

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.

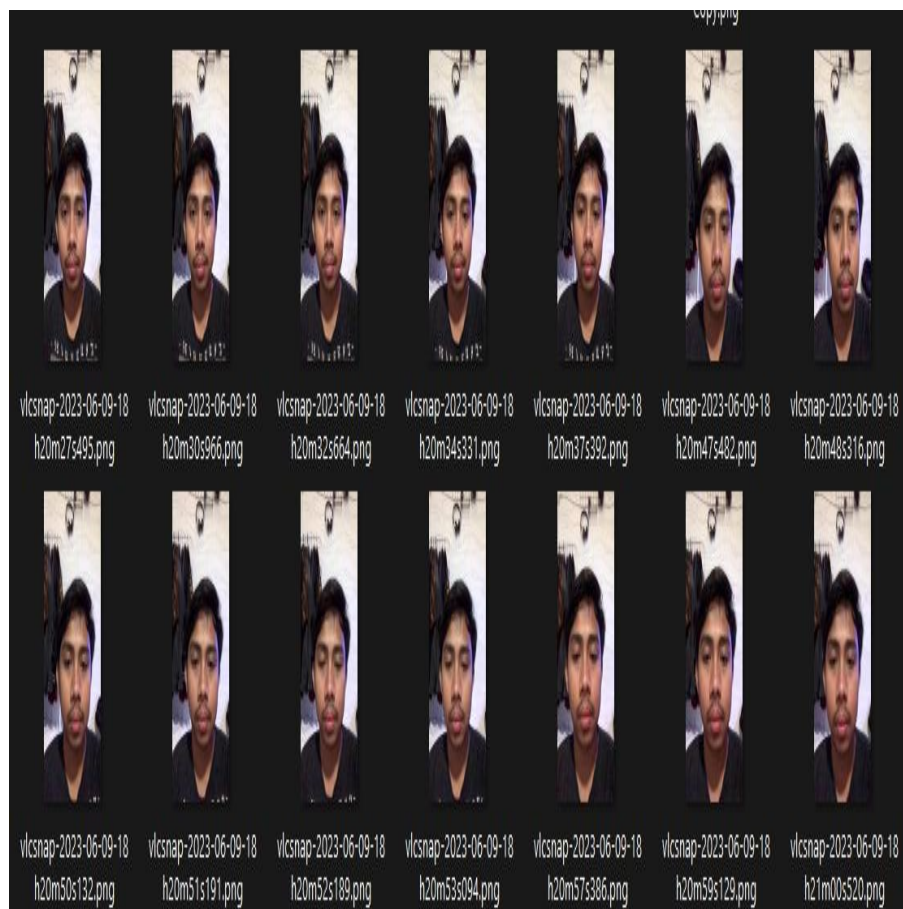
Tabel 4.1 Jumlah Data Latih

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.

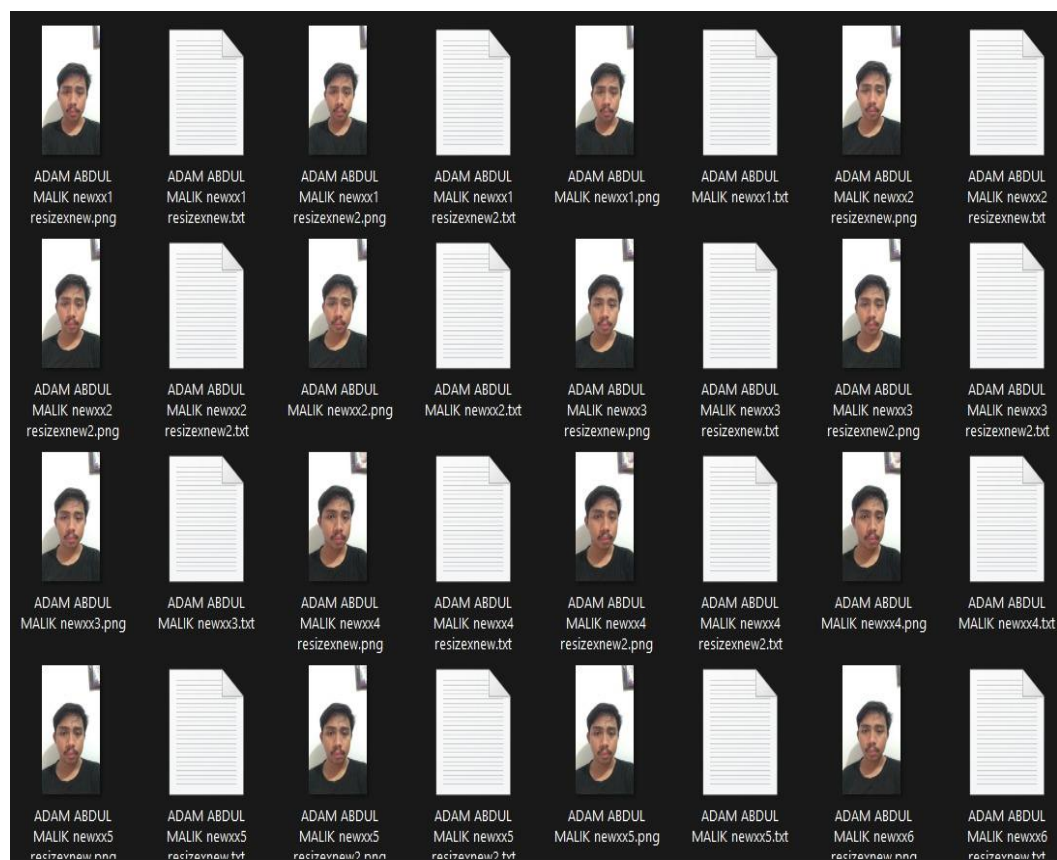
4.2. Hasil Pelabelan Objek

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.

```

0 0.726042 0.604953 0.547917 0.384434
11 0.493750 0.510024 0.516667 0.406840

```

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 weights*. File data terdapat letak citra yang implementasikan untuk data latih dan percobaan. File CFG terdapat 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
<i>Load Model</i>	Darknet
<i>Load Weight</i>	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 perbandingan. Suatu perulangan telah memperoleh angka 10000, lalu proses *training* otomatis berhenti. Nilai *max_batches* didapatkan dari Persamaan 4.1.

$$max_batches = jumlahclass.2000.....(4.1)$$

Nilai *step* didapatkan pada Persamaan 4.2 berikut ini.

$$steps = (80\%max_batches), (90\%max_batches).....(4.2)$$

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

$$filter = (jumlahclass + 2) \times 3(4.3)$$

Tabel 4. 3 Konfigurasi pada *weights* YOLOv4

Jenis Konfigurasi	Keterangan
<i>Batch</i>	64
<i>Subdivisions</i>	64
<i>Width</i>	608
<i>Height</i>	608
<i>max_batches</i>	40000
<i>Steps</i>	32000, 36000
<i>Classes</i>	20
<i>Filters</i>	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 terdeteksi 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.

```

28665: 0.015915, 0.009802 avg loss, 0.002610 rate, 0.746773 seconds, 1834560 images, 2.409669 hours left
Loaded: 0.000000 seconds
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.885576), count: 4, class_loss = 0.126026, iou_loss = 0.182285, total_loss = 0.229110
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548744, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.896618), count: 4, class_loss = 0.026531, iou_loss = 0.181656, total_loss = 0.208187
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548748, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.893450), count: 4, class_loss = 0.085537, iou_loss = 0.181783, total_loss = 0.167321
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548752, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.874822), count: 4, class_loss = 0.100197, iou_loss = 0.141534, total_loss = 0.241731
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548756, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.922782), count: 4, class_loss = 0.000000, iou_loss = 0.112251, total_loss = 0.112253
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548760, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.922281), count: 4, class_loss = 0.033142, iou_loss = 0.097320, total_loss = 0.130461
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548764, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.912847), count: 4, class_loss = 0.044676, iou_loss = 0.284071, total_loss = 0.328747
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548768, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.849244), count: 4, class_loss = 0.001411, iou_loss = 0.128062, total_loss = 0.121473
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548772, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.846514), count: 4, class_loss = 0.000040, iou_loss = 0.168349, total_loss = 0.168389
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548776, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.845774), count: 4, class_loss = 0.004629, iou_loss = 0.165742, total_loss = 0.170371
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548780, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.879721), count: 4, class_loss = 0.000002, iou_loss = 0.250565, total_loss = 0.250567
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548784, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.868190), count: 4, class_loss = 0.388397, iou_loss = 0.176635, total_loss = 0.565532
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
total_bbox = 548788, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.923792), count: 4, class_loss = 0.000020, iou_loss = 0.169389, total_loss = 0.169329
v3 (iou loss, Normalizer: (iou: 0.87, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000

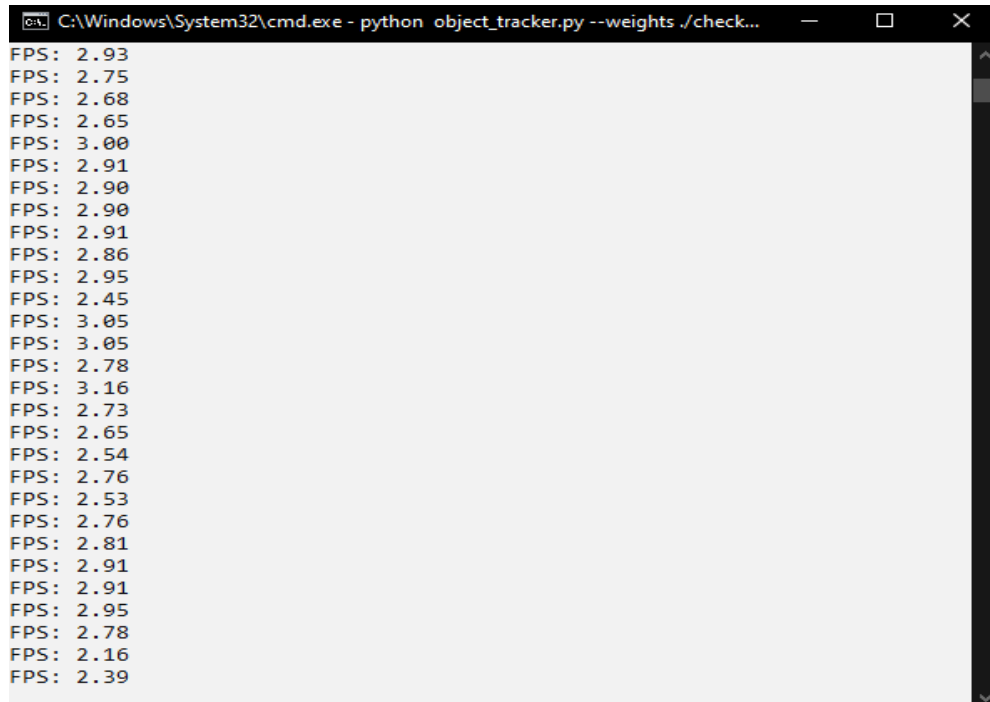
```

Gambar 4.4 Proses *Training* Data Latih

Berdasarkan Gambar 4.4 dapat dilihat diatas bisa disimpulkan 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.

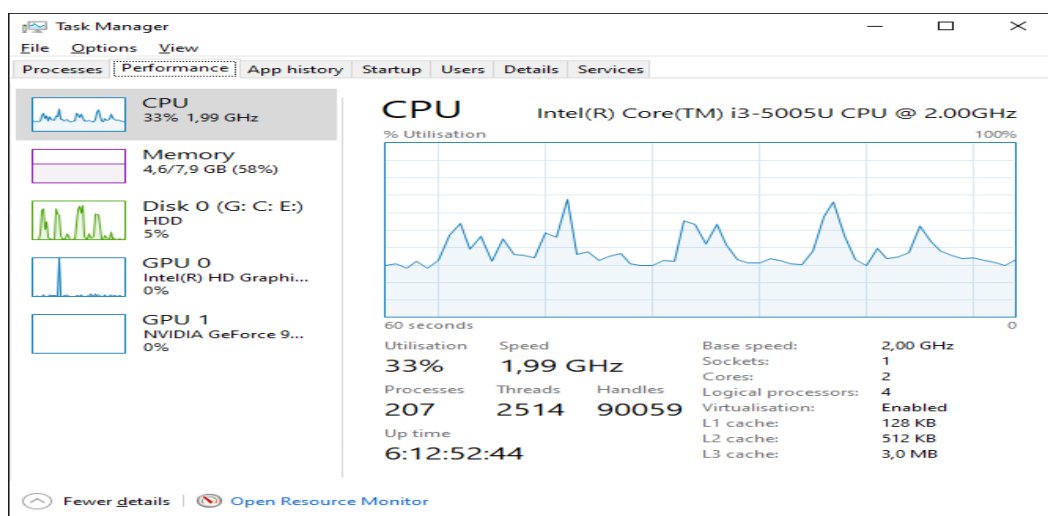
4.4 Hasil Kinerja Alat

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.



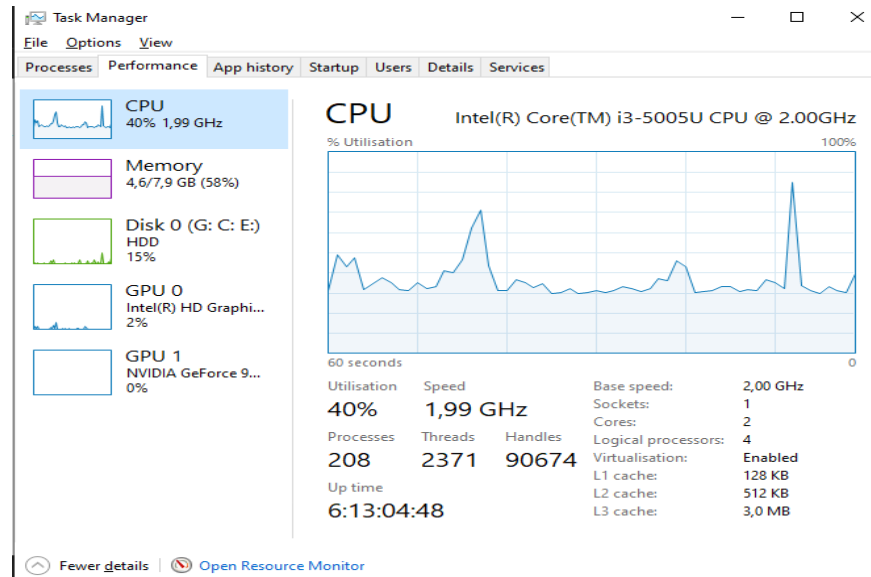
Gambar 4.5 FPS saat Program dijalankan

Berdasarkan Gambar 4.5, menunjukkan FPS yang ditangkap oleh sistem rata-rata sebesar 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 akurasi 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.



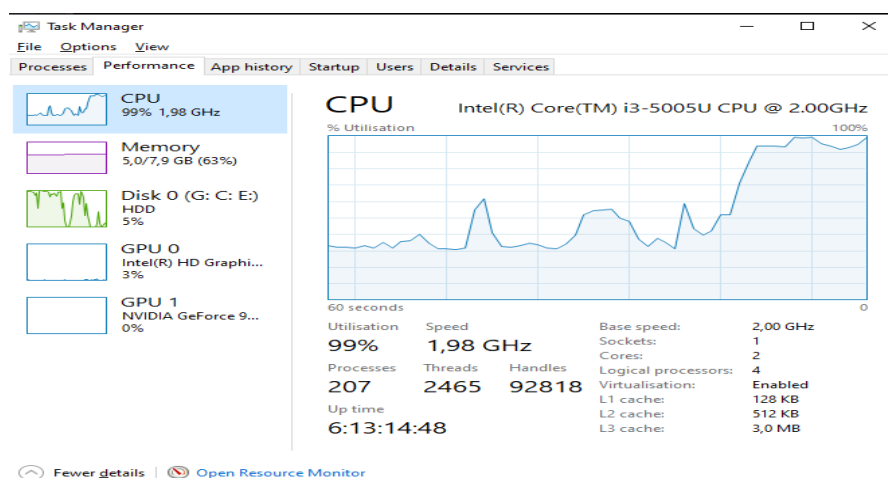
Gambar 4.6 CPU saat tidak membuka Program

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 python 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.

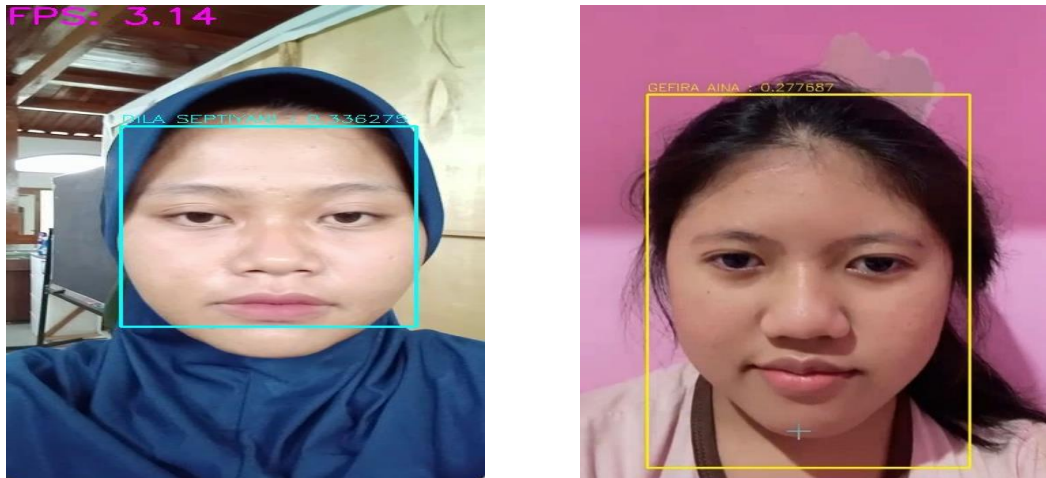
4.5. Pengujian Deteksi Identifikasi Wajah

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.



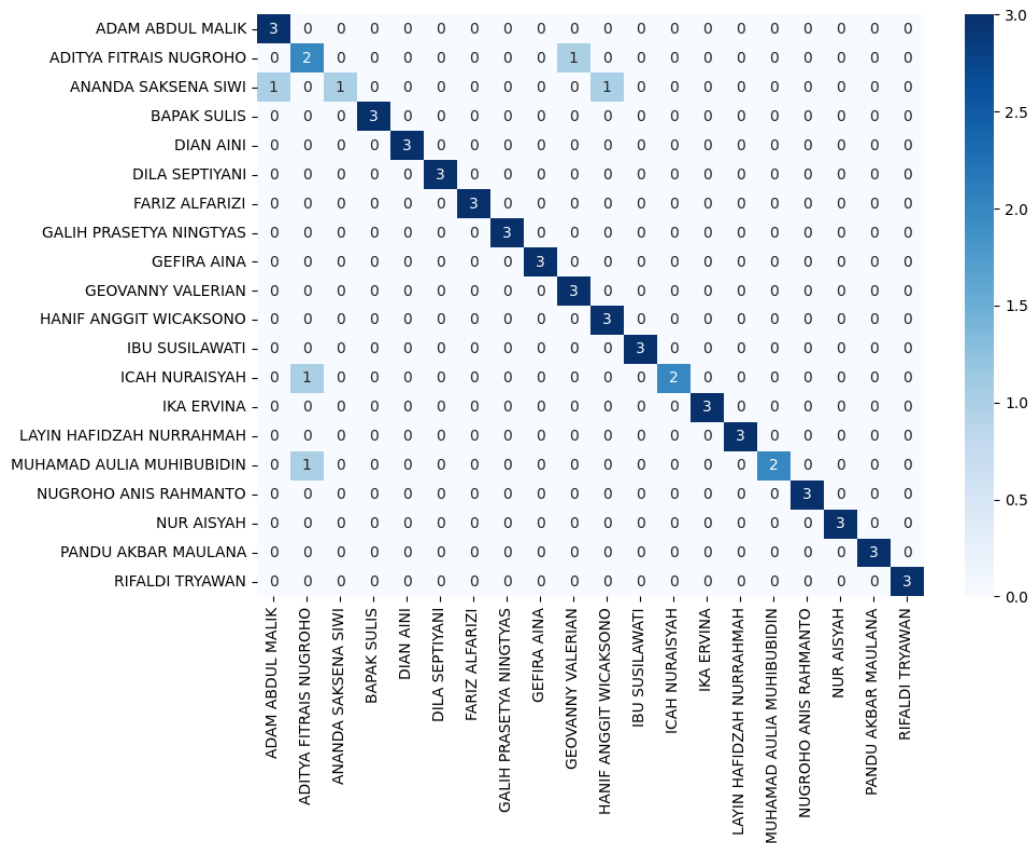


Gambar 4. 9 Hasil Pengujian pada Sisi Depan

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 melewati 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 identifikasi 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 diartikan di atas 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 di atas bisa didapatkan nilai *true positive*, *false positive*, *false negative*, dan *true negative* dengan tabel di bawah ini. Hasil nilai pengujian bisa dilihat pada Tabel 4.5 berikut.

Tabel 4.5 Hasil seluruh pengujian identifikasi wajah pada posisi depan

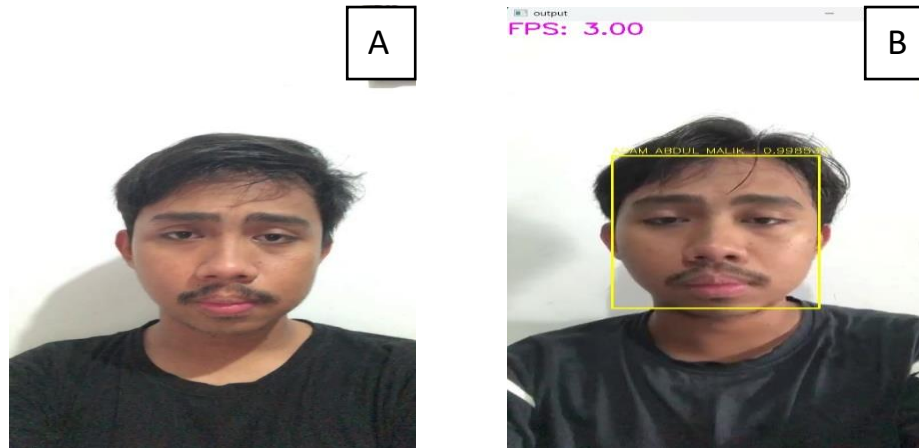
Data Wajah	Jumlah	TP	FP	FN	TN
Adam Abdul Malik	3	3	0	1	52
Aditya Fitrais Nugroho	3	2	1	2	53
Ananda Saksena Siwi	3	1	2	0	54
Bapak Sulis	3	3	0	0	52
Dian Aini	3	3	0	0	52
Dila Septiyani	3	3	0	0	52
Fariz Alfarizi	3	3	0	0	52

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 Muhububidin	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.

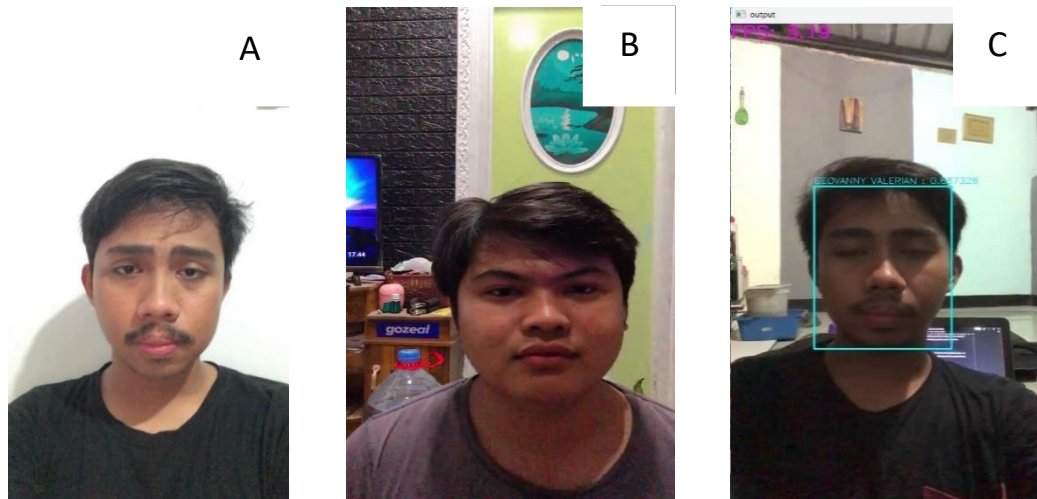


Gambar 4.10 Hasil Perbandingan Pengujian Kedua
 A) Gambar dari Data Latih B) Gambar Hasil Pengujian

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 keadaan 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.



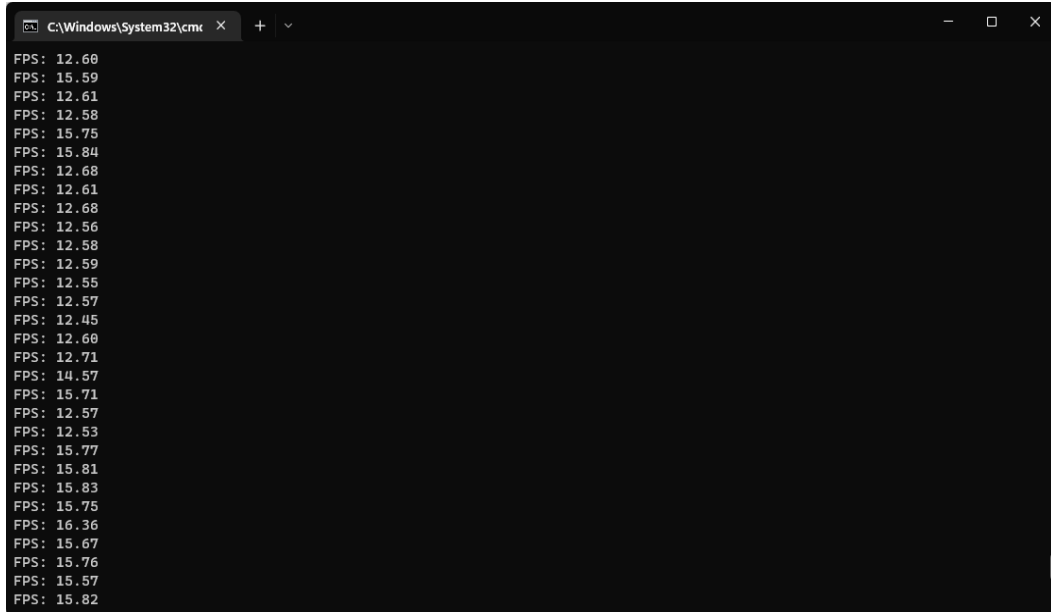
Gambar 4.12 Hasil Perbandingan Pengujian Keempat

- 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.

A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\Windows\System32\cmd'. The window contains a list of FPS values, one per line, ranging from 12.60 to 15.82. The values are: 12.60, 15.59, 12.61, 12.58, 15.75, 15.84, 12.68, 12.61, 12.68, 12.56, 12.58, 12.59, 12.55, 12.57, 12.45, 12.60, 12.71, 14.57, 15.71, 12.57, 12.53, 15.77, 15.81, 15.83, 15.75, 16.36, 15.67, 15.76, 15.57, and 15.82.

```
C:\Windows\System32\cmd
FPS: 12.60
FPS: 15.59
FPS: 12.61
FPS: 12.58
FPS: 15.75
FPS: 15.84
FPS: 12.68
FPS: 12.61
FPS: 12.68
FPS: 12.56
FPS: 12.58
FPS: 12.59
FPS: 12.55
FPS: 12.57
FPS: 12.45
FPS: 12.60
FPS: 12.71
FPS: 14.57
FPS: 15.71
FPS: 12.57
FPS: 12.53
FPS: 15.77
FPS: 15.81
FPS: 15.83
FPS: 15.75
FPS: 16.36
FPS: 15.67
FPS: 15.76
FPS: 15.57
FPS: 15.82
```

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.