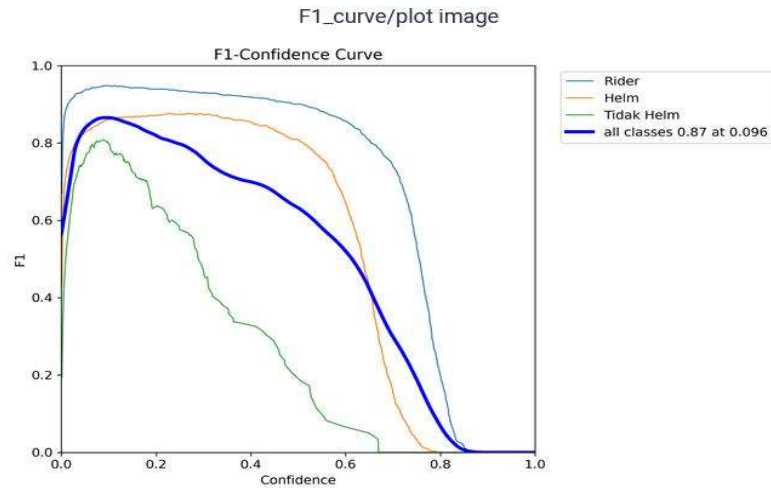


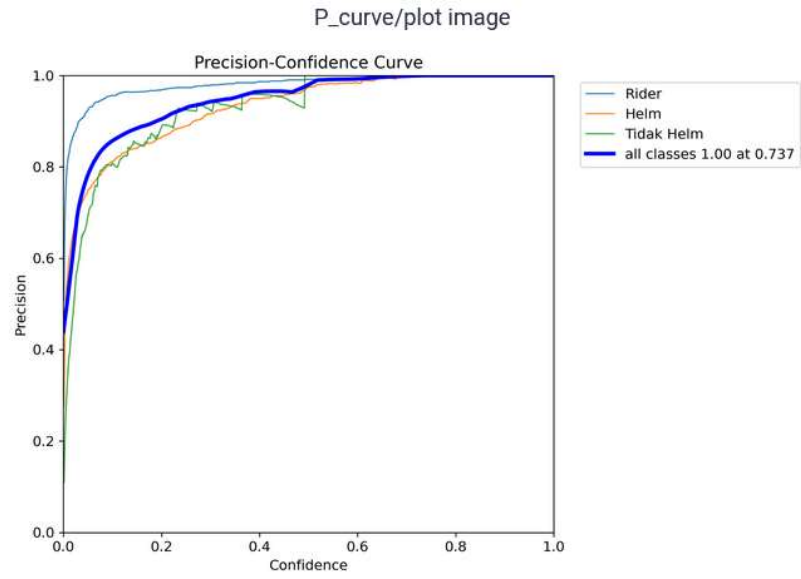
## LAMPIRAN A *OUTPUT* GRAFIK MODEL

2023-12-05 19:00:17	all	240	1880	0.779	0.614	0.701	0.382
2023-12-05 19:00:18	50 epochs completed in 3.202 hours.						
2023-12-05 19:00:19	Optimizer stripped from runs/detect/train2/weights/last.pt, 6.2MB						
	Optimizer stripped from runs/detect/train2/weights/best.pt, 6.2MB						
	Validating runs/detect/train2/weights/best.pt...						
	Ultralytics YOLOv8.0.222 Python-3.10.12 torch-2.1.0+cu118 CUDA:0 (Tesla T4, 15102MiB)						
	Model summary (fused): 168 layers, 3006233 parameters, 0 gradients, 8.1 GFLOPs						
2023-12-05 19:00:25	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 38%
2023-12-05 19:00:31	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
2023-12-05 19:00:35	all	240	1880	0.859	0.879	0.904	0.505
	Rider	240	864	0.956	0.938	0.977	0.68
	Helm	240	899	0.813	0.915	0.927	0.462
	Tidak Helm	240	117	0.806	0.783	0.807	0.373
2023-12-05 19:00:36	Speed: 0.2ms preprocess, 2.5ms inference, 0.0ms loss, 4.6ms postprocess per image						
	Results saved to runs/detect/train2						
2023-12-05 19:00:57	2023-12-05 12:00:57,248 - clearml.Task - INFO - Completed model upload to https://files.clear.ml/YOLOv8/train2.6392ced696d94e8882						

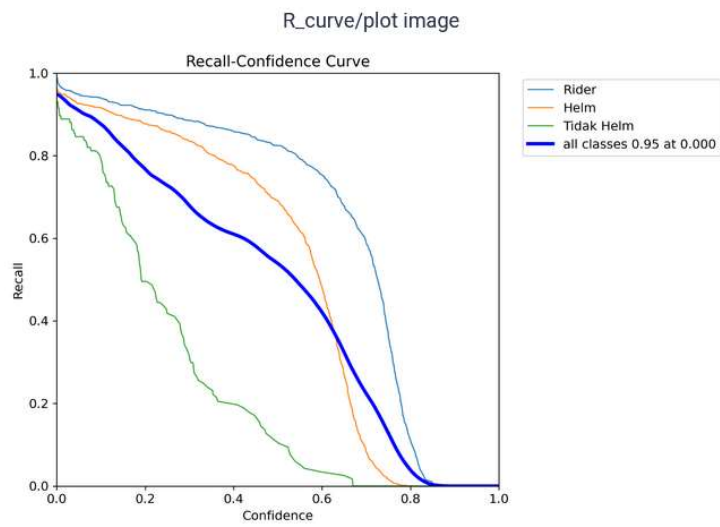
Gambar A.1 Output Model Terbaik pada ClearML



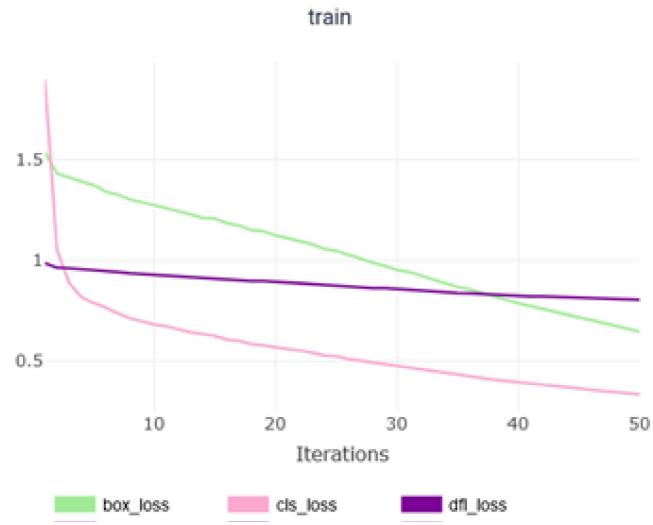
Gambar A.2 Kurva Kepercayaan  $F_1$ -score Model Training



Gambar A.3 Kurva Kepercayaan *Precision* Model Training



Gambar A.4 Kurva Kepercayaan *Recall* Model Training



Gambar A.5 Kurva Kerugian Model *Training*

## LAMPIRAN B SAMPEL DATA



Gambar B.1 Sampel *Frame* untuk Video Keperluan Dataset *Training* 1



Gambar B.2 Sampel *Frame* untuk Video Keperluan Dataset *Training* 2



Gambar B.3 Sampel *Frame* untuk Video Keperluan Dataset *Training 3*



Gambar B.4 Sampel *Frame* untuk Video Keperluan Dataset *Training 4*



Gambar B.5 Sampel *Frame* untuk Video Keperluan Dataset *Training 5*



Gambar B.6 Sampel *Frame* untuk Video Keperluan Dataset *Training* 6



Gambar B.7 Sampel *Frame* untuk Video Keperluan Dataset *Training* 7



Gambar B.8 Sampel *Frame* untuk Video Keperluan Dataset *Validasi*



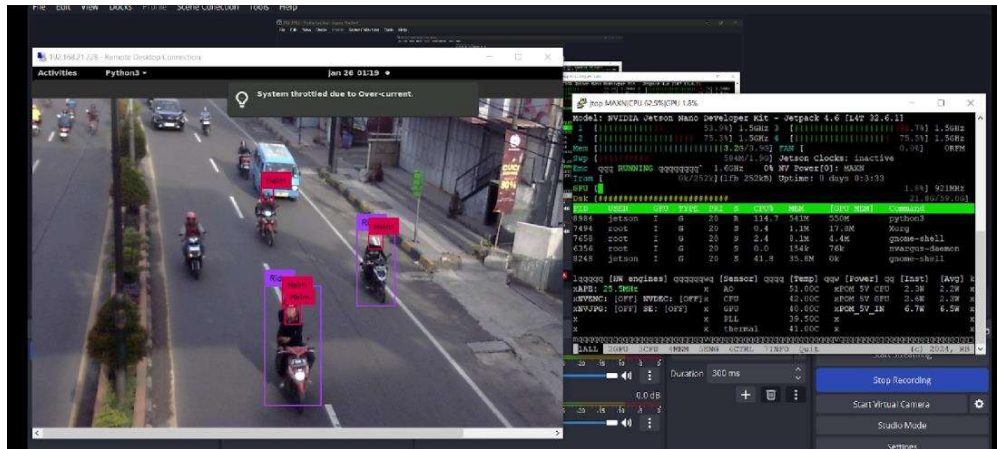
Gambar B.9 Sampel *Frame* untuk Video Keperluan *Testing Sore*



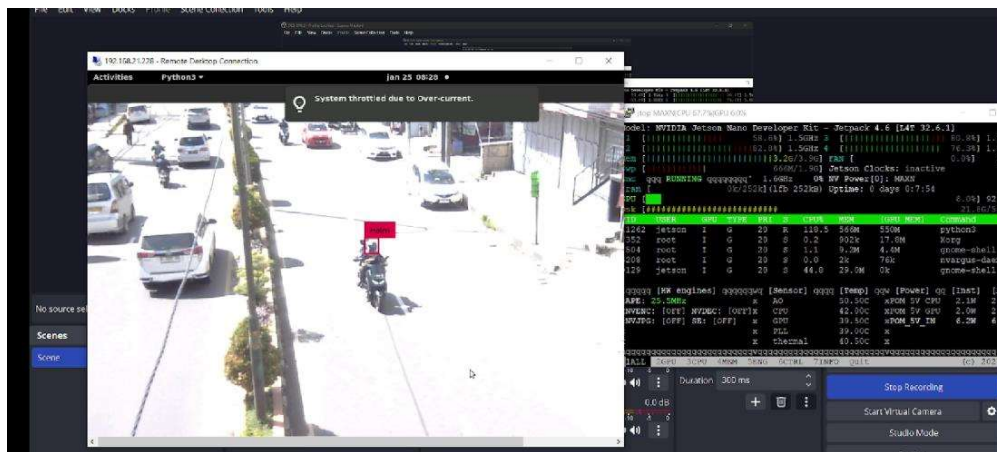
Gambar B.10 Sampel *Frame* untuk Video Keperluan *Testing Siang*



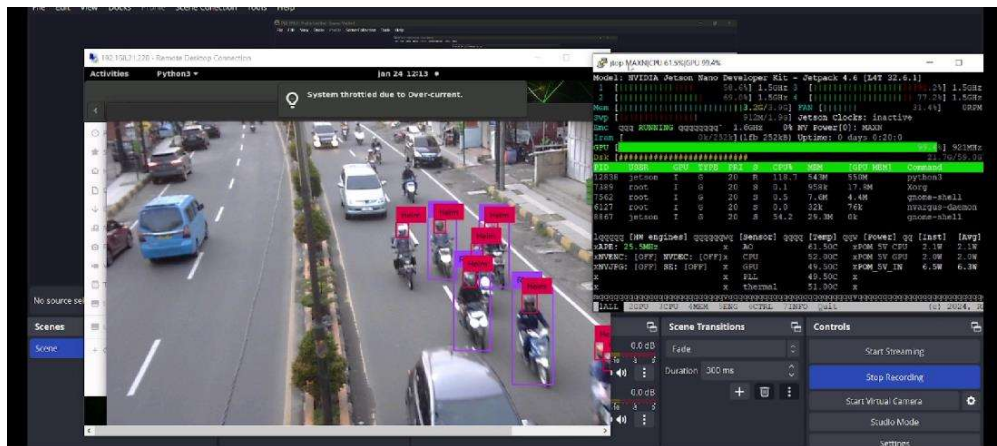
Gambar B.11 Sampel *Frame* untuk Video Keperluan *Testing* Pagi



Gambar B.12 Sampel Tampilan Penujian *Live Feed* Pagi



Gambar B.13 Sampel Tampilan Penujian *Live Feed* Siang





Gambar B.14 Sampel Tampilan Penujian *Live Feed* Sore



Gambar B.15 Posisi Kamera dan Penempatan Sistem

## LAMPIRAN C KONFIGURASI PENGUJIAN

1. Melakukan *remote access* antara *laptop* dan Jetson Nano
2. Memindahkan *folder* Sistem Deteksi Helm YOLOv8 bernama ‘Yolov8-Helmet-Detection’ ke OS Linux Jetson Nano melalui *remote access* yang diakses menggunakan SSHFS dan PuTTY.
3. Menjalankan Terminal pada GUI OS Linux pada NVIDIA Jetson Nano.
4. Lalu install model YOLOv8 satu paket dengan *library* yang dibutuhkan pada Terminal OS Linux Jetson Nano dengan menggunakan perintah berikut.

```
pip install ultralytics
```

5. Lalu install model *tracking ByteTrack* pada Terminal OS Linux Jetson Nano dengan menggunakan perintah berikut.

```
pip install supervision
```

6. Menyesuaikan *working directory* dengan *folder* sistem yang telah dipindahkan dengan menggunakan perintah berikut pada terminal.

```
cd Yolov8-Helmet-Detection
```

7. Setelah semua pengaturan selesai, lalu menjalankan file *testbtlive.py* untuk menjalankan *live feed* dengan perintah berikut.

```
python3 testbtlive.py
```