

**TUGAS AKHIR  
BENGKEL CODING  
“DETEKSI API KEBARAN HUTAN”**

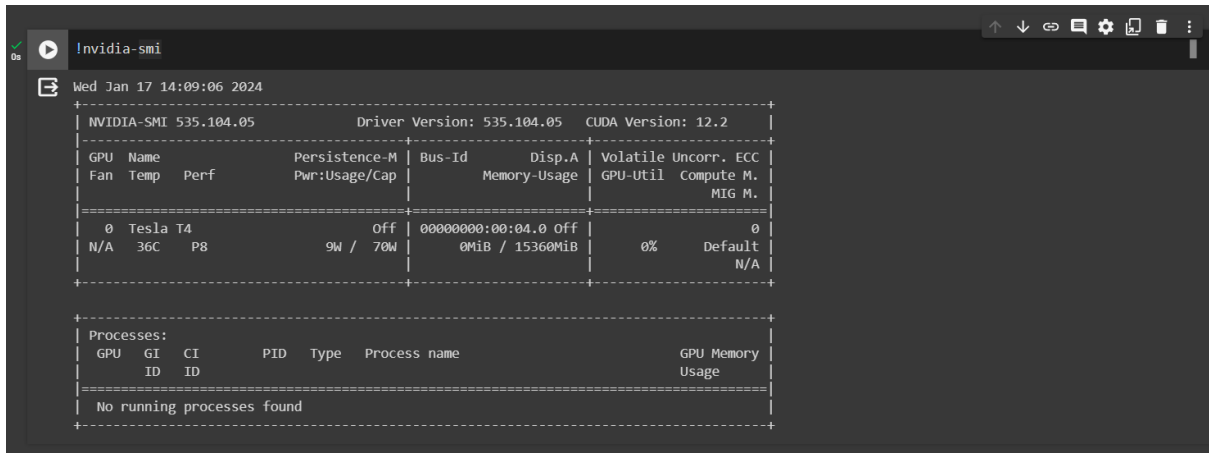


Disusun Oleh:

Nama	: Muhammad Ghozali
NIM	: A11.2020.12475
Program Studi	: Teknik Informatika

**TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DIAN NUSWANTORO  
SEMARANG  
2024**

Link Github :



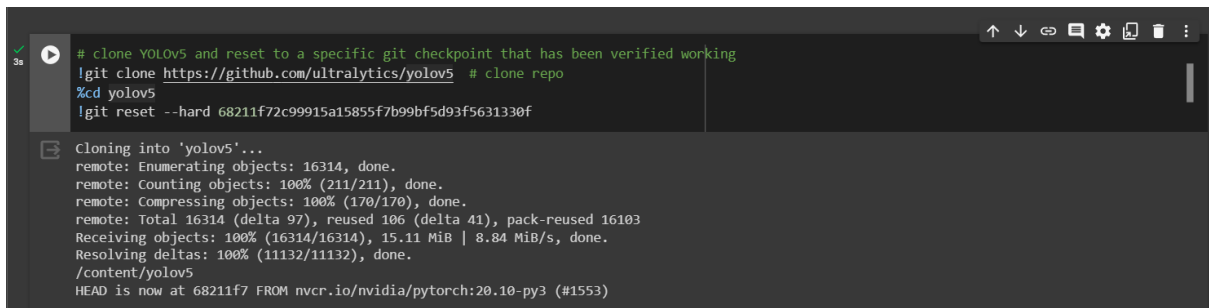
```
!nvidia-smi
Wed Jan 17 14:09:06 2024
+-----+
| NVIDIA-SMI 535.104.05                  Driver Version: 535.104.05   CUDA Version: 12.2     |
+-----+-----+
| GPU  Name Persistence-M   Bus-Id        Disp.A    Volatile Uncorr. ECC  |
| Fan  Temp  Perf    Pwr:Usage/Cap     Memory-Usage  GPU-Util  Compute M.  |
|=====-=+==+=====+=====+=====+=====+=====+=====+|
| 0   Tesla T4      Off          00000000:00:04:0  Off              0%          Default  |
| N/A   36C    P8              9W / 70W           0MiB / 15360MiB             0%          N/A  |
+-----+-----+
+-----+
| Processes:                               GPU Memory               |
|  GPU   GI   CI        PID   Type   Process name                  Usage              |
|=====-=+==+=====+=====+=====+=====+=====+=====+|
| No running processes found               |
+-----+-----+

```

1.

Penggunaan kode `!nvidia-smi`, pada dasarnya menjalankan perintah `nvidia-smi` di dalam shell atau terminal. Ini akan menampilkan informasi tentang GPU NVIDIA yang terpasang pada sistem, seperti penggunaan GPU, suhu, pemakaian memori, dan informasi lainnya yang berkaitan dengan performa dan status GPU.

2. Instalasi YOLO v5



```
# clone YOLOv5 and reset to a specific git checkpoint that has been verified working
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
!git reset --hard 68211f72c99915a15855f7b99bf5d93f5631330f

Cloning into 'yolov5'...
remote: Enumerating objects: 16314, done.
remote: Counting objects: 100% (211/211), done.
remote: Compressing objects: 100% (170/170), done.
remote: Total 16314 (delta 97), reused 106 (delta 41), pack-reused 16103
Receiving objects: 100% (16314/16314), 15.11 MiB | 8.84 MiB/s, done.
Resolving deltas: 100% (11132/11132), done.
/content/yolov5
HEAD is now at 68211f7 FROM nvcr.io/nvidia/pytorch:20.10-py3 (#1553)

```

Berikut adalah penjelasan singkat kegunaan setiap perintah:

- a. `git clone https://github.com/ultralytics/yolov5`: Melakukan cloning repositori YOLOv5 dari GitHub. Ini menciptakan salinan lokal dari seluruh repositori di system.
- b. `%cd yolov5`: Memindahkan direktori kerja (current directory) ke dalam folder `yolov5`, yang merupakan hasil dari proses cloning sebelumnya. Tanda persen (%) adalah cara untuk menjalankan perintah shell di dalam lingkungan notebook seperti Jupyter.
- c. `git reset --hard 68211f72c99915a15855f7b99bf5d93f5631330f`: Melakukan reset ke commit tertentu pada repositori. Dalam hal ini, commit yang ditentukan adalah

68211f72c99915a15855f7b99bf5d93f5631330f. Ini bermanfaat untuk kembali ke versi tertentu dari kode yang telah diverifikasi berfungsi dengan baik.

### 3. Tambahkan Dataset

```
# Tambahkan file dataset
%cd /content
!curl -L "https://app.roboflow.com/ds/KRktagHISh?key=Se13tojdab" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip

/content
% Total      % Received % Xferd Average Speed   Time    Time     Time  Current
             Dload  Upload   Total   Spent    Left   Speed
100  910    100  910    0     0   1196      0  --:--:-- --:--:-- --:--:--   1195
100 9190k  100 9190k    0     0  9433k      0  --:--:-- --:--:-- --:--:--   9433k
Archive: roboflow.zip
replace README.roboflow.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
extracting: README.roboflow.txt
extracting: data.yaml
extracting: test/images/012_rgb_png.rf.5e756616412192becbf552ff01926d53.jpg
extracting: test/images/032_rgb_png.rf.7dfc8d7aa944a16fa200fb35ac8cceb.a.jpg
extracting: test/images/070_rgb_png.rf.1b73078f3cf20321964d11a33a14ecd1.jpg
extracting: test/images/128_rgb_png.rf.1e52beaaa4bc489e74d3ecf4aa3f44c5.jpg
extracting: test/images/469_png.rf.5777ce745bdc412f0c685637e5bd78aa.jpg
extracting: test/images/coast_n708024_png.rf.24fa05715a91c9b4e5ebd52a858ab230.jpg
extracting: test/images/coast_nat910_png.rf.8e3d56deb4ccd1f67d42f96e385000a5.jpg
extracting: test/labels/012_rgb_png.rf.5e756616412192becbf552ff01926d53.txt
extracting: test/labels/032_rgb_png.rf.7dfc8d7aa944a16fa200fb35ac8cceb.a.txt
extracting: test/labels/070_rgb_png.rf.1b73078f3cf20321964d11a33a14ecd1.txt
extracting: test/labels/128_rgb_png.rf.1e52beaaa4bc489e74d3ecf4aa3f44c5.txt
extracting: test/labels/469_png.rf.5777ce745bdc412f0c685637e5bd78aa.txt
extracting: test/labels/coast_n708024_png.rf.24fa05715a91c9b4e5ebd52a858ab230.txt
extracting: test/labels/coast_nat910_png.rf.8e3d56deb4ccd1f67d42f96e385000a5.txt
```

### 4. Jumlah Kelas

```
[ ] # Jumlah Kelas dalam dataset
import yaml
with open("data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

### 5. Model konfigurasi algoritma yolo

```
#model konfigurasi algoritma yolo
%cat /content/yolov5/models/yolov5s.yaml

# parameters
nc: 80 # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]], # 0-P1/2
  [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
  [-1, 3, BottleneckCSP, [128]],
  [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
  [-1, 9, BottleneckCSP, [256]],
  [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
  [-1, 9, BottleneckCSP, [512]],
  [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
  [-1, 1, SPP, [1024, [5, 9, 13]]],
  [-1, 3, BottleneckCSP, [1024, False]], # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]]]
```

## 6. Train data YOLOv5

```

Train YOLO v5

# train yolov5s on custom data |
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 5 --data '../data.yaml' --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results

/content/yolov5
Using torch 1.7.0+cu101 CUDA:0 (Tesla T4, 15079MB)

Namespace(adam=False, batch_size=16, bucket='', cache_images=True, cfg='./models/custom_yolov5s.yaml', data='../data.yaml', device='', epochs=5,
Start Tensorboard with "tensorboard --logdir runs/train", view at http://localhost:6006/
2020-12-28 19:53:41.762884: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library libcudart.so.10
Hyperparameters {'lr0': 0.01, 'lrf': 0.2, 'momentum': 0.937, 'weight_decay': 0.0005, 'warmup_epochs': 3.0, 'warmup_momentum': 0.8, 'warmup_bias_lr': 0.01}

from n  params module arguments
0 -1 1 3520 models.common.Focus [3, 32, 3]
1 -1 1 18560 models.common.Conv [32, 64, 3, 2]
2 -1 1 19904 models.common.BottleneckCSP [64, 64, 1]
3 -1 1 73984 models.common.Conv [64, 128, 3, 2]
4 -1 1 161152 models.common.BottleneckCSP [128, 128, 3]
5 -1 1 295424 models.common.Conv [128, 256, 3, 2]
6 -1 1 641792 models.common.BottleneckCSP [256, 256, 3]
7 -1 1 1180672 models.common.Conv [256, 512, 3, 2]
8 -1 1 656896 models.common.SPP [512, 512, [5, 9, 13]]
9 -1 1 1248768 models.common.BottleneckCSP [512, 512, 1, False]
10 -1 1 131584 models.common.Conv [512, 256, 1, 1]
11 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
12 [-1, 6] 1 0 models.common.Concat [1]
13 -1 1 378624 models.common.BottleneckCSP [512, 256, 1, False]

Using 2 dataloader workers
Logging results to runs/train/yolov5s_results
Starting training for 5 epochs...

Epoch 0/4  gpu_mem  box  obj  cls  total  targets  img_size
          1.85G  0.1088  0.03796  0  0.1468  17  416: 100% 9/9 [00:03<00:00, 2.89it/s]
          Class  Images  Targets  P  R  mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00, 1.98it/s]
          all  14  10  0  0  0.00199  0.000366

Epoch 1/4  gpu_mem  box  obj  cls  total  targets  img_size
          1.85G  0.1048  0.03877  0  0.1435  21  416: 100% 9/9 [00:01<00:00, 6.22it/s]
          Class  Images  Targets  P  R  mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00, 7.28it/s]
          all  14  10  0  0  0.0044  0.00131

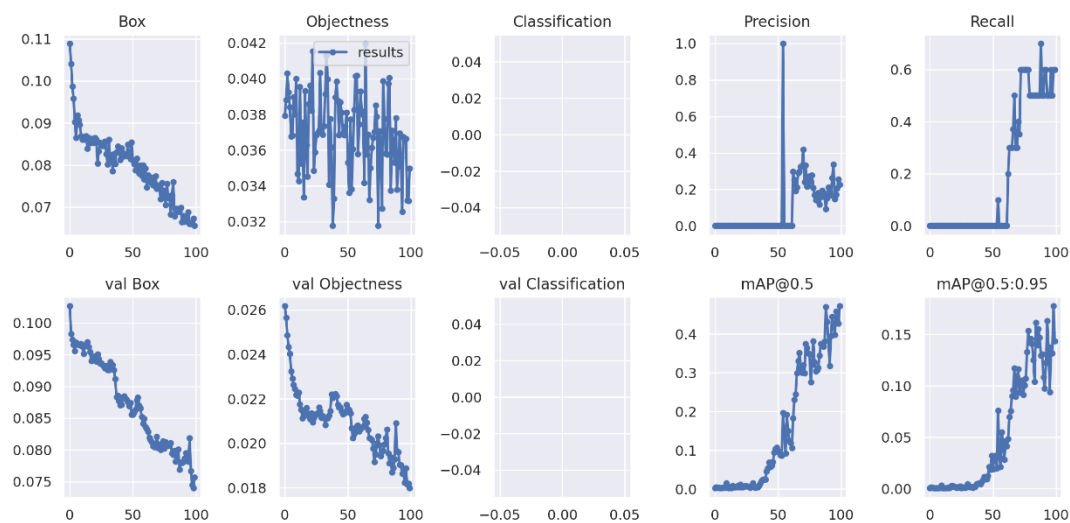
Epoch 2/4  gpu_mem  box  obj  cls  total  targets  img_size
          1.85G  0.1003  0.03979  0  0.1401  25  416: 100% 9/9 [00:01<00:00, 6.10it/s]
          Class  Images  Targets  P  R  mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00, 9.53it/s]
          all  14  10  0  0  0.0043  0.000838

Epoch 3/4  gpu_mem  box  obj  cls  total  targets  img_size
          1.85G  0.09801  0.03852  0  0.1365  14  416: 100% 9/9 [00:01<00:00, 6.05it/s]
          Class  Images  Targets  P  R  mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00, 11.32it/s]
          all  14  10  0  0  0.00323  0.000806

Epoch 4/4  gpu_mem  box  obj  cls  total  targets  img_size
          1.85G  0.09217  0.03747  0  0.1296  20  416: 100% 9/9 [00:01<00:00, 6.15it/s]
          Class  Images  Targets  P  R  mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00, 2.23it/s]
          all  14  10  0  0  0.00407  0.00107

Optimizer stripped from runs/train/yolov5s_results/weights/last.pt, 14.7MB
Optimizer stripped from runs/train/yolov5s_results/weights/best.pt, 14.7MB
5 epochs completed in 0.004 hours.

```



7.

8. Hasil

