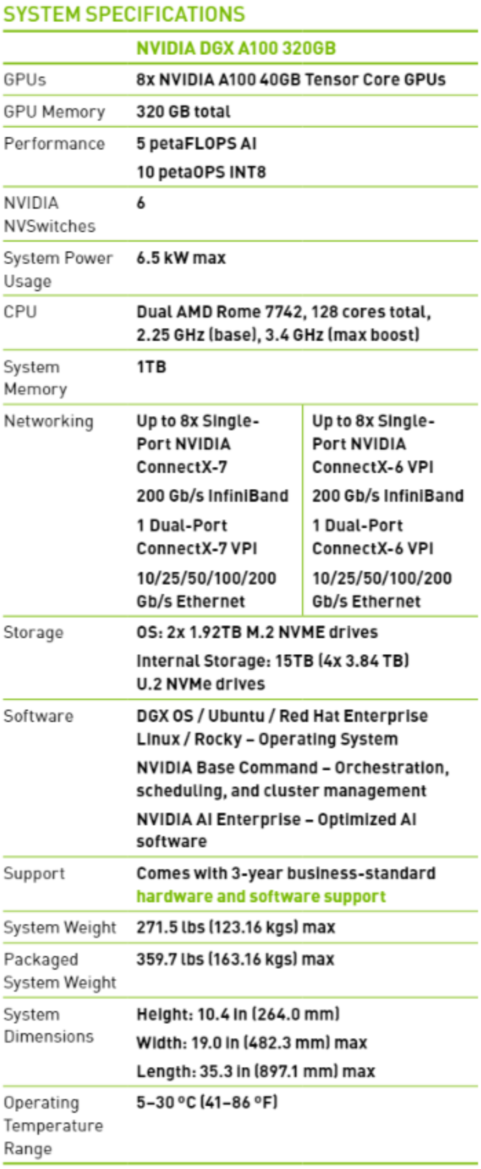
DATASHEET

NVIDIA DGX A100

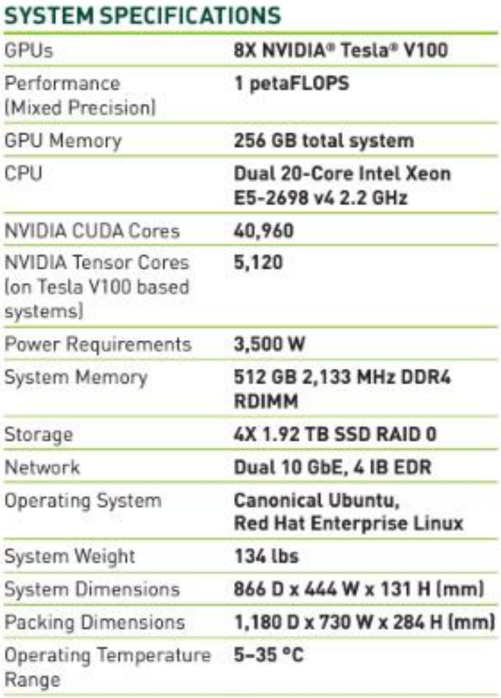
The universal system for AI infrastructure.

NVIDIA DGX-1
THE ESSENTIAL
INSTRUMENT FOR
AI RESEARCH



## SYSTEM SPECIFICATIONS

| | NVIDIA DGX A100 320GB | |
|---|---|---|
| GPUs | 8x NVIDIA A100 40GB Tensor Core GPUs | |
| GPU Memory | 320 GB total | |
| Performance | 5 petaFLOPS AI<br>10 petaOPS INT8 | |
| NVIDIA NVSwitches | 6 | |
| System Power Usage | 6.5 kW max | |
| CPU | Dual AMD Rome 7742, 128 cores total, 2.25 GHz (base), 3.4 GHz (max boost) | |
| System Memory | 1TB | |
| Networking | Up to 8x Single-Port NVIDIA ConnectX-7<br>200 Gb/s InfiniBand<br>1 Dual-Port ConnectX-7 VPI<br>10/25/50/100/200 Gb/s Ethernet | Up to 8x Single-Port NVIDIA ConnectX-6 VPI<br>200 Gb/s InfiniBand<br>1 Dual-Port ConnectX-6 VPI<br>10/25/50/100/200 Gb/s Ethernet |
| Storage | OS: 2x 1.92TB M.2 NVME drives<br>Internal Storage: 15TB (4x 3.84 TB) U.2 NVMe drives | |
| Software | DGX OS / Ubuntu / Red Hat Enterprise Linux / Rocky – Operating System<br>NVIDIA Base Command – Orchestration, scheduling, and cluster management<br>NVIDIA AI Enterprise – Optimized AI software | |
| Support | Comes with 3-year business-standard hardware and software support | |
| System Weight | 271.5 lbs (123.16 kgs) max | |
| Packaged System Weight | 359.7 lbs (163.16 kgs) max | |
| System Dimensions | Height: 10.4 in (264.0 mm)<br>Width: 19.0 in (482.3 mm) max<br>Length: 35.3 in (897.1 mm) max | |
| Operating Temperature Range | 5–30 ºC (41–86 ºF) | |

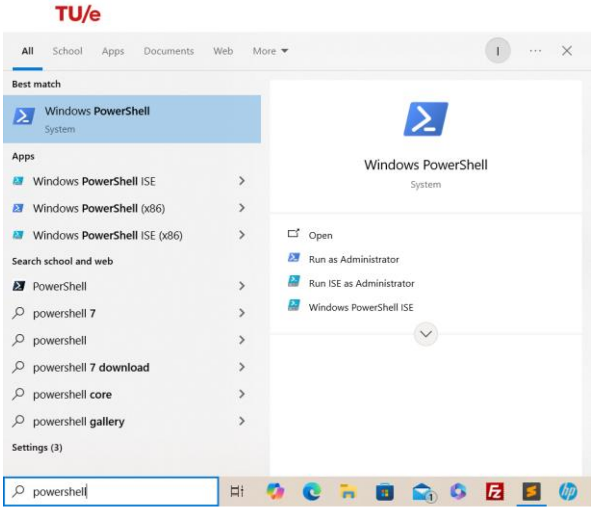


## SYSTEM SPECIFICATIONS

| | |
|---|---|
| GPUs | 8X NVIDIA® Tesla® V100 |
| Performance (Mixed Precision) | 1 petaFLOPS |
| GPU Memory | 256 GB total system |
| CPU | Dual 20-Core Intel Xeon E5-2698 v4 2.2 GHz |
| NVIDIA CUDA Cores | 40,960 |
| NVIDIA Tensor Cores (on Tesla V100 based systems) | 5,120 |
| Power Requirements | 3,500 W |
| System Memory | 512 GB 2,133 MHz DDR4 RDIMM |
| Storage | 4X 1.92 TB SSD RAID 0 |
| Network | Dual 10 GbE, 4 IB EDR |
| Operating System | Canonical Ubuntu, Red Hat Enterprise Linux |
| System Weight | 134 lbs |
| System Dimensions | 866 D x 444 W x 131 H (mm) |
| Packing Dimensions | 1,180 D x 730 W x 284 H (mm) |
| Operating Temperature Range | 5–35 °C |

## 1. Accessing server

**Pengguna windows**: bisa dengan menggunakan **Windows Powershell.** Harap koneksi via VPN-BRIN terlebih dahulu apabila mengakses server dari jaringan eksternal BRIN.

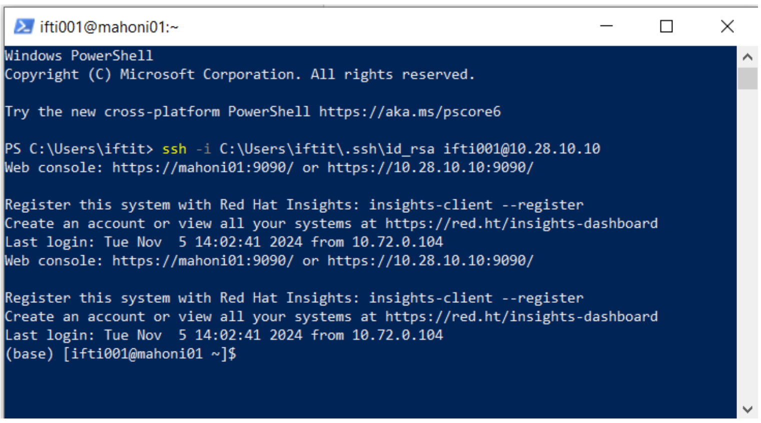


`<id_rsa location path>` lokasi di mana id_rsa disimpan di direktori lokal. Misalnya: `C:\Users\<windows_username>\.ssh\id_rsa`

`<userid>` user intra BRIN (untuk user pengguna internal BRIN). Misalnya: `ifti001`

```
ssh -i <id_rsa location path>  <userid>@10.28.10.10
```

Tampilan terminal setelah berhasil login ssh ke server.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\iftit> ssh -i C:\Users\iftit\.ssh\id_rsa ifti001@10.28.10.10
Web console: https://mahoni01:9090/ or https://10.28.10.10:9090/

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Nov  5 14:02:41 2024 from 10.72.0.104
Web console: https://mahoni01:9090/ or https://10.28.10.10:9090/

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Nov  5 14:02:41 2024 from 10.72.0.104
(base) [ifti001@mahoni01 ~]$
```

**Pengguna linux**: bisa dengan Linux terminal.

## 2. Installation: Python libraries via anaconda and pip

Dokumentasi ini hanya mencakup cara instalasi library Python untuk riset di Machine Learning/Artificial Intelligence/Natural Language Processing. Apabila ada perbedaan cara maupun tipe modul library yang diinstal, harap mencari referensi tambahan yang tidak dibahas pada source repository terkait (misalnya, github).

Instalasi python libraries dilakukan setelah berhasil **login ssh di mahoni01**.

**Catatan**: Contoh penginstalan di sini adalah apabila user pengguna ingin menginstall library utama tanpa dependen ke repository github tertentu. Sebaliknya, harap menggunakan referensi dari github atau repository acuan.

- Instalasi miniconda3 untuk penginstalan via conda

```
mkdir -p ~/miniconda3

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
-O ~/miniconda3/miniconda.sh

bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm ~/miniconda3/miniconda.sh
source ~/miniconda3/bin/activate
conda init
```

- Instalasi library via conda environment, referensi pytorch https://pytorch.org/get-started/locally/

```
conda create --name my_env python=3.11
conda activate my_env

conda install pytorch torchvision torchaudio pytorch-cuda=12.1 -c pytorch -c
nvidia

pip install transformers evaluate pandas matplotlib wandb scikit-learn
```

## 3. Writing job script

- Contoh bash script. Simpan dengan nama misal: **run_script.sh**

```
#!/bin/bash
source /home/ifti001/miniconda3/etc/profile.d/conda.sh
conda activate my_env

python python_script1.py
```

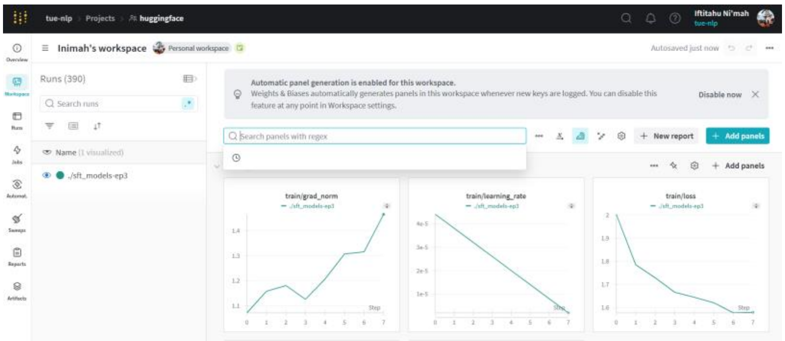
- Contoh python script yang dibaca oleh bash script. Simpan dengan nama **python_script.py**

```
import os
import sys
import numpy as np
import pandas as pd

# Please use transformers==4.45.2
import transformers
import torch


def main():

    # loading model for Huggingface hub
    model_id = "aisingapore/gemma2-9b-cpt-sea-lionv3-instruct"

    pipeline = transformers.pipeline(
        "text-generation",
        model=model_id,
        model_kwargs={"torch_dtype": torch.bfloat16},
        device_map="auto",
    )

    messages = [
        {"role": "user", "content": "Apa sentimen dari kalimat berikut ini?\nKalimat: Buku ini sangat membosankan.\nJawaban: "},
    ]

    outputs = pipeline(
        messages,
        max_new_tokens=256,
    )
    print(outputs[0]["generated_text"][-1])


if __name__ == '__main__':

    main()
```

**Q:** Bagaimana apabila saya ingin melihat progress training model secara interaktif?

**A:** Huggingface sudah mengintegrasikan module **Trainer** dengan **https://wandb.ai/**

Harap mengacu pada referensi yang ada. Contoh tampilan wanddb untuk progress training model:



Automatic panel generation is enabled for this workspace.
Weights & Biases automatically generates panels in this workspace whenever new keys are logged. You can disable this feature at any point in Workspace settings.

## 4. Transfering file

**Dari server ke lokal direktori.**

Harap menjalankan script di **Windows Powershell** untuk pengguna windows (tidak perlu login ssh ke server). Contoh:

```
scp -r ifti001@10.28.10.10:/home/ifti001/<FILEPATH> ./<LocalDIR>
```

Dari direktori lokal  ke server

```
scp -r .\<LOCAL-FILEPATH> ifti001@10.28.10.10:/home/ifti001/
```

## 5. Submit job script

```
sbatch    --partition=short    --gres=gpu:1    --mem=4G    --nodelist=a100
<BASH_SCRIPT.sh>
```

`--partition ['short', 'medium']` Ekspektasi timeline atau lama job yang akan disubmit: ‘short’ = 24 jam; ‘medium’ = 3 hari.

`--gres=gpu:1` Apakah job yang disubmit akan menggunakan device ‘gpu’ atau ‘cpu’. Untuk preprocessing atau pembacaan data dari dan ke disk space masing-masing user, utilisasinya menggunakan ‘cpu’. Untuk proses inference model yang tidak memerlukan loading model di ‘gpu’, utilisasinya bisa diarahkan ke ‘cpu’.

`--nodelist [a100, a1]` Node partisi yang akan digunakan.

`<BASH_SCRIPT.sh>` nama job script yang akan disubmit dalam antrian slurm di server.

## 6. Job Audit dan Perintah Dasar Linux

- Melihat antrian job

```
squeue -a
```

- Meng-cancel job

```
scancel <JOBID>
```

- Melihat status GPU yang tersedia

```
sinfo
```

- Melihat lokasi direktori

```
pwd
```

- Copy, move file ke directory lain

```
cp -rf <CURRENT-FILEPATH> <TARGET-DIR>
mv <CURRENT-FILEPATH> <TARGET-DIR>
```

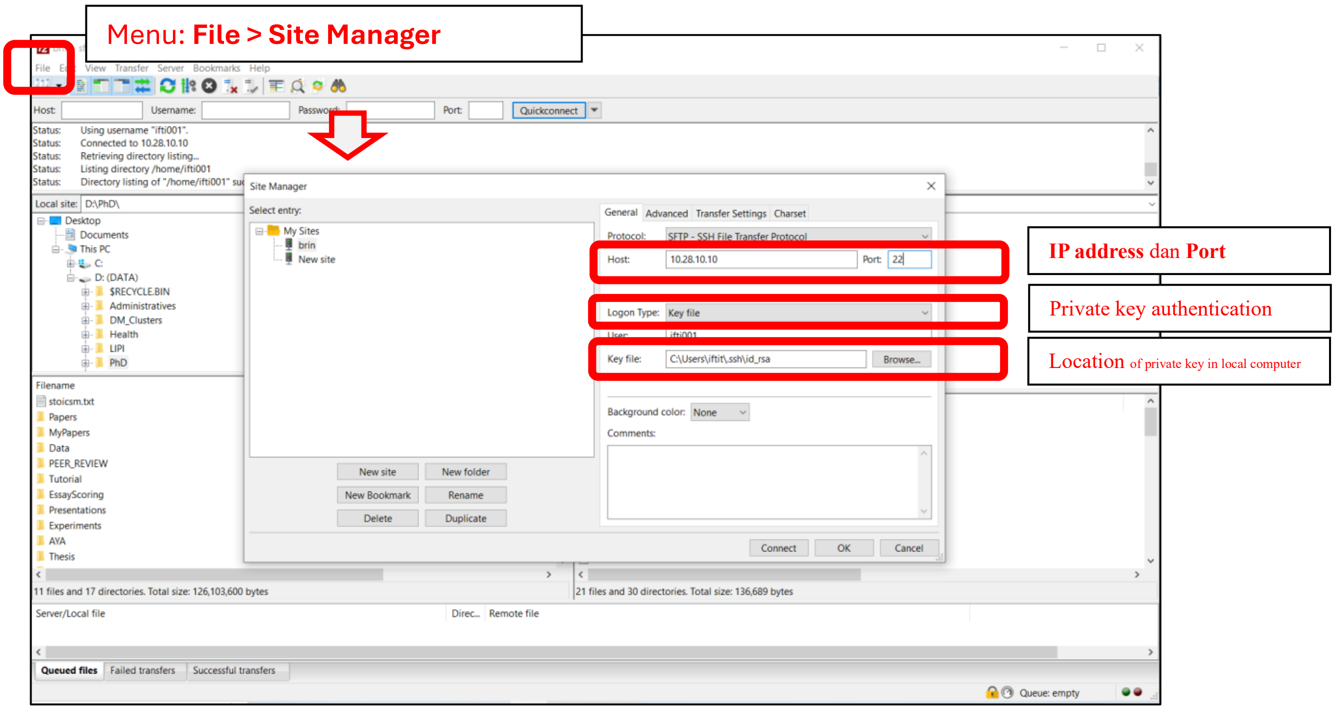
- Remove file

```
rm -rf <CURRENT-FILEPATH>
```

## 7. Setting Filezilla dengan private RSA key

Download dan install Filezilla client for windows: [https://filezilla-project.org/download.php?platform=win64](https://filezilla-project.org/download.php?platform=win64)

Konfigurasi **File > Site Manager**



Menu: **File > Site Manager**

**IP address** dan **Port**

Private key authentication

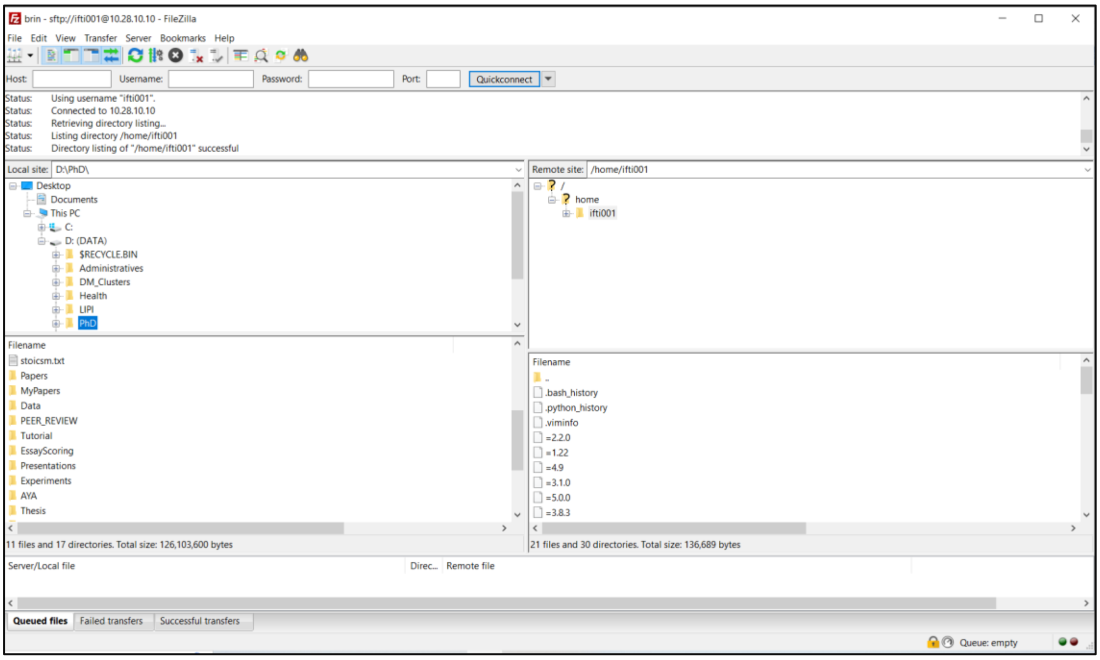
Location of private key in local computer

```
Status:    Using username "ifti001".
Status:    Connected to 10.28.10.10
Status:    Retrieving directory listing...
Status:    Listing directory /home/ifti001
Status:    Directory listing of "/home/ifti001" suc
```

Site Manager — Select entry: My Sites / brin / New site

- Protocol: SFTP - SSH File Transfer Protocol
- Host: 10.28.10.10  Port: 22
- Logon Type: Key file
- User: ifti001
- Key file: C:\Users\iftit\.ssh\id_rsa
- Background color: None

Tampilan ketika berhasil connect ke server HPC via Filezilla. Setelah ini bisa transfer file dengan drag-and-drop atau klik kanan dari dan ke server.



brin - sftp://ifti001@10.28.10.10 - FileZilla

```
Status:    Using username "ifti001".
Status:    Connected to 10.28.10.10
Status:    Retrieving directory listing...
Status:    Listing directory /home/ifti001
Status:    Directory listing of "/home/ifti001" successful
```

Local site: D:\PhD\

Remote site: /home/ifti001

Local: Filename: stoicsm.txt, Papers, MyPapers, Data, PEER_REVIEW, Tutorial, EssayScoring, Presentations, Experiments, AYA, Thesis

11 files and 17 directories. Total size: 126,103,600 bytes

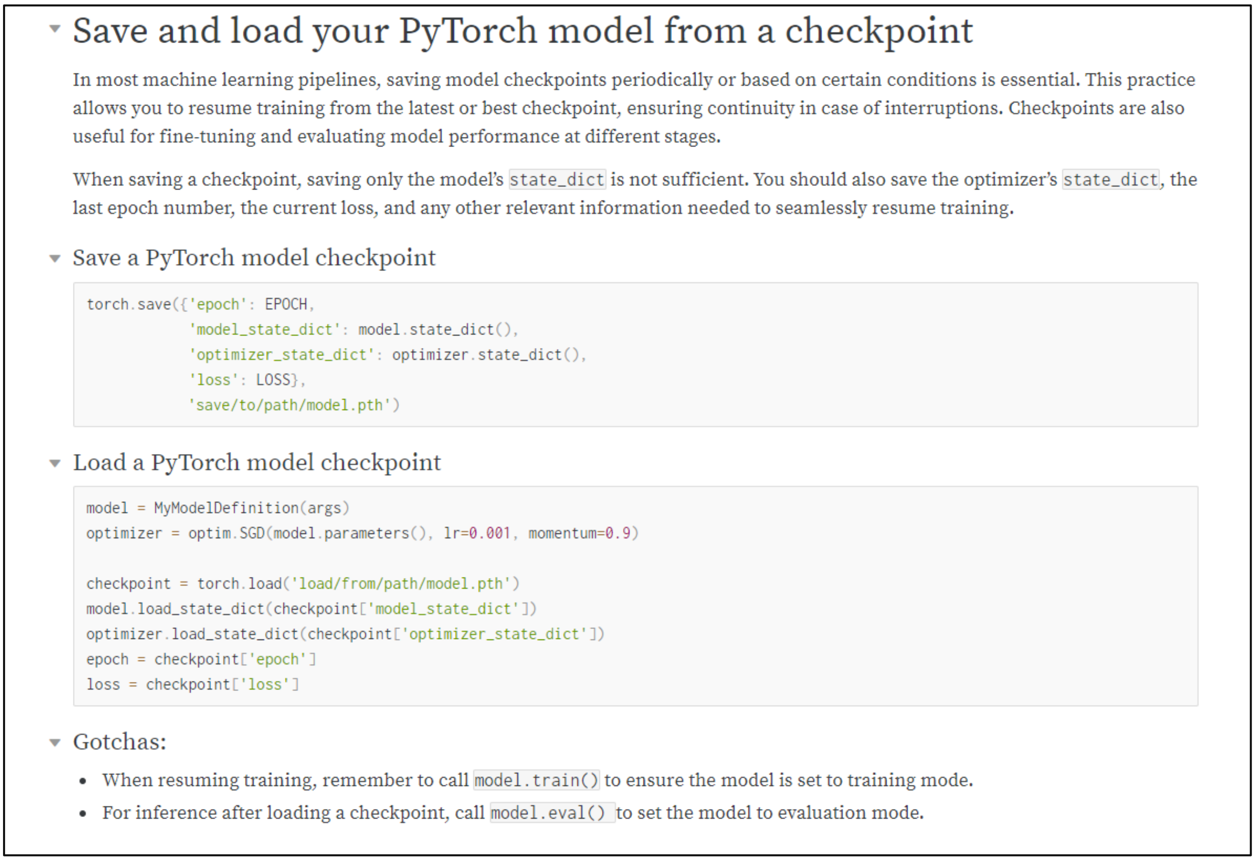
Remote: Filename: .., .bash_history, .python_history, .viminfo, =2.2.0, =1.22, =4.9, =3.1.0, =5.0.0, =3.8.3

21 files and 30 directories. Total size: 136,689 bytes

## 8. Model save checkpoints and continue training

Catatan: Silahkan disesuaikan dengan domain riset masing-masing menurut referensi yang relevant.

- Dengan Pytorch save and load (https://wandb.ai/wandb/common-ml-errors/reports/How-to-save-and-load-models-in-PyTorch--VmlldzozMjg0MTE ; https://pytorch.org/tutorials/beginner/saving_loading_models.html )



### Save and load your PyTorch model from a checkpoint

In most machine learning pipelines, saving model checkpoints periodically or based on certain conditions is essential. This practice allows you to resume training from the latest or best checkpoint, ensuring continuity in case of interruptions. Checkpoints are also useful for fine-tuning and evaluating model performance at different stages.

When saving a checkpoint, saving only the model's `state_dict` is not sufficient. You should also save the optimizer's `state_dict`, the last epoch number, the current loss, and any other relevant information needed to seamlessly resume training.

#### Save a PyTorch model checkpoint

```
torch.save({'epoch': EPOCH,
            'model_state_dict': model.state_dict(),
            'optimizer_state_dict': optimizer.state_dict(),
            'loss': LOSS},
            'save/to/path/model.pth')
```

#### Load a PyTorch model checkpoint

```
model = MyModelDefinition(args)
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

checkpoint = torch.load('load/from/path/model.pth')
model.load_state_dict(checkpoint['model_state_dict'])
optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
epoch = checkpoint['epoch']
loss = checkpoint['loss']
```

#### Gotchas:

- When resuming training, remember to call `model.train()` to ensure the model is set to training mode.
- For inference after loading a checkpoint, call `model.eval()` to set the model to evaluation mode.

- Dengan accelerator: https://huggingface.co/docs/accelerate/en/usage_guides/checkpoint
- Dengan PEFT adapter continue training: https://colab.research.google.com/drive/12pMorxvLV-VwjuNBM76L4xXnzVYg57iB?usp=sharing