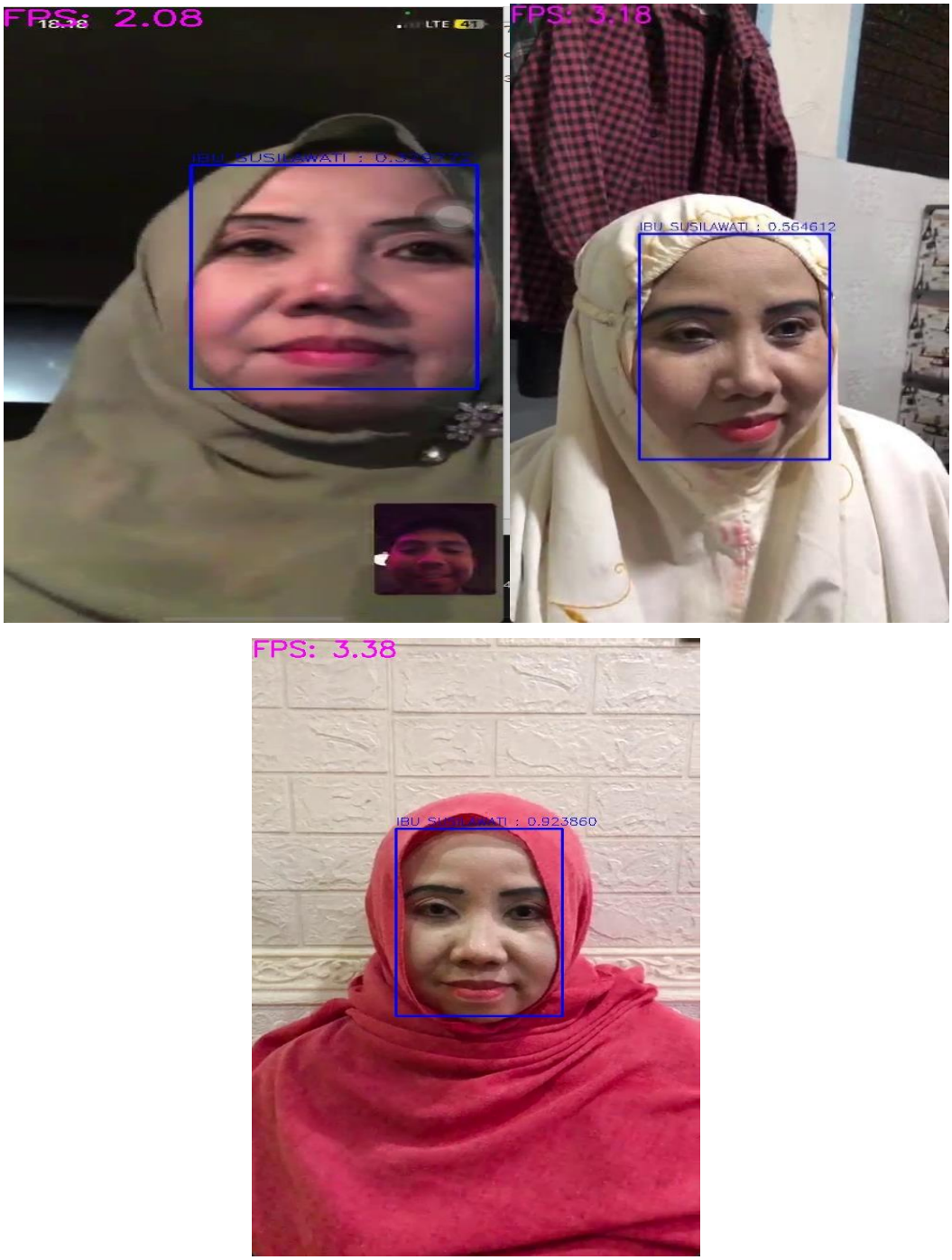


GEOVANY VALERIAN

HANIF ANGGIT

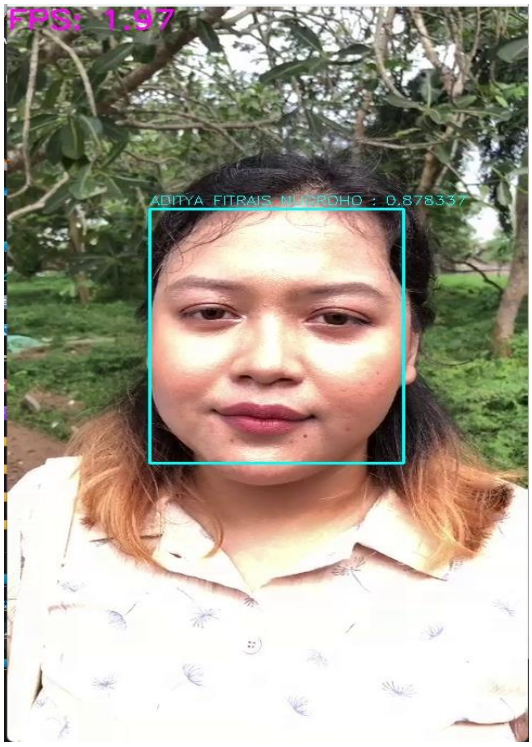


IBU SUSILAWATI

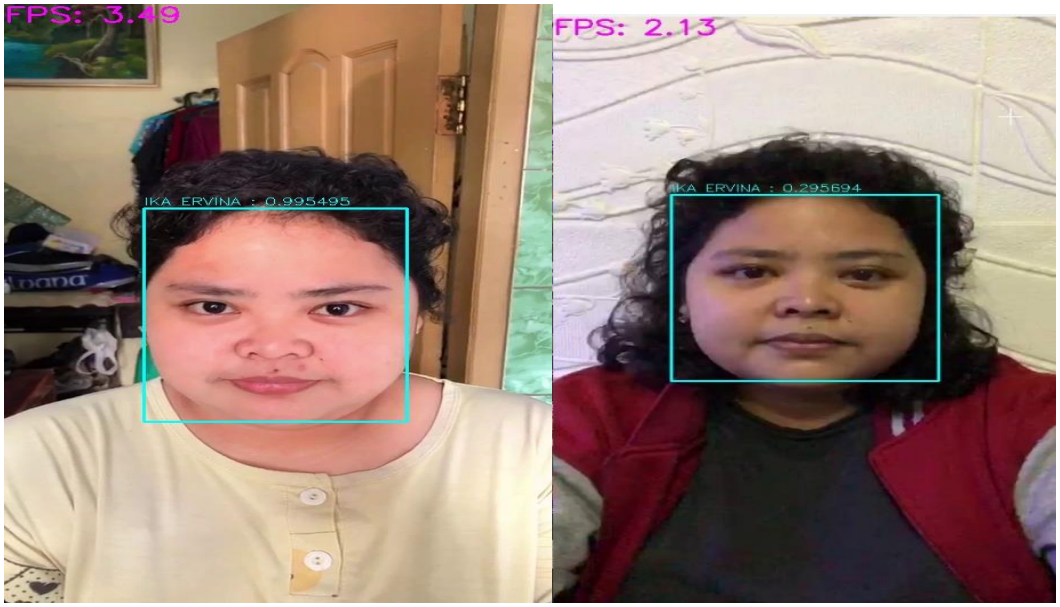


ICAH NURASIYAH

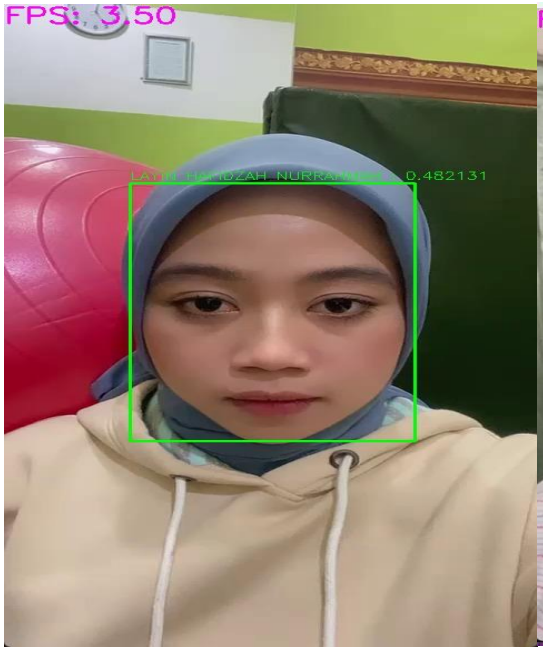

FPS: 3.17
ICAH NURAISYAH : 0.987299
FPS: 3.16
ICAH NURAISYAH : 0.999441

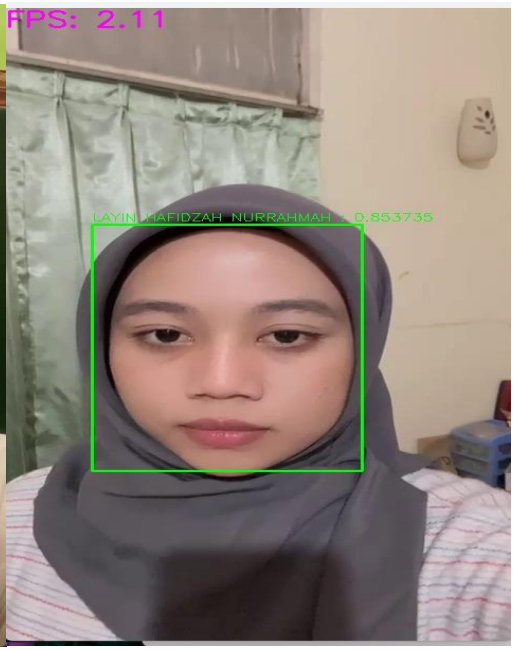

FPS: 1.97
ADITYA FITRAIS MUDRDHO : 0.878337

IKA ERVINA

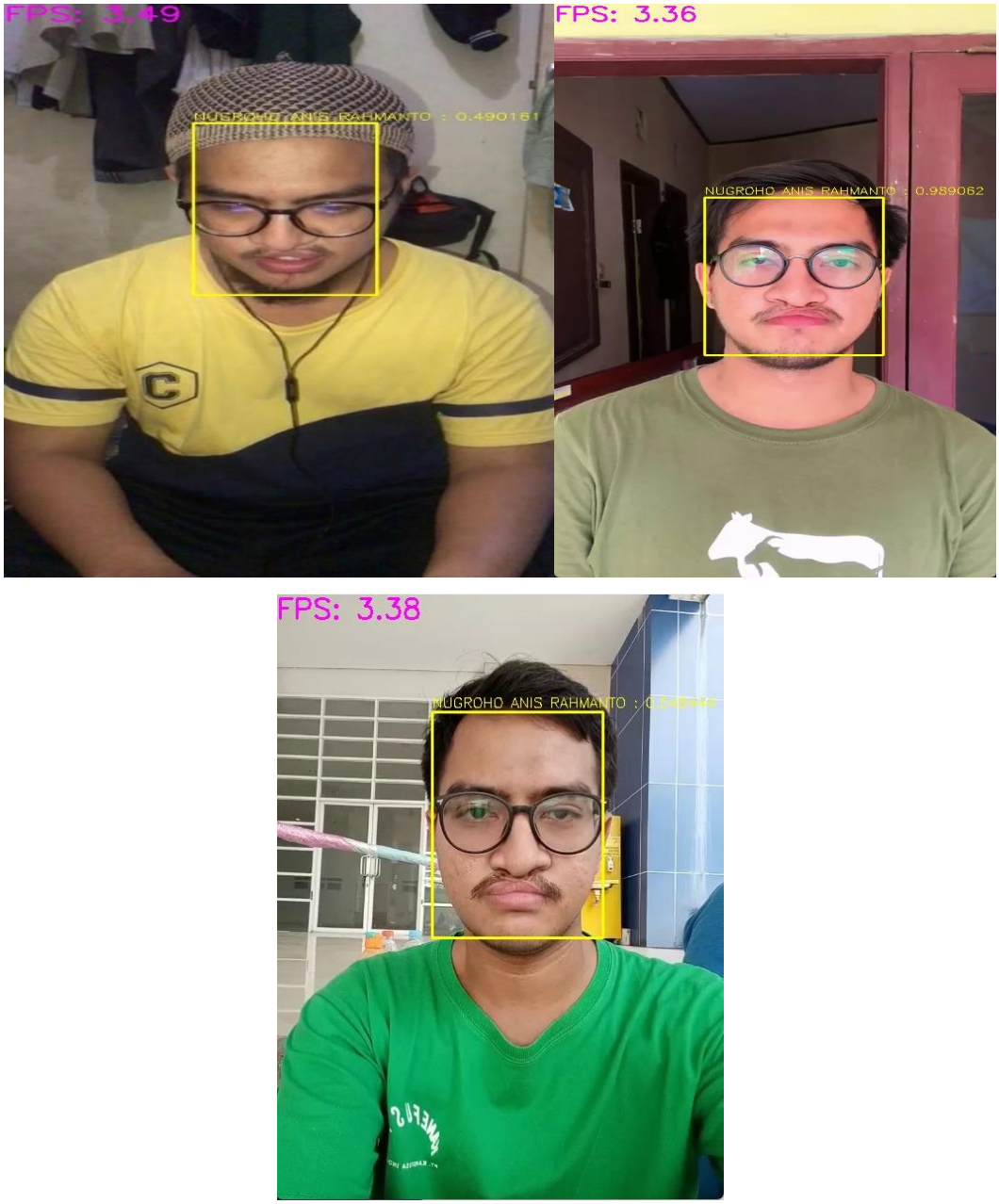


LAYIN HAFIDZAH

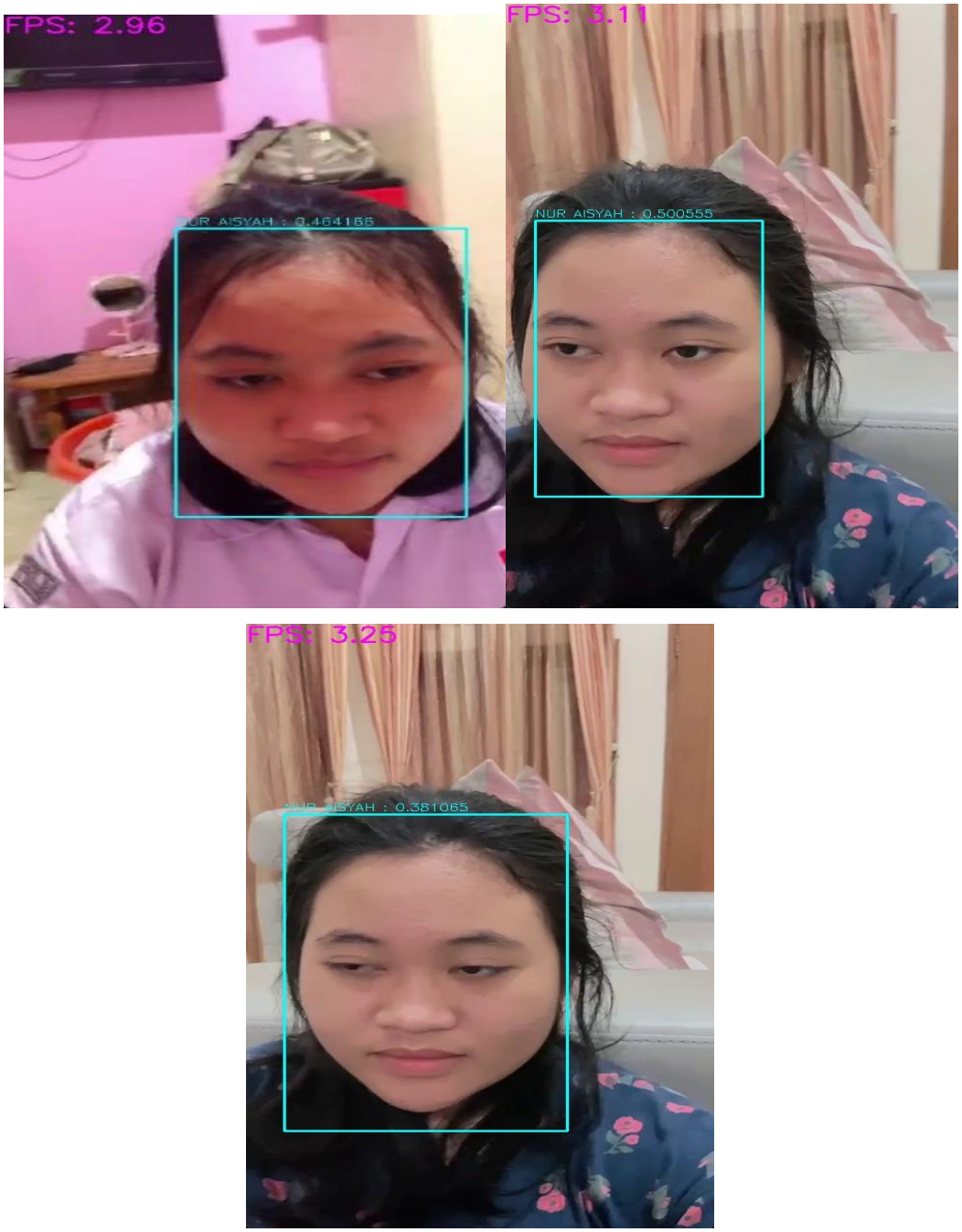




NUGROHO ANIS


FPS: 3.49
NUGROHO ANIS RAHMANTO : 0.490161
FPS: 3.36
NUGROHO ANIS RAHMANTO : 0.989062
FPS: 3.38
NUGROHO ANIS RAHMANTO :

NUR AISYAH


FPS: 2.96
NUR AISYAH : 0.464165
FPS: 3.11
NUR AISYAH : 0.500555
FPS: 3.25
NUR AISYAH : 0.381065

PANDU AKBAR


FPS: 3.07
PANDU AKBAR MAULANA : 0.995516
FPS: 3.11
PANDU AKBAR MAULANA : 0.933049

RIFALDI


FPS: 3.07
RIFALDI TRYAWAN : 0.936241