

CERDAS MENGUASAI GIT

CERDAS MENGUASAI GIT

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.’
Imam Syafi’i*

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
5 Chapter 5	9
6 Chapter 6	11
7 Chapter 7	13
8 Chapter 8	15
9 Chapter 9	45
10 Chapter 10	47
11 Chapter 11	49
12 Chapter 12	51
13 Chapter 13	53
14 Chapter 14	55

DAFTAR ISI

Foreword	xi
Kata Pengantar	xiii
Acknowledgments	xv
Acronyms	xvii
Glossary	xix
List of Symbols	xxi
Introduction	xxiii
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
	ix

5	Chapter 5	9
6	Chapter 6	11
7	Chapter 7	13
8	Chapter 8	15
8.1	1174021 - Muhammad Fahmi	15
8.1.1	Soal Teori	15
8.1.2	Praktek Program	19
8.1.3	Penanganan Error	29
8.1.4	Bukti Tidak Plagiat	30
8.2	1174017 - Muh. Rifky Prananda	30
8.2.1	Soal Teori	30
8.2.2	Praktek Program	34
8.2.3	Penanganan Error	44
8.2.4	Bukti Tidak Plagiat	44
9	Chapter 9	45
10	Chapter 10	47
11	Chapter 11	49
12	Chapter 12	51
13	Chapter 13	53
14	Chapter 14	55
	Daftar Pustaka	57
	Index	59

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

Bandung, Jawa Barat
Februari, 2019

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

CHAPTER 1

BAB 2

CHAPTER 2

BAB 3

CHAPTER 3

BAB 4

CHAPTER 4

BAB 5

CHAPTER 5

BAB 6

CHAPTER 6

BAB 7

CHAPTER 7

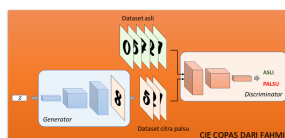
BAB 8

CHAPTER 8

8.1 1174021 - Muhammad Fahmi

8.1.1 Soal Teori

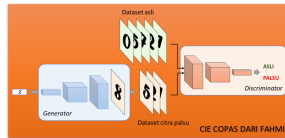
1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.
Tugas Generator sekarang sedang dibuat untuk membuat koleksi gambar palsu, yang saat ini dilihat oleh Diskriminator. Diskriminator tidak dapat membedakan antara yang asli dan yang palsu. Untuk ilustrasi, lihat gambar berikut:



Gambar 8.1 Teori 1

2. Jelaskan dengan ilustrasi gambar sendiri apa itu diskriminator dengan perumpamaan dosen anda sebagai diskriminatornya.

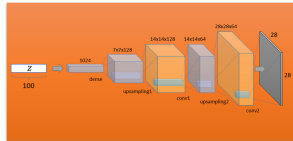
Diskriminator adalah CNN yang menerima input gambar yang dimiliki dan menghasilkan angka biner yang meminta input gambar, lalu menghasilkan gambar dari dataset asli atau menghasilkan gambar palsu. Untuk ilustrasi, lihat gambar berikut:



Gambar 8.2 Teori 2

3. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur generator dibuat.

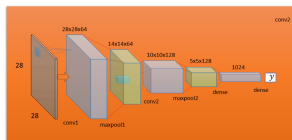
Aksitektur generator dibuat bisa dijelaskan pada gambar berikut :



Gambar 8.3 Teori 3

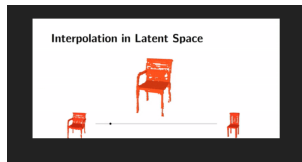
4. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur diskriminator dibuat.

Aksitektur diskriminator dibuat bisa dijelaskan pada gambar berikut :



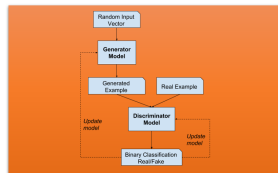
Gambar 8.4 Teori 4

5. Jelaskan dengan ilustrasi gambar apa itu latent space. Latent space dijelaskan pada gambar berikut :



Gambar 8.5 Teori 5

6. Jelaskan dengan ilustrasi gambar apa itu adversarial play.
Adversarial play dijelaskan pada gambar berikut :



Gambar 8.6 Teori 6

7. Jelaskan dengan ilustrasi gambar apa itu Nash equilibrium.
Nash equilibrium adalah Teori permainan adalah studi tentang interaksi strategis antara agen rasional. Sederhananya itu berarti itu adalah studi interaksi ketika pihak-pihak yang terlibat mencoba dan melakukan yang terbaik dari sudut pandang mereka, detailnya dapat dijelaskan pada gambar berikut :

```
In [4]: runfile('D:/OLDER KAMUS MAMPUUS/SEMESTER 6/A  
8/src/1174021/tugas8/teori3.py', wdir='D:/OLDER KAMUS  
SEMESTER 6/A/8/PULL 8/src/1174021/tugas8')  
Bi matrix game with payoff matrices:  
  
Row player:  
[[3 0]  
 [4 1]]  
  
Column player:  
[[3 4]  
 [0 1]]
```

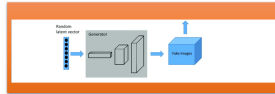
Gambar 8.7 Teori 7

8. Sebutkan dan jelaskan contoh-contoh implementasi dari GAN.

Menurut saya implementasi 3DGAN yaitu pada MAPS dan juga IKEA. Karena pada maps dan juga ikea sudah menerapkan bentuk 3 dimensi yang bisa lebih menarik perhatian pengguna.

9. Berikan contoh dengan penjelasan kode program beserta gambar arsitektur untuk membuat generator(neural network) dengan sebuah input layer, tiga hidden layer(dense layer), dan satu output layer(reshape layer).

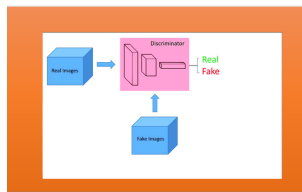
Untuk penjelasan tersebut dijelaskan pada gambar dibawah ini :



Gambar 8.8 Teori 9

10. Berikan contoh dengan ilustrasi dari arsitektur dikriminator dengan sebuah input layer, 3 buah hidden layer, dan satu output layer.

Untuk penjelasan tersebut dijelaskan pada gambar dibawah ini :



Gambar 8.9 Teori 10

11. Jelaskan bagaimana kaitan output dan input antara generator dan diskriminator tersebut. Jelaskan kenapa inputan dan outputan seperti itu.

Gambar dari Generator yang berhasil di deteksi oleh Diskriminator sebagai gambar fake, akan dikembalikan dengan feedback pke generator. Kini Generator bertugas untuk bisa membuat sekumpulan gambar palsu, yang nantinya dapat dilihat oleh Diskriminator, lalu, Diskriminator tidak bisa membedakan fake dan real.

12. Jelaskan apa perbedaan antara Kullback-Leibler divergence (KL divergence)/relative entropy / Jensen-Shannon(JS) divergence / information radius(iRaD) / total divergence to the average dalam mengukur kualitas dari model.

Perbedaan nya yaitu memiliki model dari rumus yang berbeda-beda sehingga mempengaruhi hasil train dan test

13. Jelaskan apa itu fungsi objektif yang berfungsi untuk mengukur kesamaan antara gambar yang dibuat dengan yang asli.

Ukuran penting untuk menilai kualitas model. lalu kemudian akan melihat di keseimbangan Nash.


```

19     a = BatchNormalization()(a, training=True)
20     a = Activation(activation='relu')(a)
21
22     # Next 4 3D transpose convolution(or 3D deconvolution) blocks
23     for i in range(gen_convolutional_blocks - 1):
24         a = Deconv3D(filters=gen_filters[i + 1],
25                     kernel_size=gen_kernel_sizes[i + 1],
26                     strides=gen_strides[i + 1], padding='same')(
27             a)
28         a = BatchNormalization()(a, training=True)
29         a = Activation(activation=gen_activations[i + 1])(a)
30
31     gen_model = Model(inputs=[input_layer], outputs=[a])
32     return gen_model

```

Kode di atas akan melakukan create generator ialah gloss, Bentuk jaringan Generator dapat dilihat berkebalikan dengan struktur jaringan saraf pada umumnya. Generator biasanya menerima input sebuah vektor z, yang kemudian mengubahnya menjadi sebuah output 3D atau 3 dimensi.

3. Soal 3

```

1 def build_discriminator():
2     """
3     Create a Discriminator Model using hyperparameters values
4     defined as follows
5     """
6
7     dis_input_shape = (64, 64, 64, 1)
8     dis_filters = [64, 128, 256, 512, 1]
9     dis_kernel_sizes = [4, 4, 4, 4, 4]
10    dis_strides = [2, 2, 2, 2, 1]
11    dis_paddings = ['same', 'same', 'same', 'same', 'valid']
12    dis_alphas = [0.2, 0.2, 0.2, 0.2, 0.2]
13    dis_activations = ['leaky_relu', 'leaky_relu', 'leaky_relu',
14                      'leaky_relu', 'sigmoid']
15    dis_convolutional_blocks = 5
16
17    dis_input_layer = Input(shape=dis_input_shape)
18
19    # The first 3D Convolutional block
20    a = Conv3D(filters=dis_filters[0],
21              kernel_size=dis_kernel_sizes[0],
22              strides=dis_strides[0],
23              padding=dis_paddings[0])(dis_input_layer)
24    # a = BatchNormalization()(a, training=True)
25    a = LeakyReLU(dis_alphas[0])(a)
26
27    # Next 4 3D Convolutional Blocks
28    for i in range(dis_convolutional_blocks - 1):
29        a = Conv3D(filters=dis_filters[i + 1],
30                  kernel_size=dis_kernel_sizes[i + 1],
31                  strides=dis_strides[i + 1],
32                  padding=dis_paddings[i + 1])(a)
33        a = BatchNormalization()(a, training=True)

```

```

33     if dis_activations[i + 1] == 'leaky_relu':
34         a = LeakyReLU(dis_alphas[i + 1])(a)
35     elif dis_activations[i + 1] == 'sigmoid':
36         a = Activation(activation='sigmoid')(a)
37
38     dis_model = Model(inputs=[dis_input_layer], outputs=[a])
39     return dis_model

```

Diskriminator adalah `d_loss`, Jaringan Discriminator merupakan jaringan klasifikasi biner yang menerima input gambar tiga dimensi dan mengeluarkan klasifikasi menyatakan input gambar adalah gambar asli dari dataset atau merupakan gambar buatan Generator. Diskriminator dilatih dengan dataset yang diambil dari Generator, lalu di training untuk membedakan keduanya. Gambar dari Generator yang berhasil di deteksi oleh Diskriminator sebagai gambar fake, akan dikembalikan dengan feedback pke generator. Kini Generator bertugas untuk bisa membuat sekumpulan gambar palsu, yang nantinya dapat dilihat oleh Diskriminator, lalu, Diskriminator tidak bisa membedakan fake dan real.

4. Soal 4

Proses training 3D GAN yaitu dengan melakukan epoch sebanyak yang ditentukan.

5. Soal 5

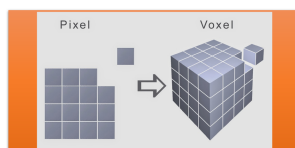
- Clone github
- Download dataset
- Buat folder baru logs dan results

6. Soal 6

Dataset yang digunakan yaitu 3DShapeNets yang berisi model model bentuk benda dll, folder train berisi train dan folder test berisi data testing. dan semua data tersebut di simpan didalam folder volumetric_data.

7. Soal 7

Volume pixel atau voxel adalah titik dalam ruang tiga dimensi. Sebuah voxel mendefinisikan posisi dengan tiga koordinat dalam arah x, y, dan z. Sebuah voxel adalah unit dasar untuk mewakili gambar 3D. Untuk gambar nya ialah sebagai berikut :



Gambar 8.10 Praktek Soal 7

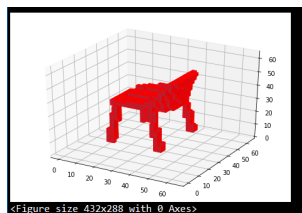
8. Soal 8

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Sun May 10 18:10:55 2020
4
5  @author: FAHMI-PC
6
7  Cieee Copas : 'v
8  """
9
10 # In []
11
12 import scipy.io as io
13 voxels = io.loadmat("data/3DShapeNets/ volumetric_data / chair / 30 /
14 test / chair_000000000_1 .mat") ['instance']
15
16 #%%
17 import numpy as np
18 voxels = np.pad(voxels, (1, 1), 'constant', constant_values=(0,
19 0))
20
21 #%%
22 import scipy.ndimage as nd
23 voxels = nd.zoom(voxels, (2, 2, 2), mode='constant', order=0)
24
25 #%%
26 import matplotlib.pyplot as plt
27 from mpl_toolkits.mplot3d import Axes3D
28 fig = plt.figure()
29 ax = Axes3D(fig)
30 ax.voxels(voxels, edgecolor="red")
31
32 plt.show()
33 plt.savefig('data')

```

Kode di atas berfungsi untuk visualisasi dataset dalam tampilan plot. langkah-langkah seperti ini : import library, load data file .mat dan lakukan read memakai matplotlib, Hasilnya adalah sebagai berikut :



Gambar 8.11 Hasil Soal 8

9. Soal 9

```

1  #%% soal9
2

```

```

3 def build_generator():
4     """
5     Create a Generator Model with hyperparameters values defined
6     as follows
7     """
8     z_size = 200
9     gen_filters = [512, 256, 128, 64, 1]
10    gen_kernel_sizes = [4, 4, 4, 4, 4]
11    gen_strides = [1, 2, 2, 2, 2]
12    gen_input_shape = (1, 1, 1, z_size)
13    gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
14    gen_convolutional_blocks = 5
15
16    input_layer = Input(shape=gen_input_shape)
17
18    # First 3D transpose convolution(or 3D deconvolution) block
19    a = Deconv3D(filters=gen_filters[0],
20                kernel_size=gen_kernel_sizes[0],
21                strides=gen_strides[0])(input_layer)
22    a = BatchNormalization()(a, training=True)
23    a = Activation(activation='relu')(a)
24
25    # Next 4 3D transpose convolution(or 3D deconvolution) blocks
26    for i in range(gen_convolutional_blocks - 1):
27        a = Deconv3D(filters=gen_filters[i + 1],
28                    kernel_size=gen_kernel_sizes[i + 1],
29                    strides=gen_strides[i + 1], padding='same')(
30            a)
31        a = BatchNormalization()(a, training=True)
32        a = Activation(activation=gen_activations[i + 1])(a)
33
34    gen_model = Model(inputs=[input_layer], outputs=[a])
35    return gen_model

```

Kode di atas berfungsi untuk membuat generator yaitu dengan ketentuan gen sebagai variabel dan membuat fungsi atau variabel genmodel lalu dilakukan return.

10. Soal 10

```

1 #%% soal10
2
3
4 def build_discriminator():
5     """
6     Create a Discriminator Model using hyperparameters values
7     defined as follows
8     """
9
10    dis_input_shape = (64, 64, 64, 1)
11    dis_filters = [64, 128, 256, 512, 1]
12    dis_kernel_sizes = [4, 4, 4, 4, 4]
13    dis_strides = [2, 2, 2, 2, 1]
14    dis_paddings = ['same', 'same', 'same', 'same', 'valid']
15    dis_alphas = [0.2, 0.2, 0.2, 0.2, 0.2]

```



```

15     dis_activations = ['leaky_relu', 'leaky_relu', 'leaky_relu',
16                        'leaky_relu', 'sigmoid']
17     dis_convolutional_blocks = 5
18
19     dis_input_layer = Input(shape=dis_input_shape)
20
21     # The first 3D Convolutional block
22     a = Conv3D(filters=dis_filters[0],
23               kernel_size=dis_kernel_sizes[0],
24               strides=dis_strides[0],
25               padding=dis_paddings[0])(dis_input_layer)
26     # a = BatchNormalization()(a, training=True)
27     a = LeakyReLU(dis_alphas[0])(a)
28
29     # Next 4 3D Convolutional Blocks
30     for i in range(dis_convolutional_blocks - 1):
31         a = Conv3D(filters=dis_filters[i + 1],
32                   kernel_size=dis_kernel_sizes[i + 1],
33                   strides=dis_strides[i + 1],
34                   padding=dis_paddings[i + 1])(a)
35         a = BatchNormalization()(a, training=True)
36         if dis_activations[i + 1] == 'leaky_relu':
37             a = LeakyReLU(dis_alphas[i + 1])(a)
38         elif dis_activations[i + 1] == 'sigmoid':
39             a = Activation(activation='sigmoid')(a)
40
41     dis_model = Model(inputs=[dis_input_layer], outputs=[a])
42     return dis_model

```

Kode di atas berfungsi untuk membangun diskriminator berfungsi untuk mendefinisikan seluruh gambar yang sudah di load generator sebagai gambar fake dan real.

11. Soal 11

Jika interpreter python menjalankan `if name == main` sebagai program utama, itu ialah menetapkan variabel `name` untuk memiliki nilai `main`. Jika file ini sedang di impor dari modul lain, `name` akan ditetapkan ke nama modul. Nama modul tersedia sebagai nilai untuk `name` variabel global.

12. Soal 12

```

1  ### soal 12
2  object_name = "airplane"
3  data_dir = "data/3DShapeNets/volumetric_data/" \
4            "{}/30/train/*.mat".format(object_name)
5  gen_learning_rate = 0.0025
6  dis_learning_rate = 10e-5
7  beta = 0.5
8  batch_size = 1
9  z_size = 200
10 epochs = 1
11 MODE = "train"

```

Kode di atas berfungsi untuk melakukan load dataset dengan ketentuan data yang hanya dalam folder chair pada data train.

13. Soal 13

```

1  ##### soal 13
2      """
3      Create models
4      """
5      gen_optimizer = Adam(lr=gen_learning_rate , beta_1=beta)
6      dis_optimizer = Adam(lr=dis_learning_rate , beta_1=beta)
7
8      discriminator = build_discriminator()
9      discriminator.compile(loss='binary_crossentropy', optimizer=
10         dis_optimizer)
11
12     generator = build_generator()
13     generator.compile(loss='binary_crossentropy', optimizer=
14         gen_optimizer)

```

Kode di atas menggunakan Adam sebagai algoritma pengoptimalan dan binary_crossentropy sebagai kerugian loss.

14. Soal 14

```

1  ##### soal14
2      discriminator.trainable = False
3
4      input_layer = Input(shape=(1, 1, 1, z_size))
5      generated_volumes = generator(input_layer)
6      validity = discriminator(generated_volumes)
7      adversarial_model = Model(inputs=[input_layer], outputs=[
8         validity])
9      adversarial_model.compile(loss='binary_crossentropy',
10         optimizer=gen_optimizer)

```

Kode di atas artinya ialah kita memasukkan random vector kedalam generator model lalu membagi 2 yaitu generated example dan real example, dan meneruskan ke diskriminator model sebagai real atau fake

15. Soal 15

```

1  ##### soal15
2      print("Loading data...")
3      volumes = get3DImages(data_dir=data_dir)
4      volumes = volumes[..., np.newaxis].astype(np.float)
5      print("Data loaded...")

```

Kode di atas berfungsi untuk melakukan load data pada dataset.

16. Soal 16

```

1  ##### soal16
2      tensorboard = TensorBoard(log_dir="logs/{}".format(time.time
   ()))
3      tensorboard.set_model(generator)
4      tensorboard.set_model(discriminator)

```

Kode di atas berfungsi untuk membuat tensorboard yang nantinya bisa diakses melalui localhost.

17. Soal 17

```

1  ##### soal17
2      labels_real = np.reshape(np.ones((batch_size,)), (-1, 1, 1,
   1, 1))
3      labels_fake = np.reshape(np.zeros((batch_size,)), (-1, 1, 1,
   1, 1))

```

Kode di atas berfungsi untuk melakukan reshape agar shape yang dihasilkan tidak terlalu besar. Dengan membuat variabel real dan fake.

18. Soal 18

```

1  ##### soal18
2      if MODE == 'train':
3          for epoch in range(epochs):
4              print("Epoch:", epoch)
5
6              gen_losses = []
7              dis_losses = []

```

Kode di atas berfungsi untuk melakukan training epoch, karena jika epoch semakin banyak maka kualitas training yang dihasilkan akan semakin baik.

19. Soal 19

```

1  ##### soal19
2      number_of_batches = int(volumes.shape[0] / batch_size
   )
3      print("Number of batches:", number_of_batches)
4      for index in range(number_of_batches):
5          print("Batch:", index + 1)

```

Batch adalah jumlah file yang akan di training.

20. Soal 20

```

1  ##### soal20
2      z_sample = np.random.normal(0, 0.33, size=[
   batch_size, 1, 1, 1, z_size]).astype(np.float32)
3      volumes_batch = volumes[index * batch_size:(index
   + 1) * batch_size, :, :, :]

```

Kode di atas berfungsi untuk membuat gambar bersih dari noise dan juga menyesuaikan shape.

21. Soal 21

```
1 #%% soal21
2         # Next, generate volumes using the generate
   network
3         gen_volumes = generator.predict_on_batch(z_sample
   )
```

Kode di atas berfungsi untuk membuat sample gambar palsu yang akan diteruskan ke diskriminator.

22. Soal 22

```
1 #%% soal22
2         discriminator.trainable = True
3         if index % 2 == 0:
4             loss_real = discriminator.train_on_batch(
   volumes_batch, labels_real)
5             loss_fake = discriminator.train_on_batch(
   gen_volumes, labels_fake)
6
7             d_loss = 0.5 * np.add(loss_real, loss_fake)
8             print("d_loss:{}".format(d_loss))
9
10        else:
11            d_loss = 0.0
```

Kode di atas berfungsi untuk membuat diskriminator bisa load gambar fake dan real dari generator, oleh karena itu ada generator loss dan diskriminator loss untuk melihat seberapa baik kualitas yang dihasilkan.

23. Soal 23

```
1 #%% soal23
2         discriminator.trainable = False
3         """
4         Train the generator network
5         """
6         z = np.random.normal(0, 0.33, size=[batch_size,
   1, 1, 1, z_size]).astype(np.float32)
7         g_loss = adversarial_model.train_on_batch(z,
   labels_real)
8         print("g_loss:{}".format(g_loss))
9
10        gen_losses.append(g_loss)
11        dis_losses.append(d_loss)
```

Kode di atas berfungsi untuk melakukan print gloss untuk generator dan juga dloss untuk diskriminator.

24. Soal 24

```

1  ##### soal24
2          # Every 10th mini-batch, generate volumes and
   save them
3          if index % 10 == 0:
4              z_sample2 = np.random.normal(0, 0.33, size=[
   batch_size, 1, 1, 1, z_size]).astype(np.float32)
5              generated_volumes = generator.predict(
   z_sample2, verbose=3)
6              for i, generated_volume in enumerate(
   generated_volumes[:5]):
7                  voxels = np.squeeze(generated_volume)
8                  voxels[voxels < 0.5] = 0.
9                  voxels[voxels >= 0.5] = 1.
10                 saveFromVoxels(voxels, "results/img-{}-{}
   -{}".format(epoch, index, i))

```

Mengapa ada perulangan ? karena untuk melakukan perbandingan dari hasil yang sudah didapat.

25. Soal 25

```

1  ##### soal25
2          # Write losses to Tensorboard
3          write_log(tensorboard, 'g_loss', np.mean(gen_losses),
   epoch)
4          write_log(tensorboard, 'd_loss', np.mean(dis_losses),
   epoch)

```

TensorBoard adalah sebuah aplikasi web localhost untuk memeriksa dan menyelesaikan grafik dari hasil TensorFlow.

26. Soal 26

```

1  ##### soal26
2          generator.save_weights(os.path.join("models", "
   generator_weights.h5"))
3          discriminator.save_weights(os.path.join("models", "
   discriminator_weights.h5"))

```

File H5 adalah file data yang disimpan dalam Format Data Hirarki (HDF). Ini berisi array multidimensi data ilmiah.

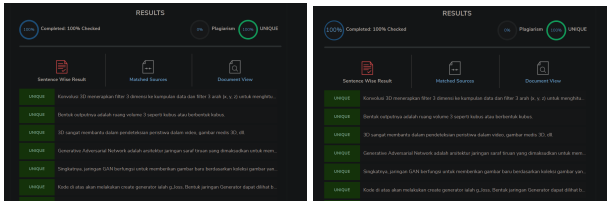
27. Soal 27

```

1  ##### soal27
2          if MODE == 'predict':
3              # Create models
4              generator = build_generator()
5              discriminator = build_discriminator()

```


8.1.4 Bukti Tidak Plagiat

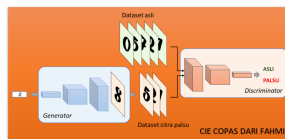


Gambar 8.14 Bukti Tidak Melakukan Plagiat Chapter 8

8.2 1174017 - Muh. Rifky Prananda

8.2.1 Soal Teori

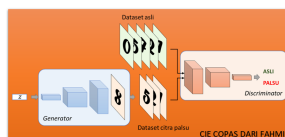
1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.
Tugas Generator sekarang sedang dibuat untuk membuat koleksi gambar palsu, yang saat ini dilihat oleh Diskriminator. Diskriminator tidak dapat membedakan antara yang asli dan yang palsu. Untuk ilustrasi, lihat gambar berikut:



Gambar 8.15 Teori 1

2. Jelaskan dengan ilustrasi gambar sendiri apa itu diskriminator dengan perumpamaan dosen anda sebagai diskriminatornya.

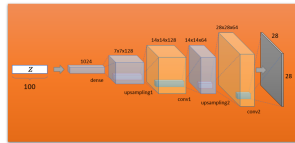
Diskriminator adalah CNN yang menerima input gambar yang dimiliki dan menghasilkan angka biner yang meminta input gambar, lalu menghasilkan gambar dari dataset asli atau menghasilkan gambar palsu. Untuk ilustrasi, lihat gambar berikut:



Gambar 8.16 Teori 2

3. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur generator dibuat.

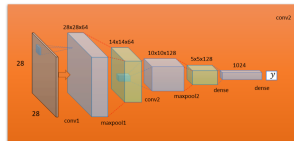
Aksitektur generator dibuat bisa dijelaskan pada gambar berikut :



Gambar 8.17 Teori 3

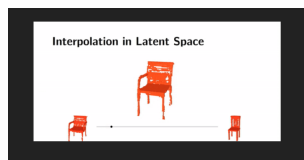
4. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur diskriminator dibuat.

Aksitektur diskriminator dibuat bisa dijelaskan pada gambar berikut :



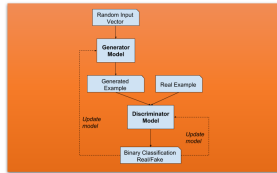
Gambar 8.18 Teori 4

5. Jelaskan dengan ilustrasi gambar apa itu latent space.
Latent space dijelaskan pada gambar berikut :



Gambar 8.19 Teori 5

6. Jelaskan dengan ilustrasi gambar apa itu adversarial play.
Adversarial play dijelaskan pada gambar berikut :



Gambar 8.20 Teori 6

7. Jelaskan dengan ilustrasi gambar apa itu Nash equilibrium.

Nash equilibrium adalah Teori permainan adalah studi tentang interaksi strategis antara agen rasional. Sederhananya itu berarti itu adalah studi interaksi ketika pihak-pihak yang terlibat mencoba dan melakukan yang terbaik dari sudut pandang mereka, detailnya dapat dijelaskan pada gambar berikut :

```

In [4]: runfile('D:/FOLDER KHASIS NGANUS/SEMESTER 6/
          8/src/1174021/tugas8/teori.py', wdir='D:/FOLDER KHASIS
          PERTEMUAN KE 8/AI PUL 8/src/1174021/tugas8')
Bi matrix game with payoff matrices:

Row player:
[[3 0]
 [4 1]]

Column player:
[[3 4]
 [0 1]]
  
```

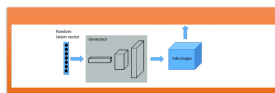
Gambar 8.21 Teori 7

8. Sebutkan dan jelaskan contoh-contoh implementasi dari GAN.

Menurut saya implementasi 3DGAN yaitu pada MAPS dan juga IKEA. Karena pada maps dan juga ikea sudah menerapkan bentuk 3 dimensi yang bisa lebih menarik perhatian pengguna.

9. Berikan contoh dengan penjelasan kode program beserta gambar arsitektur untuk membuat generator(neural network) dengan sebuah input layer, tiga hidden layer(dense layer), dan satu output layer(reshape layer).

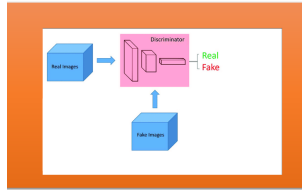
Untuk penjelasan tersebut dijelaskan pada gambar dibawah ini :



Gambar 8.22 Teori 9

10. Berikan contoh dengan ilustrasi dari arsitektur dikriminator dengan sebuah input layer, 3 buah hidden layer, dan satu output layer.

Untuk penjelasan tersebut dijelaskan pada gambar dibawah ini :



Gambar 8.23 Teori 10

11. Jelaskan bagaimana kaitan output dan input antara generator dan diskriminator tersebut. Jelaskan kenapa inputan dan outputan seperti itu.

Gambar dari Generator yang berhasil di deteksi oleh Diskriminator sebagai gambar fake, akan dikembalikan dengan feedback pke generator. Kini Generator bertugas untuk bisa membuat sekumpulan gambar palsu, yang nantinya dapat dilihat oleh Diskriminator, lalu, Diskriminator tidak bisa membedakan fake dan real.

12. Jelaskan apa perbedaan antara Kullback-Leibler divergence (KL divergence)/relative entropy / Jensen-Shannon(JS) divergence / information radius(iRaD) / total divergence to the average dalam mengukur kualitas dari model.

Perbedaan nya yaitu memiliki model dari rumus yang berbeda-beda sehingga mempengaruhi hasil train dan test

13. Jelaskan apa itu fungsi objektif yang berfungsi untuk mengukur kesamaan antara gambar yang dibuat dengan yang asli.

Ukuran penting untuk menilai kualitas model. lalu kemudian akan melihat di keseimbangan Nash.

14. Jelaskan apa itu scoring algoritma selain mean square error atau cross entropy seperti The Inception Score dan The Frechet Inception distance.

The inception score adalah algoritma penilaian yang paling banyak digunakan untuk GAN. The Frechet Inception distance adalah Untuk mengatasi berbagai kekurangan Skor awal

15. Jelaskan kelebihan dan kekurangan GAN.

Kelebihan : GAN dapat memvisualisasikan bentuk model menjadi plot. Kemudian pada Kelemahan : model susah untuk implementasikan yang membuat data training menjadi lemah.

8.2.2 Praktek Program

1. Soal 1

Konvolusi 3D. Konvolusi 3D menerapkan filter 3 dimensi ke kumpulan data dan filter 3 arah (x, y, z) untuk menghitung representasi fitur tingkat rendah. Bentuk outputnya adalah ruang volume 3 seperti kubus atau berbentuk kubus. 3D sangat membantu dalam pendeteksian peristiwa dalam video, gambar medis 3D, dll. Generative Adversarial Network adalah arsitektur jaringan saraf tiruan yang dimaksudkan untuk membuat atau membuat data yang benar-benar baru, dari nol hingga tidak ada sama sekali. Melihat target utama GAN adalah data gambar. Singkatnya, jaringan GAN berfungsi untuk memberikan gambar baru berdasarkan koleksi gambar yang telah ada sebelumnya selama proses training.

2. Soal 2

```

1 def build_generator():
2     """
3     Create a Generator Model with hyperparameters values defined
4     as follows
5     """
6     z_size = 200
7     gen_filters = [512, 256, 128, 64, 1]
8     gen_kernel_sizes = [4, 4, 4, 4, 4]
9     gen_strides = [1, 2, 2, 2, 2]
10    gen_input_shape = (1, 1, 1, z_size)
11    gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
12    gen_convolutional_blocks = 5
13
14    input_layer = Input(shape=gen_input_shape)
15
16    # First 3D transpose convolution(or 3D deconvolution) block
17    a = Deconv3D(filters=gen_filters[0],
18                kernel_size=gen_kernel_sizes[0],
19                strides=gen_strides[0])(input_layer)
20    a = BatchNormalization()(a, training=True)
21    a = Activation(activation='relu')(a)
22
23    # Next 4 3D transpose convolution(or 3D deconvolution) blocks
24    for i in range(gen_convolutional_blocks - 1):
25        a = Deconv3D(filters=gen_filters[i + 1],
26                    kernel_size=gen_kernel_sizes[i + 1],
27                    strides=gen_strides[i + 1], padding='same')(
28            a)
29        a = BatchNormalization()(a, training=True)
30        a = Activation(activation=gen_activations[i + 1])(a)
31
32    gen_model = Model(inputs=[input_layer], outputs=[a])
33    return gen_model

```

Kode di atas akan melakukan create generator ialah gloss, Bentuk jaringan Generator dapat dilihat berkebalikan dengan struktur jaringan saraf pada umum-

nya. Generator biasanya menerima input sebuah vektor z , yang kemudian mengubahnya menjadi sebuah output 3D atau 3 dimensi.

3. Soal 3

```

1 def build_discriminator():
2     """
3     Create a Discriminator Model using hyperparameters values
4     defined as follows
5     """
6     dis_input_shape = (64, 64, 64, 1)
7     dis_filters = [64, 128, 256, 512, 1]
8     dis_kernel_sizes = [4, 4, 4, 4, 4]
9     dis_strides = [2, 2, 2, 2, 1]
10    dis_paddings = ['same', 'same', 'same', 'same', 'valid']
11    dis_alphas = [0.2, 0.2, 0.2, 0.2, 0.2]
12    dis_activations = ['leaky_relu', 'leaky_relu', 'leaky_relu',
13                      'leaky_relu', 'sigmoid']
14    dis_convolutional_blocks = 5
15
16    dis_input_layer = Input(shape=dis_input_shape)
17
18    # The first 3D Convolutional block
19    a = Conv3D(filters=dis_filters[0],
20              kernel_size=dis_kernel_sizes[0],
21              strides=dis_strides[0],
22              padding=dis_paddings[0])(dis_input_layer)
23    # a = BatchNormalization()(a, training=True)
24    a = LeakyReLU(dis_alphas[0])(a)
25
26    # Next 4 3D Convolutional Blocks
27    for i in range(dis_convolutional_blocks - 1):
28        a = Conv3D(filters=dis_filters[i + 1],
29                  kernel_size=dis_kernel_sizes[i + 1],
30                  strides=dis_strides[i + 1],
31                  padding=dis_paddings[i + 1])(a)
32        a = BatchNormalization()(a, training=True)
33        if dis_activations[i + 1] == 'leaky_relu':
34            a = LeakyReLU(dis_alphas[i + 1])(a)
35        elif dis_activations[i + 1] == 'sigmoid':
36            a = Activation(activation='sigmoid')(a)
37
38    dis_model = Model(inputs=[dis_input_layer], outputs=[a])
39    return dis_model

```

Diskriminator adalah d_{loss} , Jaringan Discriminator merupakan jaringan klasifikasi biner yang menerima input gambar tiga dimensi dan mengeluarkan klasifikasi menyatakan input gambar adalah gambar asli dari dataset atau merupakan gambar buatan Generator. Diskriminator dilatih dengan dataset yang diambil dari Generator, lalu di training untuk membedakan keduanya. Gambar dari Generator yang berhasil di deteksi oleh Diskriminator sebagai gambar fake, akan dikembalikan dengan feedback pke generator. Kini Generator bertugas un-

tuk bisa membuat sekumpulan gambar palsu, yang nantinya dapat dilihat oleh Diskriminator, lalu, Diskriminator tidak bisa membedakan fake dan real.

4. Soal 4

Proses training 3D GAN yaitu dengan melakukan epoch sebanyak yang ditentukan.

5. Soal 5

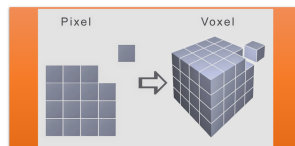
- Clone github
- Download dataset
- Buat folder baru logs dan results

6. Soal 6

Dataset yang digunakan yaitu 3DShapeNets yang berisi model model bentuk benda dll, folder train berisi train dan folder test berisi data testing. dan semua data tersebut di simpan didalam folder volumetric_data.

7. Soal 7

Volume pixel atau voxel adalah titik dalam ruang tiga dimensi. Sebuah voxel mendefinisikan posisi dengan tiga koordinat dalam arah x, y, dan z. Sebuah voxel adalah unit dasar untuk mewakili gambar 3D. Untuk gambar nya ialah sebagai berikut :



Gambar 8.24 Praktek Soal 7

8. Soal 8

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue May 12 01:51:34 2020
4
5 @author: Rifky
6
7
8 """
9
10 # In []
11
12 import scipy.io as io
13 voxels = io.loadmat("data/3DShapeNets/volumetric_data/chair/30/
    test/chair_000000000_1.mat")['instance']

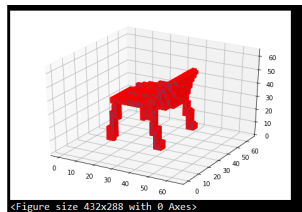
```

```

14 #%%
15 import numpy as np
16 voxels = np.pad(voxels, (1, 1), 'constant', constant_values=(0,
17 0))
18 #%%
19 import scipy.ndimage as nd
20 voxels = nd.zoom(voxels, (2, 2, 2), mode='constant', order=0)
21 #%%
22 import matplotlib.pyplot as plt
23 from mpl_toolkits.mplot3d import Axes3D
24 fig = plt.figure()
25 ax = Axes3D(fig)
26 ax.voxels(voxels, edgecolor="red")
27 plt.show()
28 plt.savefig('data')

```

Kode di atas berfungsi untuk visualisasi dataset dalam tampilan plot. langkah-langkah seperti ini : import library, load data file .mat dan lakukan read memakai matplotlib, Hasilnya adalah sebagai berikut :



Gambar 8.25 Hasil Soal 8

9. Soal 9

```

1 #%% soal9
2
3 def build_generator():
4     """
5     Create a Generator Model with hyperparameters values defined
6     as follows
7     """
8     z_size = 200
9     gen_filters = [512, 256, 128, 64, 1]
10    gen_kernel_sizes = [4, 4, 4, 4, 4]
11    gen_strides = [1, 2, 2, 2, 2]
12    gen_input_shape = (1, 1, 1, z_size)
13    gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
14    gen_convolutional_blocks = 5
15
16    input_layer = Input(shape=gen_input_shape)
17
18    # First 3D transpose convolution(or 3D deconvolution) block
19    a = Deconv3D(filters=gen_filters[0],

```

```

19         kernel_size=gen_kernel_sizes[0],
20         strides=gen_strides[0])(input_layer)
21     a = BatchNormalization()(a, training=True)
22     a = Activation(activation='relu')(a)
23
24     # Next 4 3D transpose convolution(or 3D deconvolution) blocks
25     for i in range(gen_convolutional_blocks - 1):
26         a = Deconv3D(filters=gen_filters[i + 1],
27                     kernel_size=gen_kernel_sizes[i + 1],
28                     strides=gen_strides[i + 1], padding='same')(
29             a)
30         a = BatchNormalization()(a, training=True)
31         a = Activation(activation=gen_activations[i + 1])(a)
32
33     gen_model = Model(inputs=[input_layer], outputs=[a])
34     return gen_model

```

Kode di atas berfungsi untuk membuat generator yaitu dengan ketentuan `gen` sebagai variabel dan membuat fungsi atau variabel `genmodel` lalu dilakukan `return`.

10. Soal 10

```

1  ### soal10
2
3
4  def build_discriminator():
5      """
6      Create a Discriminator Model using hyperparameters values
7      defined as follows
8      """
9
10     dis_input_shape = (64, 64, 64, 1)
11     dis_filters = [64, 128, 256, 512, 1]
12     dis_kernel_sizes = [4, 4, 4, 4, 4]
13     dis_strides = [2, 2, 2, 2, 1]
14     dis_paddings = ['same', 'same', 'same', 'same', 'valid']
15     dis_alphas = [0.2, 0.2, 0.2, 0.2, 0.2]
16     dis_activations = ['leaky_relu', 'leaky_relu', 'leaky_relu',
17                       'leaky_relu', 'sigmoid']
18     dis_convolutional_blocks = 5
19
20     dis_input_layer = Input(shape=dis_input_shape)
21
22     # The first 3D Convolutional block
23     a = Conv3D(filters=dis_filters[0],
24               kernel_size=dis_kernel_sizes[0],
25               strides=dis_strides[0],
26               padding=dis_paddings[0])(dis_input_layer)
27     # a = BatchNormalization()(a, training=True)
28     a = LeakyReLU(dis_alphas[0])(a)
29
30     # Next 4 3D Convolutional Blocks
31     for i in range(dis_convolutional_blocks - 1):
32         a = Conv3D(filters=dis_filters[i + 1],

```

```

32         kernel_size=dis_kernel_sizes[i + 1],
33         strides=dis_strides[i + 1],
34         padding=dis_paddings[i + 1])(a)
35     a = BatchNormalization()(a, training=True)
36     if dis_activations[i + 1] == 'leaky_relu':
37         a = LeakyReLU(dis_alphas[i + 1])(a)
38     elif dis_activations[i + 1] == 'sigmoid':
39         a = Activation(activation='sigmoid')(a)
40
41     dis_model = Model(inputs=[dis_input_layer], outputs=[a])
42     return dis_model

```

Kode di atas berfungsi untuk membangun diskriminator berfungsi untuk mendefinisikan seluruh gambar yang sudah di load generator sebagai gambar fake dan real.

11. Soal 11

Jika interpreter python menjalankan if name == main sebagai program utama, itu ialah menetapkan variabelan name untuk memiliki nilai main. Jika file ini sedang di impor dari modul lain, name akan ditetapkan ke nama modul. Nama modul tersedia sebagai nilai untuk variabel global.

12. Soal 12

```

1  #%% soal 12
2  object_name = "airplane"
3  data_dir = "data/3DShapeNets/volumetric_data/" \
4            "{}/30/train/*.mat".format(object_name)
5  gen_learning_rate = 0.0025
6  dis_learning_rate = 10e-5
7  beta = 0.5
8  batch_size = 1
9  z_size = 200
10 epochs = 1
11 MODE = "train"

```

Kode di atas berfungsi untuk melakukan load dataset dengan ketentuan data yang hanya dalam folder chair pada data train.

13. Soal 13

```

1  #%% soal 13
2  """
3  Create models
4  """
5  gen_optimizer = Adam(lr=gen_learning_rate, beta_1=beta)
6  dis_optimizer = Adam(lr=dis_learning_rate, beta_1=beta)
7
8  discriminator = build_discriminator()
9  discriminator.compile(loss='binary_crossentropy', optimizer=
10 discriminator)

```



```

11 generator = build_generator()
12 generator.compile(loss='binary_crossentropy', optimizer=
    gen_optimizer)

```

Kode di atas menggunakan Adam sebagai algoritma pengoptimalan dan `binary_crossentropy` sebagai kerugian loss.

14. Soal 14

```

1  ### soal14
2  discriminator.trainable = False
3
4  input_layer = Input(shape=(1, 1, 1, z_size))
5  generated_volumes = generator(input_layer)
6  validity = discriminator(generated_volumes)
7  adversarial_model = Model(inputs=[input_layer], outputs=[
    validity])
8  adversarial_model.compile(loss='binary_crossentropy',
    optimizer=gen_optimizer)

```

Kode di atas artinya ialah kita memasukkan random vector kedalam generator model lalu membagi 2 yaitu generated example dan real example, dan meneruskan ke diskriminator model sebagai real atau fake

15. Soal 15

```

1  ### soal15
2  print("Loading data ...")
3  volumes = get3DImages(data_dir=data_dir)
4  volumes = volumes[..., np.newaxis].astype(np.float)
5  print("Data loaded ...")

```

Kode di atas berfungsi untuk melakukan load data pada dataset.

16. Soal 16

```

1  ### soal16
2  tensorboard = TensorBoard(log_dir="logs/{}".format(time.time
    ()))
3  tensorboard.set_model(generator)
4  tensorboard.set_model(discriminator)

```

Kode di atas berfungsi untuk membuat tensorboard yang nantinya bisa diakses melalui localhost.

17. Soal 17

```

1  ### soal17
2  labels_real = np.reshape(np.ones((batch_size,)), (-1, 1, 1,
    1, 1))
3  labels_fake = np.reshape(np.zeros((batch_size,)), (-1, 1, 1,
    1, 1))

```

Kode di atas berfungsi untuk melakukan reshape agar shape yang dihasilkan tidak terlalu besar. Dengan membuat variabel real dan fake.

18. Soal 18

```
1 #%% soal18
2 if MODE == 'train':
3     for epoch in range(epochs):
4         print("Epoch:", epoch)
5
6         gen_losses = []
7         dis_losses = []
```

Kode di atas berfungsi untuk melakukan training epoch, karena jika epoch semakin banyak maka kualitas training yang dihasilkan akan semakin baik.

19. Soal 19

```
1 #%% soal19
2     number_of_batches = int(volumes.shape[0] / batch_size
3 )
4     print("Number of batches:", number_of_batches)
5     for index in range(number_of_batches):
6         print("Batch:", index + 1)
```

Batch adalah jumlah file yang akan di training.

20. Soal 20

```
1 #%% soal20
2     z_sample = np.random.normal(0, 0.33, size=[
3         batch_size, 1, 1, 1, z_size]).astype(np.float32)
4     volumes_batch = volumes[index * batch_size:(index
5         + 1) * batch_size, :, :, :]
```

Kode di atas berfungsi untuk membuat gambar bersih dari noise dan juga menyesuaikan shape.

21. Soal 21

```
1 #%% soal21
2     # Next, generate volumes using the generate
3     network
4     gen_volumes = generator.predict_on_batch(z_sample
5 )
```

Kode di atas berfungsi untuk membuat sample gambar palsu yang akan diteruskan ke diskriminator.

22. Soal 22

```

1  #%% soal22
2      discriminator.trainable = True
3      if index % 2 == 0:
4          loss_real = discriminator.train_on_batch(
volumes_batch, labels_real)
5          loss_fake = discriminator.train_on_batch(
gen_volumes, labels_fake)
6
7          d_loss = 0.5 * np.add(loss_real, loss_fake)
8          print("d_loss:{}".format(d_loss))
9
10         else:
11             d_loss = 0.0

```

Kode di atas berfungsi untuk membuat diskriminator bisa load gambar fake dan real dari generator, oleh karena itu ada generator loss dan diskriminator loss untuk melihat seberapa baik kualitas yang dihasilkan.

23. Soal 23

```

1  #%% soal23
2      discriminator.trainable = False
3      """
4      Train the generator network
5      """
6      z = np.random.normal(0, 0.33, size=[batch_size,
1, 1, 1, z_size]).astype(np.float32)
7      g_loss = adversarial_model.train_on_batch(z,
labels_real)
8      print("g_loss:{}".format(g_loss))
9
10     gen_losses.append(g_loss)
11     dis_losses.append(d_loss)

```

Kode di atas berfungsi untuk melakukan print gloss untuk generator dan juga dloss untuk diskriminator.

24. Soal 24

```

1  #%% soal24
2      # Every 10th mini-batch, generate volumes and
save them
3      if index % 10 == 0:
4          z_sample2 = np.random.normal(0, 0.33, size=[
batch_size, 1, 1, 1, z_size]).astype(np.float32)
5          generated_volumes = generator.predict(
z_sample2, verbose=3)
6          for i, generated_volume in enumerate(
generated_volumes[:5]):
7              voxels = np.squeeze(generated_volume)
8              voxels[voxels < 0.5] = 0.
9              voxels[voxels >= 0.5] = 1.
10             saveFromVoxels(voxels, "results/img-{}-{}
-{}".format(epoch, index, i))

```

Mengapa ada perulangan ? karena untuk melakukan perbandingan dari hasil yang sudah didapat.

25. Soal 25

```

1  #%% soal25
2      # Write losses to Tensorboard
3      write_log(tensorboard , 'g_loss' , np.mean(gen_losses) ,
4      epoch)
5      write_log(tensorboard , 'd_loss' , np.mean(dis_losses) ,
6      epoch)

```

TensorBoard adalah sebuah aplikasi web localhost untuk memeriksa dan menyelesaikan grafik dari hasil TensorFlow.

26. Soal 26

```

1  #%% soal26
2      generator.save_weights(os.path.join("models", "
3      generator_weights.h5"))
4      discriminator.save_weights(os.path.join("models", "
5      discriminator_weights.h5"))

```

File H5 adalah file data yang disimpan dalam Format Data Hirarki (HDF). Ini berisi array multidimensi data ilmiah.

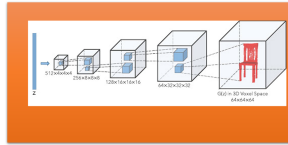
27. Soal 27

```

1  #%% soal27
2  if MODE == 'predict':
3      # Create models
4      generator = build_generator()
5      discriminator = build_discriminator()
6
7      # Load model weights
8      generator.load_weights(os.path.join("models", "
9      generator_weights.h5"), True)
10     discriminator.load_weights(os.path.join("models", "
11     discriminator_weights.h5"), True)
12
13     # Generate 3D models
14     z_sample = np.random.normal(0, 1, size=[batch_size, 1, 1,
15     1, z_size]).astype(np.float32)
16     generated_volumes = generator.predict(z_sample, verbose
17     =3)
18
19     for i, generated_volume in enumerate(generated_volumes
20     [:2]):
21         voxels = np.squeeze(generated_volume)
22         voxels[voxels < 0.5] = 0.
23         voxels[voxels >= 0.5] = 1.
24         saveFromVoxels(voxels, "results/gen_{}".format(i))

```

Ini adalah tahap akhir untuk melakukan testing dari model yang telah dibuat dan buat model dari yang sudah di create sebelumnya yaitu generator dan diskriminator. Untuk ilustrasi gambar sebagai berikut :



Gambar 8.26 Praktek Soal 27

8.2.3 Penanganan Error

1. ValueError

[illegible]

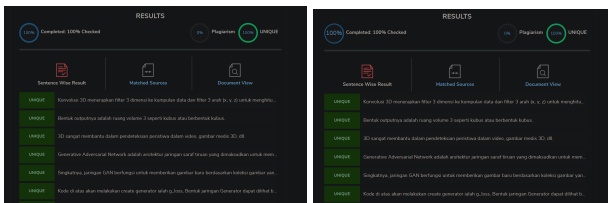
Gambar 8.27 ValueError

2. Cara Penanganan Error

- ValueError

Error tersebut karena disebabkan gagal load dataset karena salah penamaan direktori.

8.2.4 Bukti Tidak Plagiat



Gambar 8.28 Bukti Tidak Melakukan Plagiat Chapter 8

BAB 9

CHAPTER 9

BAB 10

CHAPTER 10

BAB 11

CHAPTER 11

BAB 12

CHAPTER 12

BAB 13

CHAPTER 13

BAB 14

CHAPTER 14

DAFTAR PUSTAKA

- [1] R. Awangga, “Sampeu: Servicing web map tile service over web map service to increase computation performance,” in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.

Index

disruptif, **xxiii**
modern, **xxiii**