

## **BAB IV**

### **ANALISA DATA DAN PEMBAHASAN**

Hasil pembahasan dan pengujian yang dilakukan oleh penulis akan dijelaskan pada bab ini. Tujuan dari bab ini adalah untuk mengetahui tingkat keberhasilan dari perancangan sistem yang telah dikerjakan. Dimana pengujian yang dilakukan meliputi, tingkat akurasi objek yang terdeteksi, nilai dari FPS (*Frame per second*), dan jumlah objek yang terdeteksi.

#### **4.1 Pengujian Tingkat Akurasi Pada ARM Prosessor Nanopi M4V2**

Pada prose pengujian tingkat akurasi ini digunakan untuk mengetahui seberapa tingkat akurasi dari perangkat yang telah dibuat sehingga dapat dikatakan bahwa perangkat tersebut dapat digunakan secara baik dengan tingkat akurasi yang sempurna.

##### **4.1.1 Tujuan Pengujian Tingkat Akurasi**

Pada penelitian ini pengujian tingkat akurasi pada penelitian ini bertujuan untuk apakah perangkat dapat mendeteksi objek dengan sempurna. Pada pengujian ini apabila webcam atau vidio yang dimasukkan terdapat objek manusia maka sistem akan mendeteksi dan memberikan sebuah tanda berupa rectangle tepat dibagian wajah objek dan akan menampilkan nilai tingkat akurasi

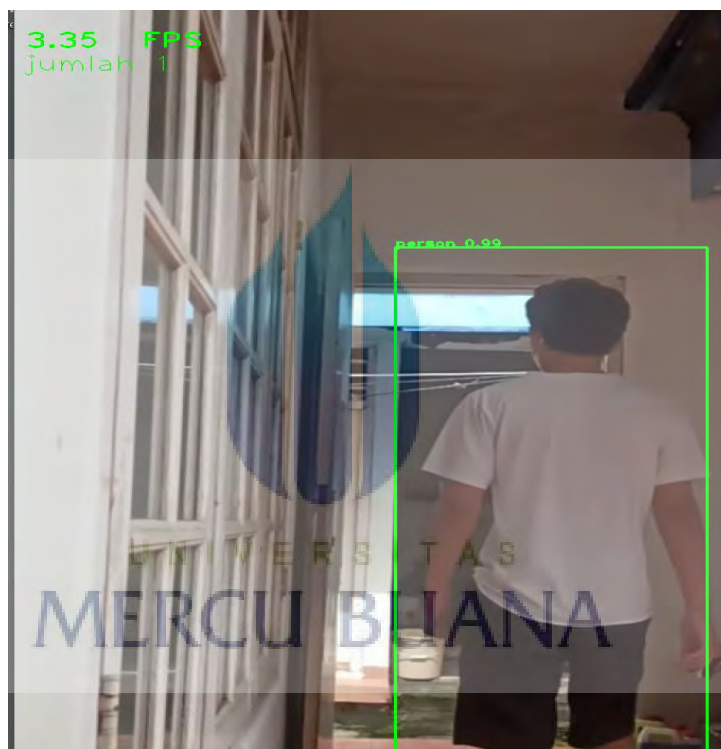
##### **4.1.2 Prosedur Pengujian Tingkat Akurasi**

Dalam melakukan pengujian tingkat akurasi pada sistem pendeteksian objek dibutuhkan beberapa prosedur yang perlu dilakukan sehingga perangkat bisa bekerja optimal. berikut prosedur tersebut :

1. Memasukkan data *training* yang berformat .weights, kedalam program
2. Menjalankan program yang sudah dibuat dalam format \*Py

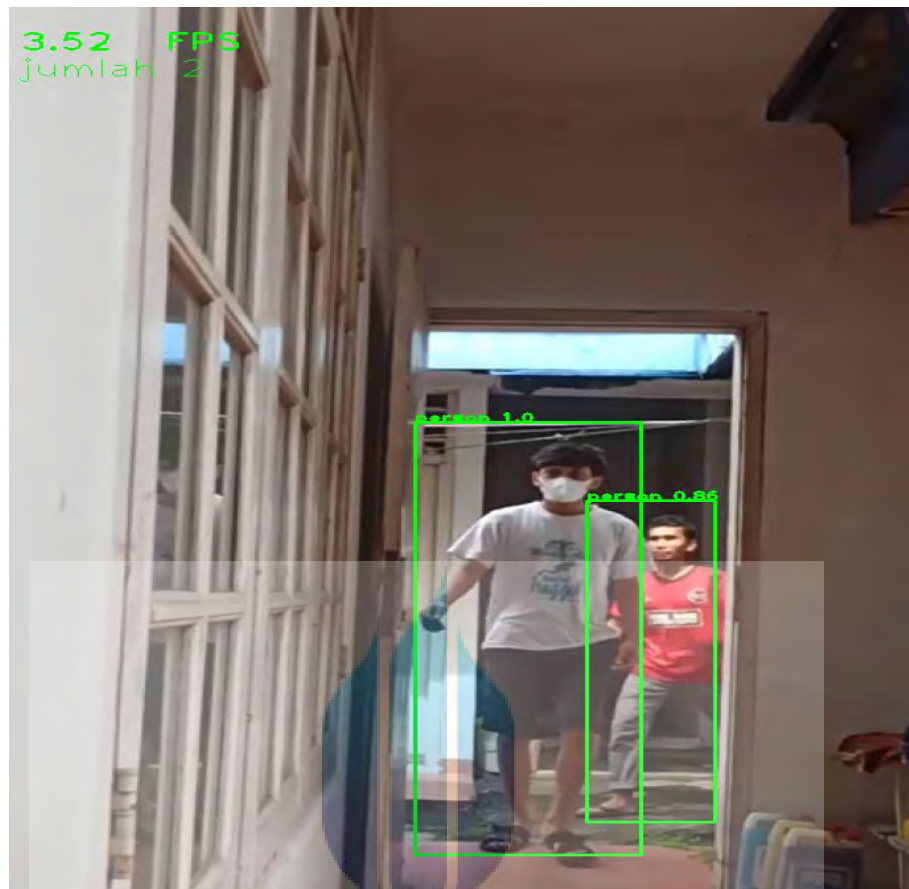
#### 4.1.3 Hasil Pengujian Tingkat Akurasi Pada ARM Processor NanoPi M4V2

Pada saat melakukan pemrosesan, video rekaman yang berbentuk video, dimasukkan kedalam program yang telah dibuat, selanjutnya video akan diproses oleh program, hasil pemrosesan akan ditampilkan berupa nilai fps dari video tersebut, tingkat akurasi dan nama objek yang terdeteksi. Berikut merupakan contoh dari pemrosesan video dengan masukan secara manual.



Gambar 4.1 Hasil 1 Objek Hadap Belakang

Pada hasil tangkapan layar diatas, dilakukan percobaan dengan objek membelakangi kamera dimana didapatkan hasil bahwa objek terdeteksi dengan dengan tingkat akurasi yang tinggi dan nilai dari FPS yang tidak terlalu rendah.



Gambar 4.2 Hasil 2 Objek Hadap Depan

MERCU BUANA

Pada tangkapan layar diatas, dilakukan percobaan dengan menggunakan 2 buah objek manusia yang dimana terdapat objek yang menggunakan masker dan yang tidak menggunakan masker, dengan hasil kedua objek tersebut terdeteksi oleh perangkat dan mendapatkan tingkat akurasi yang tinggi dengan nilai FPS yang menurun.



Gambar 4.3 Hasil Pada Objek Yang Banyak

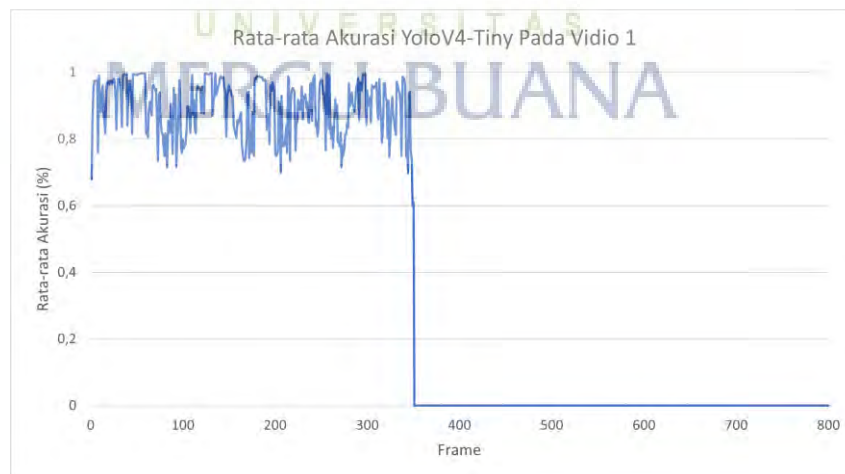
Pada tangkapan layar diatas, dilakukan percobaan dengan menggunakan objek manusia yang berjalan di jalan raya dengan terdapat objek seperti tiang sebagai objek yang menyerupai manusia, dari percobaan tersebut didapatkan hasil objek manusia yang berjalan terdeteksi oleh perangkat dengan tingkat akurasi yang tinggi dan objek yang lain tidak terdeteksi sedangkan nilai FPS yang sedikit tinggi.

Dari ketiga gambar diatas didapatkan tingkat akurasi yang tinggi dan sedikit penurunan nilai FPS. Selain itu tingkat akurasi pada pendeteksian objek tidak terpengaruh dengan posisi objek, apakah membelakangi kamera atau menghadap kamera, dikarenakan nilai akurasi masih diatas 90%. Berikut tabel dari tangkapan layar diatas.

Tabel 4.1 Deteksi objek YOLOv4-Tiny

No	Objek	Akurasi (%)	Jumlah Terdeteksi	Jumlah Objek
1	1 Objek	0.99	1	1
2	2 Objek	- 1.0 - 0.86	2	2
3	Banyak Objek	- 0.81 - 0.87 - 0.6 - 0.84 - 0.81	5	5
FPS		2.81 – 3.55		

Dari tabel diatas dapat dilihat bahwa dengan menggunakan metode YOLOv4-Tiny kita dapat melakukan pemrosesan citra digital dengan mendeteksi objek khususnya objek manusia dengan tingkat akurasi yang tinggi dan tidak terpengaruh dengan objek yang lainnya.



Gambar 4.4 Rata-rata Akurasi Vidio 1 dengan ARM Nanopi M4V2

Pada gambar 4.4 menampilkan rata rata tingkat akurasi dari objek yang dideteksi dimasing masing frame, dapat dilihat bahwa tingkat akurasi yang didapatkan sangat tinggi dengan rata-rata diatas 0,5% yang artinya tingkat akurasi pada vidio sangatlah baik.

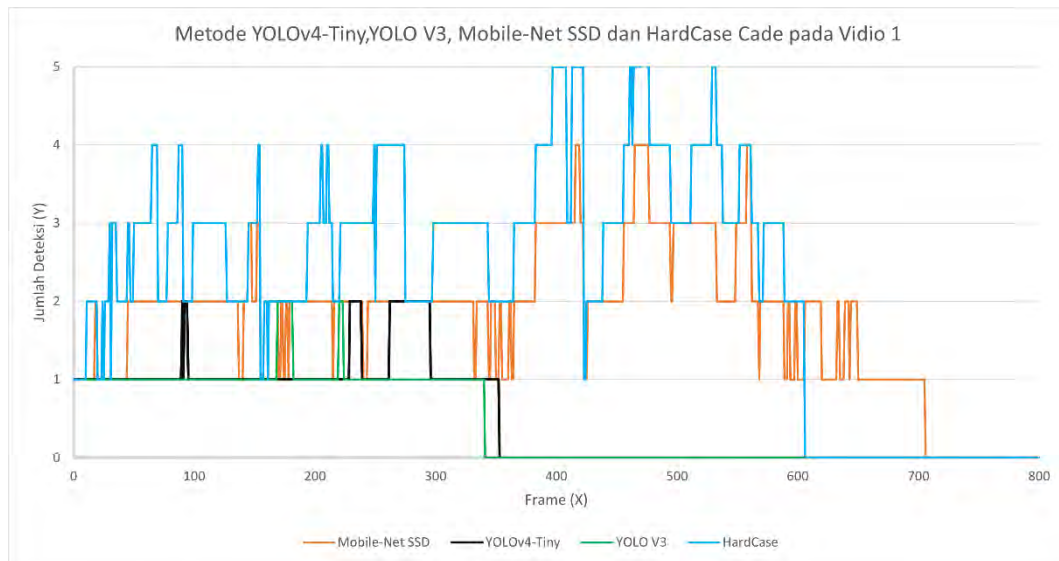


Gambar 4.5 Rata-rata Akurasi Vidio 2 dengan ARM Nanopi M4V2

Pada gambar 4.5 menampilkan rata rata tingkat akurasi dari objek yang dideteksi dimasing masing frame, dapat dilihat bahwa tingkat akurasi yang didapatkan sangat tinggi dengan rata-rata diatas 0,5% yang artinya tingkat akurasi pada video sangatlah baik.

#### 4.2 Perbandingan Tingkat Akurasi Deteksi Objek Pada NanoPi M4V2

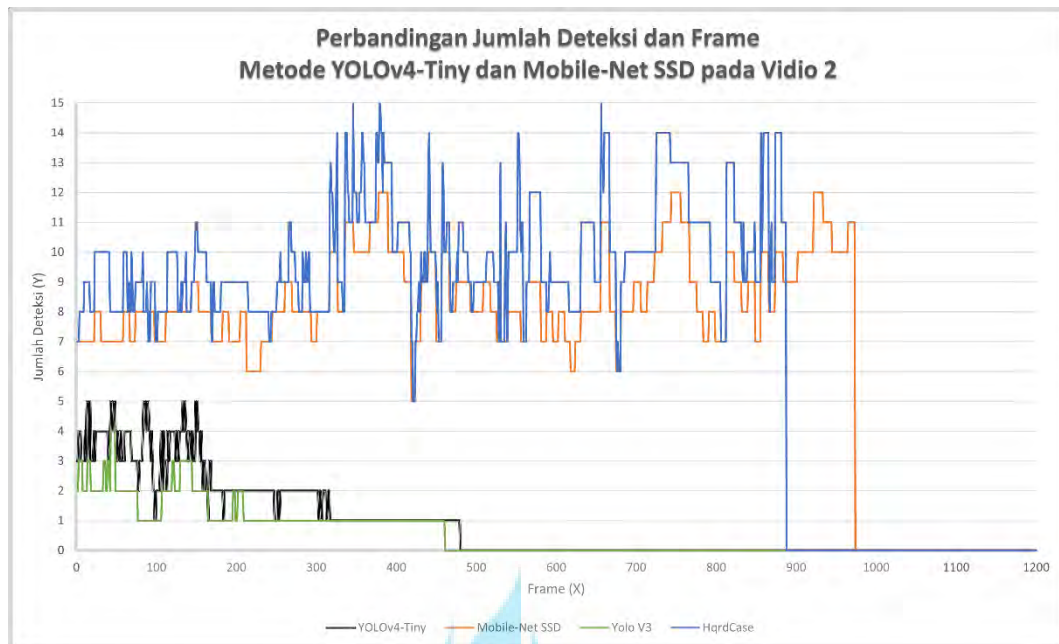
Dalam penelitian ini, penulis juga melakukan perbandingan dengan 4 metode pendeteksian objek yang berbeda diantaranya : YOLOv4-Tiny, YOLOv3, HardCase dan Mobile-Net SSD. Tujuan dari perbandingan ini adalah untuk mengetahui bagaimana kualitas dari ARM Prosesore pada 4 metode yang berbeda, dan untuk mengetahui apakah metode dari YOLOv4-Tiny yang penulis gunakan mendapatkan hasil yang lebih baik dari ketiga metode tersebut. Berikut curve dari hasil perbandingan.



Gambar 4.6 Kurva Perbandingan Pada Vidio 1

Pada gambar diatas terlihat perbedaan dari 4 metode yang digunakan dimana metode YOLOv4-Tiny mendapatkan hasil yang stabil dengan jumlah objek yang sesuai dengan objek pada vidio, sedangkan hasil pada Mobile-Net SSD didapatkan hasil dengan objek yang terdeteksi lebih banyak dari pada objek yang terdapat pada vidio, selain itu pada metode YOLOv3 didapatkan penurunan jumlah objek yang terdeteksi dari pada jumlah objek yang terdapat pada vidio dan yang terakhir dengan menggunakan metode HardCase didapatkan hasil terdapat banyak objek yang terdeteksi jika dibandingkan dengan ketiga metode. Sehingga didapatkan hasil bahwa YOLOv4-Tiny lebih baik jika dibandingkan dengan ketiga metode diatas. Akan tetapi jumlah FPS (jumlah frame perdetik) pada YOLOv4-Tiny lebih rendah daripada metode Mobile-Net SSD dan Hardcase, sedangkan dengan metode YOLOv3 FPS.





Gambar 4.7 Kurva Perbandingan Pada Vidio 2

Pada gambar diatas terlihat perbedaan dari 4 metode yang digunakan dimana metode YOLOV4-Tiny mendapatkan hasil yang stabil dengan jumlah objek yang sesuai dengan objek pada vidio, sedangkan hasil pada Mobile-Net SSD didapatkan hasil dengan objek yang terdeteksi lebih banyak dari pada objek yang terdapat pada vidio, selain itu pada metode YOLOv3 didapatkan penurunan jumlah objek yang terdeteksi dari pada jumlah objek yang terdapat pada vidio dan yang terakhir dengan menggunakan metode HardCase didapatkan hasil terdapat banyak objek ayng terdeteksi jika dibandingkan dengan ketiga metode. Sehingga didapatkan hasil bahwa YOLOv4 Tiny lebih baik jika dibandingkan dengan ketiga metode diatas. Akan tetapi jumlah FPS (jumlah frame perdetik) pada YOLOv4-Tiny lebih rendah daripada metode Mobile-Net SSD dan Hardcase, sedangkan dengan metode YOLOv3 FPS.

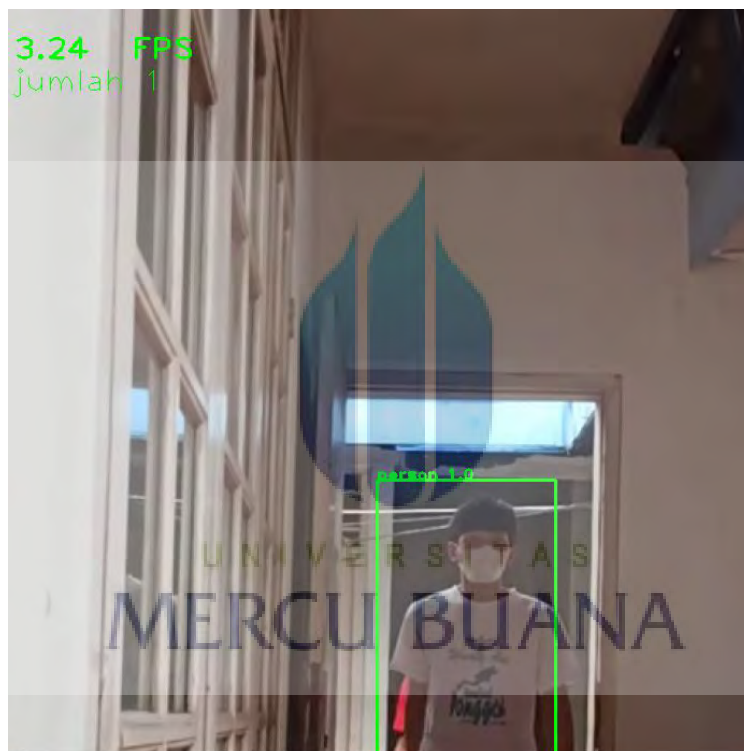
### 4.3 Pengujian Tingkat Akurasi Menggunakan Laptop

Pada penelitian ini, selain penulis menggunakan perangkat ARM Prosesor Nanopi M4V2, penulis juga menggunakan perangkat laptop untuk mengetahui perbandingan kinerja antar perangkat Nanopi M4V2 dan Laptop Acer Aspire Core i3.



#### 4.3.1 Hasil Pengujian Tingkat Akurasi

Pada saat melakukan pemrosesan, vidio rekaman yang berbentuk vidio, dimasukkan kedalam program yang telah dibuat, selanjutnya vidio akan diproses oleh program, hasil pemrosesan akan ditampilkan berupa nilai FPS dari vidio tersebut, tingkat akurasi dan nama objek yang terdeteksi. Berikut merupakan contoh dari pemrosesan vidio dengan masukan secara manual.



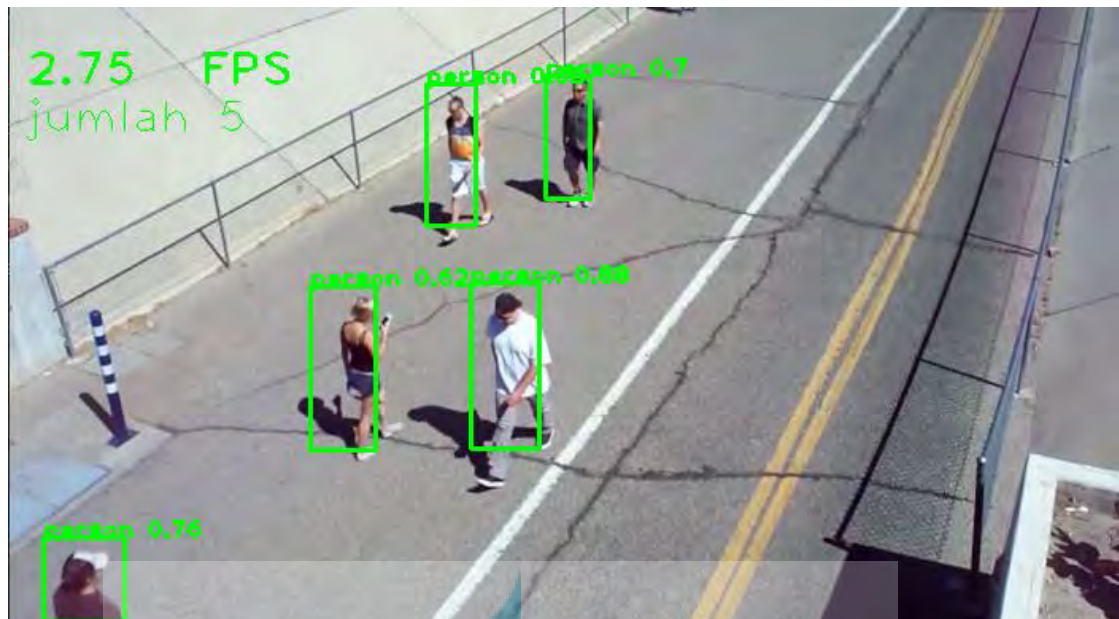
Gambar 4.8 Hasil 1 Objek Hadap Kamera Dengan Laptop

Pada hasil tangkapan layar diatas, dilakukan percobaan dengan objek membelakangi kamera dimana didapatkan hasil bahwa objek terdeteksi dengan dengan tingkat akurasi yang tinggi dan nilai dari FPS yang tidak terlalu rendah.



Gambar 4.9 Hasil 2 Objek Hadap Belakang Dengan Laptop

Pada tangkapan layar diatas, dilakukan percobaan dengan menggunakan 2 buah objek manusia yang sama sama membelakangi kamera, dengan hasil kedua objek tersebut terdeteksi oleh perangkat dan mendapatkan tingkat akurasi yang tinggi dengan nilai FPS yang menurun.



Gambar 4.10 Hasil Pada Objek Yang Banyak Dengan Laptop

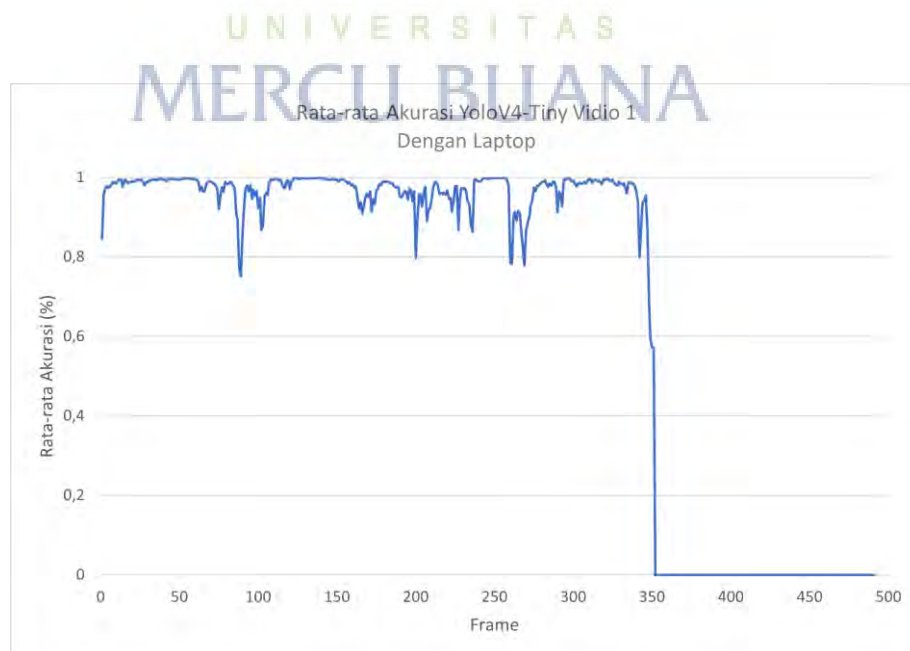
Pada tangkapan layar diatas, dilakukan percobaan dengan menggunakan objek manusia pada perangkat laptop dimana terdapat banyak objek yang berjalan di jalan raya dengan dan terdapat objek seperti tiang sebagai objek yang menyerupai manusia, dari percobaan tersebut didapatkan hasil objek manusia yang berjalan terdeteksi oleh perangkat dengan tingkat akurasi yang sedikit tinggi dan objek yang lain tidak terdeteksi sedangkan nilai FPS yang sedikit tinggi.

Dari ketiga gambar diatas didapatkan tingkat akurasi yang tinggi dan sedikit penurunan nilai FPS. Selain itu tingkat akurasi pada pendeteksian objek tidak terpengaruh dengan posisi objek, apakah membelakangi kamera atau menghadap kamera, dikarenakan nilai akurasi masih diatas 90%. Berikut tabel dari tangkapan layar diatas.

Tabel 4.2 Deteksi objek YOLOv4-Tiny pada perangkat Laptop

No	Objek	Akurasi (%)	Jumlah Terdeteksi	Jumlah Objek
1	1 Objek	1.0	1	1
2	2 Objek	- 1.0 - 0.97	2	2
3	Banyak Objek	- 0.76 - 0.62 - 0.88 - 0.88 - 0.70	5	5
FPS		2.70 – 3.55		

Dari tabel diatas dapat dilihat bahwa dengan menggunakan metode YOLOv4-Tiny kita dapat melakukan pemerosesan citra digital pada perangkat laptop dimana didapatkan hasil dalam pendeteksian objek khususnya objek manusia dengan tingkat akurasi yang tinggi dan tidak terpengaruh dengan objek yang lainnya.



Gambar 4.11 Rata-rata Akurasi Vidio 1 Dengan Laptop

Pada gambar 4.11 menampilkan rata rata tingkat akurasi dari objek yang dideteksi disetiap frame pada video 1, dari kurva tersebut dapat dilihat bahwa tingkat akurasi yang didapatkan sangat tinggi dengan rata-rata diatas 0,50% yang artinya tingkat akurasi pada video sangatlah baik.



Gambar 4.12 Rata-rata Akurasi Vidio 2 Dengan Laptop

Pada gambar 4.12 menampilkan rata rata tingkat akurasi dari objek yang dideteksi disetiap frame yang terdapat di video 2, dimana dari kurva tersebut dapat dilihat bahwa tingkat akurasi yang didapatkan sangat tinggi dengan rata-rata diatas 0,50% yang artinya tingkat akurasi pada video sangatlah baik.

#### 4.4 Performa CPU dan RAM

Pada proses ini penulis akan membahas bagaimana performa dari RAM (*Random Access Memory*) dan CPU (*Central Processore Unit*) ketika pengolahan citra digital dijalankan dan pada saat perangkat belum digunakan untuk melakukan pengolahan citra digital.

##### 4.4.1 Performa CPU dan ARM pada perangkat NanoPi M4V2

Pada proses ini penulis akan membahas bagaimana performa dari CPU dan RAM pada perangkat NanoPi M4V2 dimana akan menampilkan hasil performa pada saat



perangkat belum digunakan dan setelah digunakan untuk melakukan pemerosesan pengolahan citra digital.

```

1 [ ] 0.0% Hostname: nanopim4v2
2 [ ] 0.0% Tasks: 70, 96 thr; 1 running
3 [ ] 24.8% Load average: 0.30 0.17 0.07
4 [ ] 0.0% Uptime: 00:03:49
5 [ ] 0.0% CpuFreq1: 1.42 GHz ( .LITTLE)
6 [ ] 2.6% CpuFreq2: 1.42 GHz ( .LITTLE)
Mem [ ] 379M/3.72G CpuFreq3: 1.42 GHz ( .LITTLE)
Swp [ ] 0K/1024M CpuFreq4: 1.42 GHz ( .LITTLE)
Cpu Temp: 38 C CpuFreq5: 1.80 GHz (big. )
CpuFreq6: 1.80 GHz (big. )

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1047 root 20 0 677M 90928 41124 S 25.2 2.3 0:23.05 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /v
1456 nanopi 20 0 293M 64968 16880 S 3.9 1.7 0:03.12 xfdesktop
1905 nanopi 20 0 371M 36284 26508 S 3.2 0.9 0:01.73 xfce4-terminal
1199 root 20 0 677M 90928 41124 S 2.6 2.3 0:02.14 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /v
1448 nanopi 20 0 32124 19352 15740 S 1.9 0.5 0:01.20 xfwm4
1452 nanopi 20 0 214M 31608 21980 S 1.3 0.8 0:02.29 xfce4-panel
1918 nanopi 20 0 6076 3448 2600 R 1.3 0.1 0:01.68 htop
716 root 20 0 7680 4384 1300 R 0.6 0.1 0:01.14 /usr/sbin/haveged --Foreground --verbose=1 -w 10
783 root 20 0 393M 17488 14248 S 0.6 0.4 0:01.35 /usr/sbin/NetworkManager --no-daemon
1373 nanopi 20 0 6808 3080 2264 S 0.6 0.1 0:00.40 /usr/bin/dbus-daemon --syslog --fork --print-pid
819 root 20 0 1816 72 0 S 0.0 0.0 0:00.18 /usr/sbin/rngd -r /dev/urandom
1633 nanopi 20 0 155M 6252 5560 S 0.0 0.2 0:00.07 /usr/lib/at-spi2-core/at-spi2-registrard --use-gn
743 messagebu 20 0 7192 4228 3100 S 0.0 0.1 0:00.67 /usr/bin/dbus-daemon --system --address=systemd:
781 root 20 0 9684 6116 5492 S 0.0 0.2 0:00.07 /sbin/wpa_supplicant -u -s -O /run/wpa_supplican
1632 nanopi 20 0 953M 62108 46876 S 0.0 1.6 0:00.10 nm-applet
1461 nanopi 20 0 203M 16556 13928 S 0.0 0.4 0:00.21 xfsettingsd
1626 nanopi 20 0 6308 3140 2804 S 0.0 0.1 0:00.04 /usr/bin/dbus-daemon --config-file=/usr/share/de
796 root 20 0 10404 5252 4536 S 0.0 0.1 0:00.11 /lib/systemd/systemd-logind
1 root 20 0 157M 7928 5792 S 0.0 0.2 0:04.07 /sbin/init

```

Gambar 4.13 Pada Saat Program Tidak Dijalankan

```

1 [ ] 65.8% Hostname: nanopim4v2
2 [ ] 42.1% Tasks: 93, 222 thr; 6 running
3 [ ] 50.9% Load average: 1.90 2.01 1.90
4 [ ] 51.6% Uptime: 00:55:46
5 [ ] 74.2% CpuFreq1: 1.42 GHz ( .LITTLE)
6 [ ] 74.8% CpuFreq2: 1.42 GHz ( .LITTLE)
Mem [ ] 1.02G/3.72G CpuFreq3: 1.42 GHz ( .LITTLE)
Swp [ ] 0K/1024M CpuFreq4: 1.42 GHz ( .LITTLE)
Cpu Temp: 62 C CpuFreq5: 1.80 GHz (big. )
CpuFreq6: 1.80 GHz (big. )

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
10807 nanopi 20 0 1329M 165M 56344 R 345. 4.3 0:45.22 python3.8 Object_detection.py
10825 nanopi 20 0 1329M 165M 56344 R 62.2 4.3 0:07.35 python3.8 Object_detection.py
10824 nanopi 20 0 1329M 165M 56344 R 59.6 4.3 0:07.00 python3.8 Object_detection.py
10826 nanopi 20 0 1329M 165M 56344 S 58.3 4.3 0:07.00 python3.8 Object_detection.py
10823 nanopi 20 0 1329M 165M 56344 R 47.8 4.3 0:06.65 python3.8 Object_detection.py
10822 nanopi 20 0 1329M 165M 56344 S 45.9 4.3 0:06.12 python3.8 Object_detection.py
1010 root 20 0 719M 131M 43868 S 7.2 3.4 4:30.81 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/l
2004 nanopi 20 0 6152 3588 2648 R 3.3 0.1 1:27.05 htop
1863 nanopi 20 0 373M 38300 26956 S 0.7 1.0 0:32.54 xfce4-terminal
1470 nanopi 20 0 235M 53176 22124 S 0.7 1.4 0:09.64 xfce4-panel
1466 nanopi 20 0 55776 42636 15288 S 0.7 1.1 0:21.11 xfwm4
10819 nanopi 20 0 1329M 165M 56344 S 0.7 4.3 0:00.03 python3.8 Object_detection.py
10816 nanopi 20 0 1329M 165M 56344 S 0.7 4.3 0:00.05 python3.8 Object_detection.py

```

Gambar 4.14 Pada Saat Program Dijalankan

Dari kedua gambar diatas, dapat dilihat bahwa penggunaan CPU pada saat program tidak dijalankan lebih sedikit jika dibandingkan ketika program dijalankan.

Dibawah ini adalah tabel perbandingan dari kedua gambar diatas.



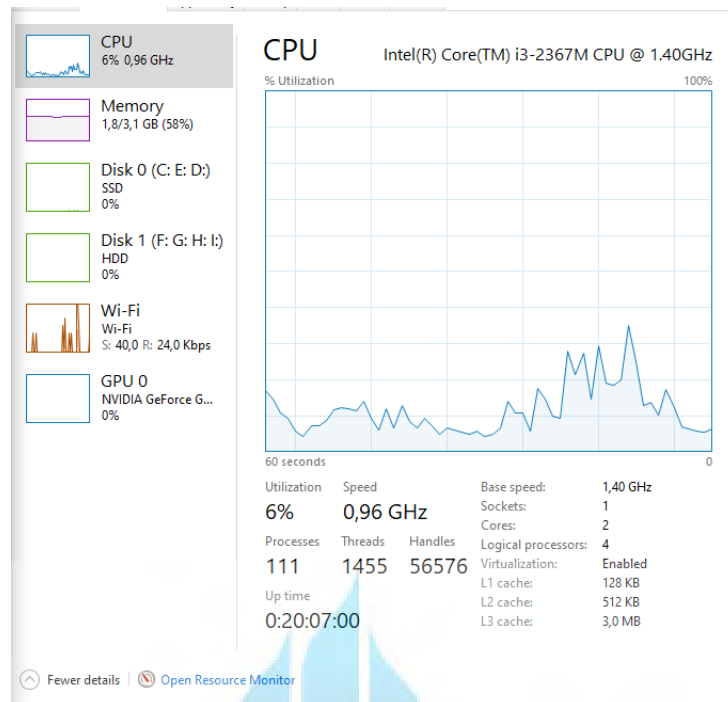
Tabel 4.3 Performa Cpu Arm Pada Deteksi Objek Dengan Metode

Nama	CPU Frequency	Tanpa program	Program dijalankan
Little Core	1.42 GHz	5.3 %	65.8 %
	1.42 GHz	0 %	42.1 %
	1.42 GHz	4 %	59.9 %
	1.42 GHz	0 %	51.6 %
Big Core	1.80 GHz	0 %	75.2 %
	1.80 GHz	2.6 %	74.8 %
RAM	4 GB	10 %	25 %
Suhu CPU		38 °C	62 °C

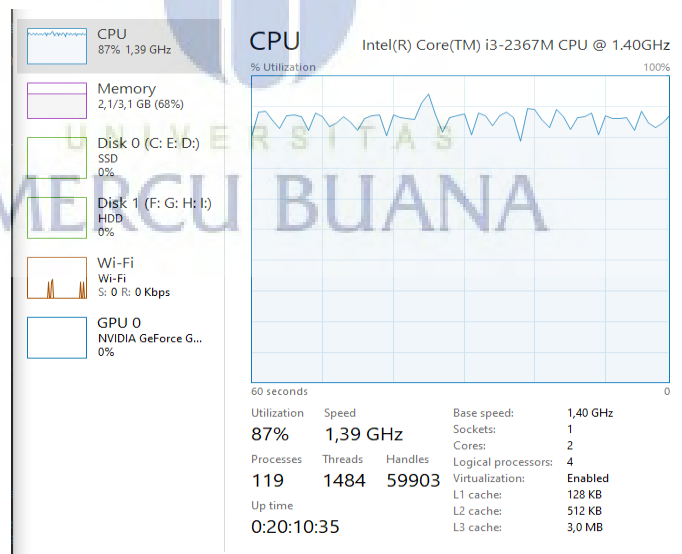
Dari data yang didapatkan penggunaan CPU pada perangkat tidak terlalu tinggi atau tidak sampai 100% dan RAM yang terpakai hanya 25% dari 100% atau 1,02Gb dari 4Gb RAM yang tersedia selain itu suhu pada perangkat juga tidak mengalami kepanasan yang terlalu berlebihan. Sehingga secara keseluruhan program dan perangkat berjalan dengan baik dan tidak terdapat masalah dalam penggunaannya.

#### 4.4.2 Performa CPU dan RAM pada perangkat Laptop

Pada proses ini penulis akan membahas bagaimana performa dari CPU dan RAM pada perangkat laptop yang digunakan untuk melakukan pengolahan citra digital, dimana pada pembahasan ini akan menampilkan hasil performa pada saat perangkat belum digunakan dan setelah digunakan untuk melakukan pemrosesan pengolahan citra digital.



Gambar 4.15 Pada Saat Program Tidak Dijalankan



Gambar 4.16 Pada Saat Program Dijalankan

Dari kedua gambar diatas, dapat dilihat bahwa penggunaan CPU pada saat program tidak dijalankan lebih sedikit jika dibandingkan ketika program dijalankan.

Dibawah ini adalah tabel perbandingan dari kedua gambar diatas.

Tabel 4.4 Performa CPU pada deteksi objek dengan metode YOLOv4 Tiny

Nama	Tanpa program	Program dijalankan
CPU	6 %	87 %
RAM	58 %	68 %

Dari data yang didapatkan penggunaan CPU pada perangkat terlihat sedikit lebih berat pada saat program dijalankan, dimana CPU bekerja lebih dari 80% begitu juga dengan penggunaan RAM yang terpakai melebihi 50%. Sehingga dapat disimpulkan bahwa secara keseluruhan program dan perangkat berjalan dengan baik akan tetapi penggunaan CPU dan RAM pada perangkat yang digunakan membutuhkan kinerja yang tinggi.

