

**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 Tunnisia

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. Sariasisih 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	89
10 Chapter 10	91
11 Chapter 11	93
12 Chapter 12	95
13 Chapter 13	97
14 Chapter 14	99



# DAFTAR ISI

---

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

<b>5</b>	<b>Chapter 5</b>	<b>9</b>
<b>6</b>	<b>Chapter 6</b>	<b>11</b>
<b>7</b>	<b>Chapter 7</b>	<b>13</b>
<b>8</b>	<b>Chapter 8</b>	<b>15</b>
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	1174027 - Harun Ar - Rasyid	30
8.2.1	Soal Teori	30
8.2.2	Praktek Program	33
8.2.3	Penanganan Error	43
8.2.4	Bukti Tidak Plagiat	44
8.3	1174017 - Muh. Rifky Prananda	44
8.3.1	Soal Teori	44
8.3.2	Praktek Program	48
8.3.3	Penanganan Error	58
8.3.4	Bukti Tidak Plagiat	59
8.4	1174096 - Nico Ekklesia Sembiring	59
8.4.1	Soal Teori	59
8.4.2	Praktek Program	64
8.4.3	Penanganan Error	74
8.4.4	Bukti Tidak Plagiat	75
8.5	1174008 - Arjun Yuda Firwanda	75
8.5.1	Teori	75
8.5.2	Praktek	79
8.5.3	Penanganan Error	87
8.5.4	Bukti Tidak Plagiat	87
<b>9</b>	<b>Chapter 9</b>	<b>89</b>
<b>10</b>	<b>Chapter 10</b>	<b>91</b>
<b>11</b>	<b>Chapter 11</b>	<b>93</b>

<b>12 Chapter 12</b>	<b>95</b>
<b>13 Chapter 13</b>	<b>97</b>
<b>14 Chapter 14</b>	<b>99</b>
Daftar Pustaka	101
Index	103



# **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

$B$  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].

$$ABC\mathcal{DEF}\alpha\beta\Gamma\Delta \sum_{def}^{abc} \quad (I.1)$$



# **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

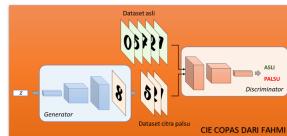
1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.  
Tugas Generator sekarang sedang dibuat untuk membuat koleksi gambar palsu, yang saat ini dilihat oleh Diskriminator. Diskriminator tidak dapat membedakan antara yang asli dan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

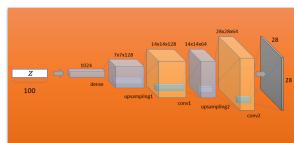
Diskriminator adalah CNN yang menerima input gambar yang dimiliki dan menghasilkan angka biner yang meminta input gambar, lalu menghasilkan gambar dari dataset asli atau menghasilkan gambar palsu. Untuk ilustrasi, lihat gambar berikut:



**Gambar 8.2** Teori 2

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

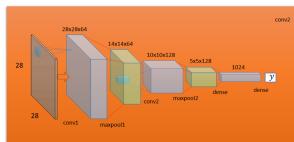
Aksitektur generator dibuat bisa dijelaskan pada gambar berikut :



**Gambar 8.3** Teori 3

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

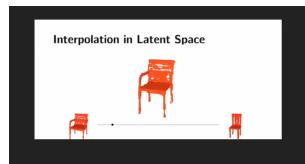
Aksitektur diskriminator dibuat bisa dijelaskan pada gambar berikut :



**Gambar 8.4** Teori 4

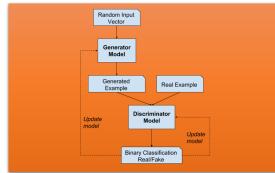
5. Jelaskan dengan ilustrasi gambar apa itu latent space.

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:/FOLDER KHUSUS NGANPUS/SEMESTER 8/src/tugosp/teori1.py', wdir='D:/FOLDER KHUSUS NGANPUS/SEMESTER 8/src/tugosp')
PERTAMA KEGAIATAN KE 8/AI PULL 8/src/174021/tugosp8
B1 matrix game with payoff matrices:
Row player:
[[3 0]
 [4 1]]
Column player:
[[3 4]
 [0 1]]
  
```

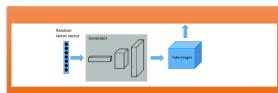
**Gambar 8.7** Teori 7

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 perhatikan 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).

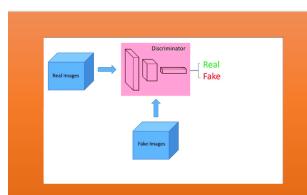
Untuk penjelasan tersebut dijelaskan pada gambar dibawah ini :



**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 diskriminatator tersebut. Jelaskan kenapa inputan dan outputan seperti itu.

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

### **8.1.2 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 pendekripsi 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
7     z_size = 200
8     gen_filters = [512, 256, 128, 64, 1]
9     gen_kernel_sizes = [4, 4, 4, 4, 4]
10    gen_strides = [1, 2, 2, 2, 2]
11    gen_input_shape = (1, 1, 1, z_size)
12    gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
13    gen_convolutional_blocks = 5
14
15    input_layer = Input(shape=gen_input_shape)
16
17    # First 3D transpose convolution(or 3D deconvolution) block
18    a = Deconv3D(filters=gen_filters[0],
19                  kernel_size=gen_kernel_sizes[0],
20                  strides=gen_strides[0])(input_layer)
```

```

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')(a)
27         a = BatchNormalization()(a, training=True)
28         a = Activation(activation=gen_activations[i + 1])(a)
29
30     gen_model = Model(inputs=[input_layer], outputs=[a])
31     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

```

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

#### 4. Soal 4

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

#### 5. Soal 5

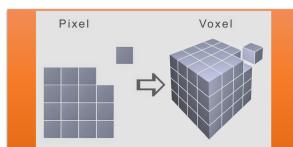
- Clone github
- Download dataset
- Buat folder baru logs dan results

#### 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

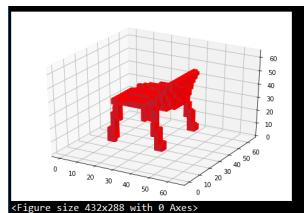
## 8. Soal 8

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun May 10 18:10:55 2020
4
5 @author: FAHMI-PC
6
7 Ciee 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 import numpy as np
17 voxels = np.pad(voxels, (1, 1), 'constant', constant_values=(0,
18     0))
19 #%%%
20 import scipy.ndimage as nd
21 voxels = nd.zoom(voxels, (2, 2, 2), mode='constant', order=0)
22 #%%%
23 import matplotlib.pyplot as plt
24 from mpl_toolkits.mplot3d import Axes3D
25 fig = plt.figure()
26 ax = Axes3D(fig)
27 ax.voxels(voxels, edgecolor="red")
28 plt.show()
29 plt.savefig('data')

```

Kode di atas befungsi untuk visualisasidataset 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')(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 befungsi untuk membuat generator yaitu dengan ketentuan gen sebagai variabel dan membuat fungsi atau variabel genmodel lalu dilakukan return.

## 10. Soal 10

```

1 %% soall10
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 befungsi untuk membangun diskriminasi 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 befungsi 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=[validity])
8 adversarial_model.compile(loss='binary_crossentropy',
9   optimizer=gen_optimizer)

```

Kode di atas artinya ialah kita memasukkan random vector kedalam generator model lalu bagi 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 befungsi 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,
3         1))
3     labels_fake = np.reshape(np.zeros((batch_size,)), (-1, 1, 1,
4         1))

```

Kode di atas befungsi 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 befungsi 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=[batch_size, 1, 1, 1, z_size]).astype(np.float32)
3             volumes_batch = volumes[index * batch_size:(index + 1) * batch_size, :, :, :]

```

Kode di atas befungsi 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 befungsi 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(
5                 volumes_batch, labels_real)
6                 loss_fake = discriminator.train_on_batch(
7                     gen_volumes, labels_fake)
8
9         d_loss = 0.5 * np.add(loss_real, loss_fake)
10        print("d_loss:{}".format(d_loss))
11
12     else:
13         d_loss = 0.0
```

Kode di atas befungsi untuk membuat diskriminatator bisa load gambar fake dan real dari generator, oleh karena itu ada generator loss dan diskriminatator 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,
7 1, 1, 1, z_size]).astype(np.float32)
8 g_loss = adversarial_model.train_on_batch(z,
9 labels_real)
10 print("g_loss:{}".format(g_loss))
11
12 gen_losses.append(g_loss)
13 dis_losses.append(d_loss)
```

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

## 24. Soal 24

```

1 #%% soal24
2             # Every 10th mini-batch, generate volumes and
3             save them
4             if index % 10 == 0:
5                 z_sample2 = np.random.normal(0, 0.33, size=[batch_size, 1, 1, 1, z_size]).astype(np.float32)
6                 generated_volumes = generator.predict(z_sample2, verbose=3)
7                 for i, generated_volume in enumerate(generated_volumes[:5]):
8                     voxels = np.squeeze(generated_volume)
9                     voxels[voxels < 0.5] = 0.
10                    voxels[voxels >= 0.5] = 1.
11                    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", "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()

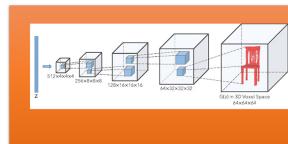
```

```

6      # Load model weights
7      generator.load_weights(os.path.join("models", "generator_weights.h5"), True)
8      discriminator.load_weights(os.path.join("models", "discriminator_weights.h5"), True)
9
10     # Generate 3D models
11     z_sample = np.random.normal(0, 1, size=[batch_size, 1, 1,
12                                     1, z_size]).astype(np.float32)
13     generated_volumes = generator.predict(z_sample, verbose
14 =3)
15
16     for i, generated_volume in enumerate(generated_volumes
17 [:2]):
18         voxels = np.squeeze(generated_volume)
19         voxels[voxels < 0.5] = 0.
20         voxels[voxels >= 0.5] = 1.
21         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 8.12** Praktek Soal 27

### **8.1.3 Penanganan Error**

## 1. ValueError

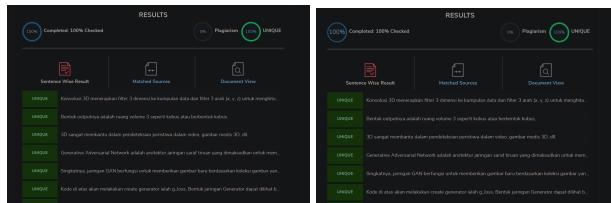
**Gambar 8.13** ValueError

## 2. Cara Penanganan Error

- **ValueError**

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

### 8.1.4 Bukti Tidak Plagiat



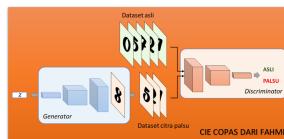
**Gambar 8.14** Bukti Tidak Melakukan Plagiat Chapter 8

## 8.2 1174027 - Harun Ar - Rasyid

### 8.2.1 Soal Teori

1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.

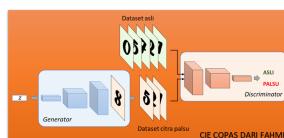
Tugas Generator sekarang sedang dibuat untuk membuat koleksi gambar palsu, yang saat ini dilihat oleh Diskriminator. Diskriminator tidak dapat membedakan antara yang asli dan yang palsu. Untuk ilustrasi, lihat gambar berikut:



**Gambar 8.15** Teori 1

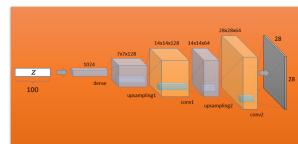
2. Jelaskan dengan ilustrasi gambar sendiri apa itu diskriminator dengan perumpamaan dosen anda sebagai diskriminatornya.

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:



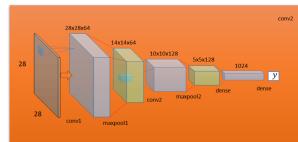
**Gambar 8.16** Teori 2

3. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur generator dibuat.  
 Aksitektur generator dibuat bisa dijelaskan pada gambar berikut :



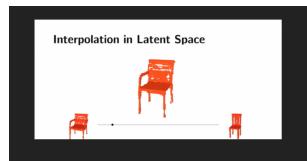
**Gambar 8.17** Teori 3

4. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur diskriminatator dibuat.  
 Aksitektur diskriminatator dibuat bisa dijelaskan pada gambar berikut :



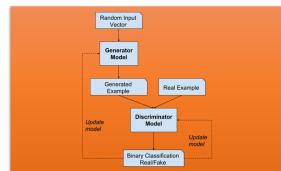
**Gambar 8.18** Teori 4

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 KHUSUS NGAPUS/SEMESTER 6/8/scr/174021/tupgs8/theori.py', wdir='D:/FOLDER KHUSUS NGAPUS/SEMESTER 6/8/scr/174021/tupgs8')
B1 matrix game with payoff matrices:
```

Row player:  
 $\begin{bmatrix} 3 & 0 \\ 4 & 1 \end{bmatrix}$

Column player:  
 $\begin{bmatrix} 3 & 4 \\ 0 & 1 \end{bmatrix}$

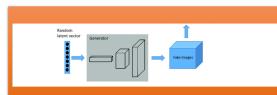
**Gambar 8.21** Teori 7

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).

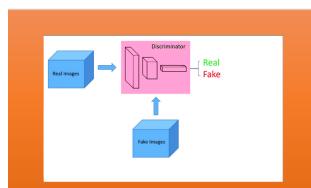
Untuk penjelasan tersebut dijelaskan pada gambar dibawah ini :



**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 diskriminatator tersebut. Jelaskan kenapa inputan dan outputan seperti itu.  
Gambar dari Generator yang berhasil di deteksi oleh Diskriminatator sebagai gambar fake, akan dikembalikan dengan feedback pke generator. Kini Generator bertugas untuk bisa membuat sekumpulan gambar palsu, yang nantinya dapat dilihat oleh Diskriminatator, lalu, Diskriminatator 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 pendekripsi 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

```
| 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')(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

```

```

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

```

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

#### 4. Soal 4

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

#### 5. Soal 5

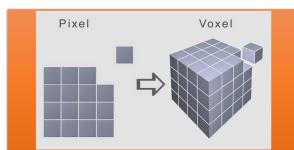
- Clone github
- Download dataset
- Buat folder baru logs dan results

#### 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 :



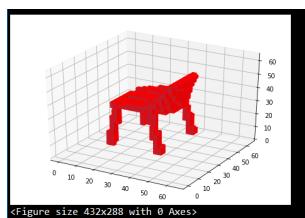
**Gambar 8.24** Praktek 7

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

Kode di atas befungsi untuk visualisasidataset 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 Praktek 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
9     z_size = 200
10    gen_filters = [512, 256, 128, 64, 1]
11    gen_kernel_sizes = [4, 4, 4, 4, 4]
12    gen_strides = [1, 2, 2, 2, 2]
13    gen_input_shape = (1, 1, 1, z_size)
14    gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
15    gen_convolutional_blocks = 5
16
17    input_layer = Input(shape=gen_input_shape)
18
19    # First 3D transpose convolution(or 3D deconvolution) block
20    a = Deconv3D(filters=gen_filters[0],
21                  kernel_size=gen_kernel_sizes[0],
22                  strides=gen_strides[0])(input_layer)
23    a = BatchNormalization()(a, training=True)
24    a = Activation(activation='relu')(a)
25
26    # Next 4 3D transpose convolution(or 3D deconvolution) blocks
27    for i in range(gen_convolutional_blocks - 1):
28        a = Deconv3D(filters=gen_filters[i + 1],
29                      kernel_size=gen_kernel_sizes[i + 1],
30                      strides=gen_strides[i + 1], padding='same')(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 befungsi untuk membuat generator yaitu dengan ketentuan gen sebagai variabel dan membuat fungsi atau variabel genmodel lalu dilakukan return.

## 10. Soal 10

```

1 #%% soall10
2
3
4 def build_discriminator():
5     """
6         Create a Discriminator Model using hyperparameters values
7         defined as follows
8     """
9
9     dis_input_shape = (64, 64, 64, 1)
10    dis_filters = [64, 128, 256, 512, 1]
11    dis_kernel_sizes = [4, 4, 4, 4, 4]
12    dis_strides = [2, 2, 2, 2, 1]
13    dis_paddings = ['same', 'same', 'same', 'same', 'valid']
14    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 befungsi untuk membangun diskriminator berfungsi untuk mendefenisikan 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 befungsi 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 diskriminatior 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 befungsi 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,
3     1))
3     labels_fake = np.reshape(np.zeros((batch_size,)), (-1, 1, 1,
4     1))

```

Kode di atas befungsi 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 befungsi 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=[batch_size, 1, 1, 1, z_size]).astype(np.float32)
3             volumes_batch = volumes[index * batch_size:(index + 1) * batch_size, :, :, :]

```

Kode di atas befungsi 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)

```

Kode di atas befungsi 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(
5                     volumes_batch, labels_real)
6                     loss_fake = discriminator.train_on_batch(
7                         gen_volumes, labels_fake)
8
9                     d_loss = 0.5 * np.add(loss_real, loss_fake)
10                    print("d_loss:{}".format(d_loss))
11

```

Kode di atas befungsi 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
3
4
5
6
7
8
9
10
11
discriminator.trainable = False
"""
Train the generator network
"""
z = np.random.normal(0, 0.33, size=[batch_size,
1, 1, 1, z_size]).astype(np.float32)
g_loss = adversarial_model.train_on_batch(z,
labels_real)
print("g_loss:{}".format(g_loss))

gen_losses.append(g_loss)
dis_losses.append(d_loss)

```

Kode di atas befungsi untuk melakukan print gloss untuk generator dan juga dloss untuk diskriminatot.

#### 24. Soal 24

```

1 #%% soal24
2
3
4
5
6
7
8
9
10
# Every 10th mini-batch, generate volumes and
# save them
if index % 10 == 0:
    z_sample2 = np.random.normal(0, 0.33, size=[batch_size,
1, 1, 1, z_size]).astype(np.float32)
    generated_volumes = generator.predict(
z_sample2, verbose=3)
    for i, generated_volume in enumerate(
generated_volumes[:5]):
        voxels = np.squeeze(generated_volume)
        voxels[voxels < 0.5] = 0.
        voxels[voxels >= 0.5] = 1.
        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
3
4
# Write losses to Tensorboard
write_log(tensorboard, 'g_loss', np.mean(gen_losses),
epoch)
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.

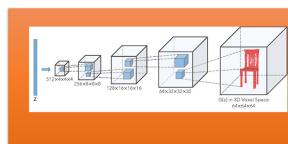
## 27. Soal 27

```

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



**Gambar 8.26** Praktek 27

### 8.2.3 Penanganan Error

#### 1. ValueError

```

execfile(filename, namespaces)

def _get_filelist(directory, filelist, namespaces):
    """Recursively walk through directory, collecting filenames
    in filelist, and adding them to namespaces under their
    relative path.
    """
    for name in os.listdir(directory):
        if os.path.isdir(os.path.join(directory, name)):
            _get_filelist(os.path.join(directory, name), filelist, namespaces)
        else:
            filelist.append(os.path.join(directory, name))

def _get_all_files(directory, namespaces):
    """Get all files in directory, in getlist order.
    """
    filelist = []
    _get_filelist(directory, filelist, namespaces)
    all_files = os.pathsep.join(filelist)
    size_mb = len(all_files) / (1024 * 1024)
    return all_files, size_mb

def _get_file_size(filename):
    """Get size of file in bytes.
    """
    return os.path.getsize(filename)

def _get_file_md5(filename):
    """Get MD5 hash of file.
    """
    md5sum = hashlib.md5()
    with open(filename, "rb") as f:
        for chunk in iter(lambda: f.read(4096), b""):
            md5sum.update(chunk)
    return md5sum.hexdigest()

def _get_file_mtime(filename):
    """Get modification time of file.
    """
    return os.path.getmtime(filename)

def _get_file_permissions(filename):
    """Get file permissions in octal format.
    """
    return oct(stat.S_IMODE(os.stat(filename).st_mode))

def _get_file_size_mb(filename):
    """Get file size in MB.
    """
    return os.path.getsize(filename) / (1024 * 1024)

def _get_file_md5_hex(filename):
    """Get MD5 hash of file in hex format.
    """
    return _get_file_md5(filename).encode("hex")

def _get_file_mtime_hex(filename):
    """Get modification time of file in hex format.
    """
    return _get_file_mtime(filename).encode("hex")

def _get_file_permissions_hex(filename):
    """Get file permissions in hex format.
    """
    return _get_file_permissions(filename).encode("hex")

def _get_file_size_hex(filename):
    """Get file size in hex format.
    """
    return _get_file_size_mb(filename).encode("hex")

```

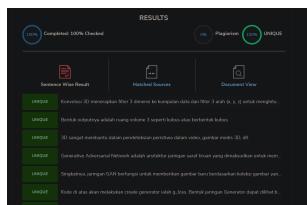
**Gambar 8.27** Value error

## 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

8.3 1174017 - Muh. Rifky Prananda

### 8.3.1 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 8.29** Teori 1

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

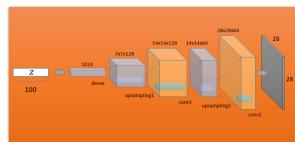
Diskriminator adalah CNN yang menerima input gambar yang dimiliki dan menghasilkan angka biner yang meminta input gambar, lalu menghasilkan gambar dari dataset asli atau menghasilkan gambar palsu. Untuk ilustrasi, lihat gambar berikut:



**Gambar 8.30** Teori 2

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

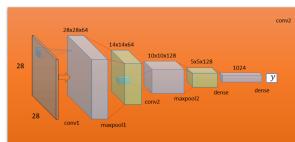
Aksitektur generator dibuat bisa dijelaskan pada gambar berikut :



**Gambar 8.31** Teori 3

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

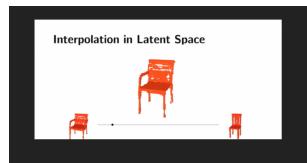
Aksitektur diskriminatator dibuat bisa dijelaskan pada gambar berikut :



**Gambar 8.32** Teori 4

5. Jelaskan dengan ilustrasi gambar apa itu latent space.

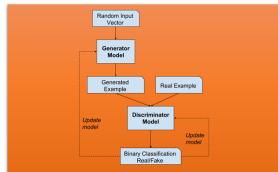
Latent space dijelaskan pada gambar berikut :



**Gambar 8.33** Teori 5

6. Jelaskan dengan ilustrasi gambar apa itu adversarial play.

Adversarial play dijelaskan pada gambar berikut :



**Gambar 8.34** 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 KHSUS NGANPUS/SEMESTER 8/src/174021/tugas8/teori1.py', wdir='D:/FOLDER KHSUS NGANPUS/SEMESTER 8/src/174021/tugas8')
          PERTAMA KEGAIATAN KE 8/AJ PULL 8/src/174021/tugas8
B1: matrix game with payoff matrices:

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

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

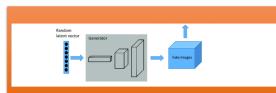
**Gambar 8.35** Teori 7

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 perhatikan 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).

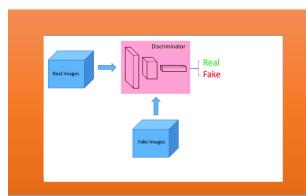
Untuk penjelasan tersebut dijelaskan pada gambar dibawah ini :



**Gambar 8.36** 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 8.37** Teori 10

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

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

### 8.3.2 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 pendekripsi 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
7     z_size = 200
8     gen_filters = [512, 256, 128, 64, 1]
9     gen_kernel_sizes = [4, 4, 4, 4, 4]
10    gen_strides = [1, 2, 2, 2, 2]
11    gen_input_shape = (1, 1, 1, z_size)
12    gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
13    gen_convolutional_blocks = 5
14
15    input_layer = Input(shape=gen_input_shape)
16
17    # First 3D transpose convolution (or 3D deconvolution) block
18    a = Deconv3D(filters=gen_filters[0],
19                  kernel_size=gen_kernel_sizes[0],
20                  strides=gen_strides[0])(input_layer)
```

```

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')(a)
27
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

```

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

#### 4. Soal 4

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

#### 5. Soal 5

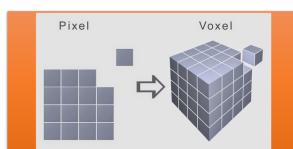
- Clone github
- Download dataset
- Buat folder baru logs dan results

#### 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 8.38** Praktek Soal 7

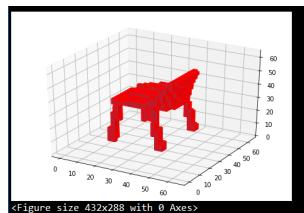
## 8. Soal 8

```

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/
14     test/chair_000000000_1.mat")['instance']
15 #%%
16 import numpy as np
17 voxels = np.pad(voxels, (1, 1), 'constant', constant_values=(0,
18     0))
19 #%%
20 import scipy.ndimage as nd
21 voxels = nd.zoom(voxels, (2, 2, 2), mode='constant', order=0)
22 #%%
23 import matplotlib.pyplot as plt
24 from mpl_toolkits.mplot3d import Axes3D
25 fig = plt.figure()
26 ax = Axes3D(fig)
27 ax.voxels(voxels, edgecolor="red")
28 plt.show()
29 plt.savefig('data')

```

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



**Gambar 8.39** 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')(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 befungsi untuk membuat generator yaitu dengan ketentuan gen sebagai variabel dan membuat fungsi atau variabel genmodel lalu dilakukan return.

## 10. Soal 10

```

1 %% soall10
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 befungsi untuk membangun diskriminasi 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 befungsi 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=[validity])
8 adversarial_model.compile(loss='binary_crossentropy',
9   optimizer=gen_optimizer)

```

Kode di atas artinya ialah kita memasukkan random vector kedalam generator model lalu bagi 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 befungsi 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,
3         1))
3     labels_fake = np.reshape(np.zeros((batch_size,)), (-1, 1, 1,
4         1))

```

Kode di atas befungsi 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 befungsi 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=[batch_size, 1, 1, 1, z_size]).astype(np.float32)
3             volumes_batch = volumes[index * batch_size:(index + 1) * batch_size, :, :, :]

```

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

22. Soal 22

```
1 #%% soal22
2
3     discriminator.trainable = True
4
5     if index % 2 == 0:
6         loss_real = discriminator.train_on_batch(
7             volumes_batch, labels_real)
8         loss_fake = discriminator.train_on_batch(
9             gen_volumes, labels_fake)
10
11     d_loss = 0.5 * np.add(loss_real, loss_fake)
12     print("d_loss:{}".format(d_loss))
13
14 else:
15     d_loss = 0.0
```

Kode di atas befungsi untuk membuat diskriminatator bisa load gambar fake dan real dari generator, oleh karena itu ada generator loss dan diskriminatator 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,
7 1, 1, 1, z_size]).astype(np.float32)
8 g_loss = adversarial_model.train_on_batch(z,
9 labels_real)
10 print("g_loss:{}\n".format(g_loss))
11
12 gen_losses.append(g_loss)
13 dis_losses.append(d_loss)
```

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

## 24. Soal 24

```

1 #%% soal24
2             # Every 10th mini-batch, generate volumes and
3             save them
4             if index % 10 == 0:
5                 z_sample2 = np.random.normal(0, 0.33, size=[batch_size, 1, 1, 1, z_size]).astype(np.float32)
6                 generated_volumes = generator.predict(z_sample2, verbose=3)
7                 for i, generated_volume in enumerate(generated_volumes[:5]):
8                     voxels = np.squeeze(generated_volume)
9                     voxels[voxels < 0.5] = 0.
10                    voxels[voxels >= 0.5] = 1.
11                    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", "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()

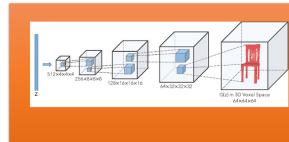
```

```

6      # Load model weights
7      generator.load_weights(os.path.join("models", "generator_weights.h5"), True)
8      discriminator.load_weights(os.path.join("models", "discriminator_weights.h5"), True)
9
10     # Generate 3D models
11     z_sample = np.random.normal(0, 1, size=[batch_size, 1, 1,
12                                     1, z_size]).astype(np.float32)
13     generated_volumes = generator.predict(z_sample, verbose
14 =3)
15
16     for i, generated_volume in enumerate(generated_volumes
17 [:2]):
18         voxels = np.squeeze(generated_volume)
19         voxels[voxels < 0.5] = 0.
20         voxels[voxels >= 0.5] = 1.
21         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 8.40** Praktek Soal 27

### 8.3.3 Penanganan Error

## 1. ValueError

```

execfile(filename, namespace)

#File 'filenames' contains a list of file names, one in each line
execfile('filenames', filename='wacc', namespace)

#File 'filenames' contains a list of file names, one in each line
#with the extension .dat
#The following command reads all files in 'filenames' and stores them in 'values'
#values = getallfiles(filenames, dir=datadir)

#File 'filenames' contains a list of file names, one in each line
#with the extension .dat
#The following command reads all files in 'filenames' and stores them in 'values'
#values = readonddatafilelist(filenames, datadir, size=1000)

#File 'filenames' contains a list of file names, one in each line
#with the extension .dat
#The following command reads all files in 'filenames' and stores them in 'values'
#values = readonddatafilelist(filenames, datadir, size=1000, readatstart=True, readatend=True)

```

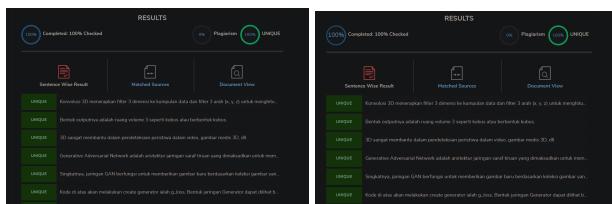
**Gambar 8.41** ValueError

## 2. Cara Penanganan Error

- **ValueError**

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

### 8.3.4 Bukti Tidak Plagiat



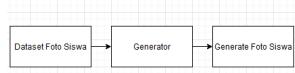
**Gambar 8.42** Bukti Tidak Melakukan Plagiat Chapter 8

## 8.4 1174096 - Nico Ekklesia Sembiring

### 8.4.1 Soal Teori

1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.

Generator merupakan sebuah metode atau proses untuk menghasilkan data baru berdasarkan dengan data sebelumnya atau data yang ada. Biasanya output yang diminta haruslah sejenis sehingga misalkan dari Gambar seorang mahasiswa tidak bisa dijadikan output nilai dari mahasiswa yang berbentuk teks. Untuk generator sendiri, proses latihan bergantung pada seberapa sering input diterima dan perbandingan data yang asli dengan data yang dibuat oleh generator. Misalkan diberikan foto Mahasiswa 1 dijadikan patokan untuk output. Maka setiap input yang diterima akan dibandingkan dengan data yang sebenarnya. Semakin sering dilakukan generate atau pelatihan data, maka akurasi semakin tinggi.

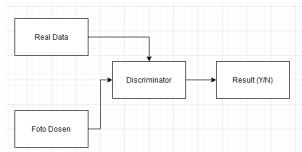


**Gambar 8.43** Ilustrasi generator

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

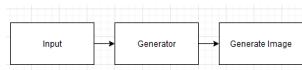
Discriminator merupakan pemisah antara data asli dengan data yang dihasilkan oleh generator. Discriminator akan menyimpan data yang ada ke kategori yang sudah didefinisikan sebelumnya. Discriminator sendiri akan membandingkan data input dengan pembanding sehingga dapat diketahui jika input itu adalah data palsu. Namun, jika terlalu sering dilakukan pelatihan generator maka discriminator sendiri akan berkurang akurasinya. Contohnya seperti misalkan Data dosen 1 dijadikan patokan dan input selalu dilakukan. Semakin sering dilatih,

maka seseorang yang hampir mirip dengan dosen tersebut akan dapat diidentifikasi sebagai data asli oleh discriminator.



**Gambar 8.44** ilustrasi diskriminator

3. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur generator dibuat  
Arsitektur generator yang ada terbentuk dengan neural network sederhana yang terdiri dari beberapa layer. Untuk layernya sendiri ada layer input untuk menerima data. alu Dense layer atau layer proses untuk melakukan proses yang ada. Dan layer reshape untuk membentuk kembali data yang sudah diproses. Jumlah dari layer proses sendiri tergantung dari spesifikasi data yang ada.



**Gambar 8.45** Ilustrasi pembuatan arsitektur generator

4. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur diskriminator dibuat  
Arsitektur Discriminator terdapat beberapa layer, diantaranya yaitu input layer untuk menerima data dari generator, output layer untuk mengindikasikan jika data yang diproduksi oleh generator yaitu asli atau tidak, dan layer proses untuk membandingkan data asli dengan data yang dihasilkan oleh generator.



**Gambar 8.46** ilustrasi pembuatan arsitektur diskriminator

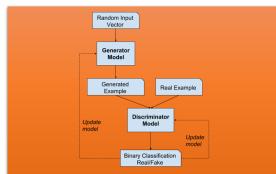
5. Jelaskan dengan ilustrasi gambar apa itu latent space.  
Latent space merupakan sebuah model yang menjabarkan isi dari data yang berbentuk media atau data yang tidak dikenali komputer, Seperti gambar, musik, dan lain sebagainya.



**Gambar 8.47** Ilustrasi latent space

6. Jelaskan dengan ilustrasi gambar apa itu adversarial play

Adversarial play adalah dimana para jaringan di latih, dimana jaringan satu dan lainnya saling berkompetisi. dapat disimpulkan dimana jaringan generator dan jaringan discriminator saling bertemu berulang ulang kali.



**Gambar 8.48** Ilustrasi adversarial play

7. Jelaskan dengan ilustrasi gambar apa itu Nash equilibrium

Nash equilibrium adalah konsep dalam teori permainan di mana hasil optimal dari permainan adalah di mana tidak ada insentif untuk menyimpang dari strategi awal mereka. Lebih khusus lagi, keseimbangan Nash adalah konsep teori permainan di mana hasil optimal dari permainan adalah di mana tidak ada pemain yang memiliki insentif untuk menyimpang dari strategi yang dipilihnya setelah mempertimbangkan pilihan lawan.

	A	B
A	1,1	1,-1
B	-1,1	0,0

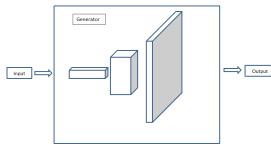
**Gambar 8.49** ilustrasi nash equilibrium

8. Sebutkan dan jelaskan contoh-contoh implementasi dari GAN

Pada bidang mode, seni, dan iklan, GAN dapat digunakan untuk membuat foto-foto model fashion imajinier tanpa perlu menyewa model. Pada bidang sains GAN dapat meningkatkan citra astronomi dan mensimulasikan pelensaan gravitasi untuk penelitian materi gelap.

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)

```
gen=Sequential() #Inisiasi dari sequensial
gen.add(Dense(units=200,input_dim=np.shape(train_in))
gen.add(Dense(units=400))#Menambah dense layer dengan
gen.add(Dense(units=784, activation='tanh')) #Menambah
gen.compile(loss='binary_crossentropy', optimizer=ad
gen.summary() #Memproses data yang sudah disetting c
```



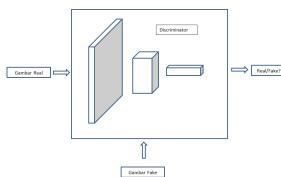
**Gambar 8.50** arsitektur membuat generator

Pada contoh tersebut, data akan diambil dari hasil proses sebelumnya yaitu proses ekstrasi data gambar. Dari contoh ini, terdapat 3 layer dense, 1 input layer dan 1 output layer. Dari input layer nanti akan dimasukkan terlebih dahulu ke dense layer pertama lalu diproses oleh 2 layer selanjutnya dan terakhir akan ditampilkan oleh layer output.

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

```
diskrim=Sequential() #Inisiasi dari sequensial
diskrim.add(Dense(units=784, input_dim=np.shape(train_in)
diskrim.add(Dense(units=400)) #Mensetting Dense
diskrim.add(Dense(units=200, activation='sigmoid'))
diskrim.compile(loss='binary_crossentropy', optimizer=ad
diskrim.summary()#Memproses data yang sudah disetting c
```

Pada contoh tersebut, data akan diambil dari hasil proses sebelumnya yaitu proses ekstrasi data gambar. Dari contoh ini, terdapat 3 layer dense, 1 input layer dan 1 output layer. Pada proses ini, seluruh data akan dibandingkan dengan data sebelumnya yaitu dari generator dan dari data aslinya yang sudah dijadikan data vector.



**Gambar 8.51** arsitektur diskriminator

11. Jelaskan bagaimana kaitan output dan input antara generator dan diskriminator tersebut. Jelaskan kenapa inputan dan outputan seperti itu.  
Pada kedua metode tersebut, akan disebutkan berapa akurasi dari setiap metode. Pada setiap metode tersebut (Discriminator dan generator) akan dilakukan pelatihan dan akan dibandingkan hasilnya. Generator akan menghasilkan data baru sesuai dengan hasil latihan dan dari data tersebut, discriminator akan membandingkan dengan data set apakah data tersebut "asli" atau tidak.
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.  
Relative entropy adalah ukuran dari bagaimana satu distribusi probabilitas berbeda dari yang kedua, distribusi probabilitas referensi, Divergensi Jensen-Shannon adalah ukuran divergensi berprinsip yang selalu terbatas untuk variabel acak terbatas.
13. Jelaskan apa itu fungsi objektif yang berfungsi untuk mengukur kesamaan antara gambar yang dibuat dengan yang asli.  
Fungsi objektif adalah fungsi yang digunakan sebagai penujuk berapa nilai kesamaan antara gambar yang dibuat dengan yang asli.
14. Jelaskan apa itu scoring algoritma selain mean square error atau cross entropy seperti The Inception Score dan The Frechet Inception distance.  
Inception Score digunakan untuk mengukur seberapa realistik output dari GAN, dimana ada dua parameter, yaitu : gambarnya punya variasi dan setiap gambar jelas terlihat seperti sesuatu. Frechet Inception Distance adalah ukuran kesamaan antara dua dataset gambar. Itu terbukti berkorelasi baik dengan penilaian manusia terhadap kualitas visual dan paling sering digunakan untuk mengevaluasi kualitas sampel Generative Adversarial Networks. FID dihitung dengan menghitung jarak Fr'echet antara dua Gaussians dipasang ke representasi fitur dari jaringan Inception.
15. Jelaskan kelebihan dan kekurangan GAN
  - GAN Menghasilkan data baru yang bisa hampir mirip dengan data asli. Karena hasil pelatihannya, GAN dapat menghasilkan data gambar, teks,

audio, dan video yang dapat dibilang hampir mirip dengan yang aslinya. Berkat hal tersebut, GAN dapat digunakan dalam sistem marketing, e-commerce, games, iklan, dan industri lainnya

- GAN mempelajari representasi data secara internal sehingga beberapa masalah pada machine learning dapat diatasi dengan mudah
- Discriminator yang sudah dilatih dapat menjadi sebuah classifier atau pen-deteksi jika data sudah sesuai. Karena Discriminator yang akan menjadi tidak efisien berkat seringnya dilatih
- GAN dapat dilatih menggunakan data yang belum dilabeled

Kerugian :

- Data saat diproses oleh metode gan tidak konvergensi
- Jenis sampel yang dihasilkan oleh generator terbatas karena modenya ter-batas
- Ketidak seimbangnya antara generator dan discriminator dapat menyebabkan overfitting atau terlalu dekat dengan hasil sampel
- Sangat sensitif dengan data yang sudah diinisiasi sebelumnya

#### 8.4.2 Praktek Program

##### 1. Jelaskan apa itu 3D convolutions

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 pendekripsi peristiwa dalam video,gambarmedis 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. Jelaskan dengan kode program arsitektur dari generator networknya, beserta penjelasan input dan output dari generator network.

```

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]
```

```

9   gen_input_shape = (1, 1, 1, z_size)
10  gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
11  gen_convolutional_blocks = 5
12
13  input_layer = Input(shape=gen_input_shape)
14
15  # First 3D transpose convolution(or 3D deconvolution) block
16  a = Deconv3D(filters=gen_filters[0],
17                kernel_size=gen_kernel_sizes[0],
18                strides=gen_strides[0])(input_layer)
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')(a)
27      a = BatchNormalization()(a, training=True)
28      a = Activation(activation=gen_activations[i + 1])(a)
29
30  gen_model = Model(inputs=[input_layer], outputs=[a])
31  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.

- Jelaskan dengan kode program arsitektur dari diskriminator network, beserta penjelasan input dan outputnya.

```

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], padding='same')(dis_input_layer)
23    a = LeakyReLU(alpha=dis_alphas[0])(a)
24
25    for i in range(dis_convolutional_blocks - 1):
26        a = Conv3D(filters=dis_filters[i + 1],
27                   kernel_size=dis_kernel_sizes[i + 1],
28                   strides=dis_strides[i + 1], padding='same')(a)
29        a = LeakyReLU(alpha=dis_alphas[i + 1])(a)
30
31    dis_model = Model(inputs=[dis_input_layer], outputs=[a])
32    return dis_model

```

```

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

```

Diskriminato adalah d loss, Jaringan Diskriminato 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. Diskriminato dilatih dengan dataset yang diambil dari Generator, lalu di training untuk membedakan keduanya. Gambar dari Generator yang berhasil di deteksi oleh Diskriminato sebagai gambar fake, akan dikembalikan dengan feedback pke generator. Kini Generator bertugas untuk bisa membuat sekumpulan gambar palsu, yang nantinya dapat dilihat oleh Diskriminato, lalu, Diskriminato tidak bisa membedakan fake dan real.

#### 4. Jelaskan proses training 3D-GANs

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

#### 5. Jelaskan bagaimana melakukan settingan awal chapter 02 untuk memenuhi semua kebutuhan sebelum melanjutkan ke tahapan persiapan data.

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

#### 6. Jelaskan tentang dataset yang digunakan, dari mulai tempat unduh, cara membuka dan melihat data. Sampai deskripsi dari isi dataset dengan detail penjelasan setiap folder/file yang membuat orang awam paham.

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. Jelaskan apa itu voxel dengan ilustrasi dan bahasa paling awam

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.

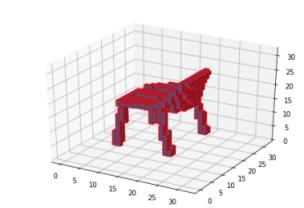
- Visualisasikan dataset tersebut dalam tampilan visual plot, jelaskan cara melakukan visualisasinya

```

1 # In[ Visualisasi data]
2 import scipy.io as io
3 voxels = io.loadmat("data/3DShapeNets/volumetric_data/chair/30/
4                         test/chair_000000000_1.mat")['instance']
5
6 #%%
7 import numpy as np
8 voxels = np.pad(voxels, (1, 1), 'constant', constant_values=(0,
9                     0))
10
11 #%%
12 import scipy.ndimage as nd
13 voxels = np.zoom(voxels, (2, 2, 2), mode='constant', order=0)
14
15 #%%
16 import matplotlib.pyplot as plt
17 from mpl_toolkits.mplot3d import Axes3D
18 fig = plt.figure()
19 ax = Axes3D(fig)
20 ax.voxels(voxels, edgecolor="red")
21 plt.show()
22 plt.savefig('data')

```

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



**Gambar 8.52** Hasil Soal 8.

- buka file run.py jelaskan perbaris kode pada fungsi untuk membuat generator yaitu build generator.

```

1 def build_generator():
2     """
3         Create a Generator Model with hyperparameters values defined
4             as follows

```

```

4 """
5 z_size = 200
6 gen_filters = [512, 256, 128, 64, 1]
7 gen_kernel_sizes = [4, 4, 4, 4, 4]
8 gen_strides = [1, 2, 2, 2, 2]
9 gen_input_shape = (1, 1, 1, z_size)
10 gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
11 gen_convolutional_blocks = 5
12
13 input_layer = Input(shape=gen_input_shape)
14
15 # First 3D transpose convolution(or 3D deconvolution) block
16 a = Deconv3D(filters=gen_filters[0],
17               kernel_size=gen_kernel_sizes[0],
18               strides=gen_strides[0])(input_layer)
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')(a)
27     a = BatchNormalization()(a, training=True)
28     a = Activation(activation=gen_activations[i + 1])(a)
29
30 gen_model = Model(inputs=[input_layer], outputs=[a])
31 return gen_model

```

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

10. jelaskan juga fungsi untuk membangun diskriminator pada fungsi build discriminator.

```

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 # 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

```

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

11. jelaskan apa maksud dari kode program name == ' main '

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. jelaskan secara detil perbaris dan per parameter apa arti dari kode program :

```

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

```

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

13. Jelaskan secara detil dari kode program pembuatan dan kompilasi arsitektur berikut :

```

1     gen_optimizer = Adam(lr=gen_learning_rate, beta_1=beta)
2     dis_optimizer = Adam(lr=dis_learning_rate, beta_1=beta)

```

```

3     discriminator = build_discriminator()
4     discriminator.compile(loss='binary_crossentropy', optimizer=
5         dis_optimizer)
6
7     generator = build_generator()
8     generator.compile(loss='binary_crossentropy', optimizer=
9         gen_optimizer)

```

Kode di atas menggunakan Adam sebagai algoritma pengoptimalan dan binary\_crossentropy sebagai kerugian loss.

14. Jelaskan secara detil kode program untuk membuat dan melakukan kompilasi model adversarial berikut:

```

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

```

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

15. Jelaskan Ekstrak dan load data kursi dengan menggunakan fungsi getVoxels-Format dan get3DImages yang digunakan pada kode program berikut :

```

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

```

Kode di atas befungsi untuk melakukan load data pada dataset.

16. Jelaskan maksud dari kode program instansiasi TensorBoard yang menambahkan generator dan diskriminator pada program berikut:

```

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

```

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

17. Jelaskan apa fungsi dari np reshape ones zeros pada kode program berikut dengan parameteranya:

```

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

```

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

18. Jelaskan kenapa harus ada perulangan dalam meraih epoch. Dan jelaskan apa itu epoch terkait kode program berikut:

```

1     for epoch in range(epochs):
2         print("Epoch:", epoch)
3
4     gen_losses = []
5     dis_losses = []

```

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

19. Jelaskan apa itu batches dan kaitannya dengan kode program berikut, dan kenapa berada di dalam epoch:

```

1             number_of_batches = int(volumes.shape[0] / batch_size
2             )
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. Berikut adalah kode program pengambilan gambar dan noise. Jelaskan apa fungsi np.random.normal serta astype, serta jelaskan apa arti parameter titik dua dan jelaskan isi dari z sample dan volumes batch:

```

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

```

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

21. Berikut adalah kode program generator gambar palsu. Jelaskan apa fungsi generator.predict on batch, serta jelaskan apa arti parameter z sample:

```

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

```

Kode di atas befungsi untuk membuat sample gambar palsu yang akan diteruskan ke diskriminatot.

22. Berikut adalah kode program training diskriminator dengan gambar palsu dari generator dan gambar asil. Jelaskan apa maksudnya harus dilakukan training diskriminator secara demikian dan jelaskan apa isi loss fake dan loss real serta d loss dan fungsi train on batch.

```

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

```

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

23. Berikut adalah kode program training model adversarial yang terdapat generator dan diskriminator. Jelaskan apa bagaimana proses terbentuknya parameter z dan g loss:

```

1     z = np.random.normal(0, 0.33, size=[batch_size,
2     1, 1, 1, z_size]).astype(np.float32)
3     g_loss = adversarial_model.train_on_batch(z,
4     labels_real)
5     print("g_loss:{}".format(g_loss))
6
7     gen_losses.append(g_loss)
8     dis_losses.append(d_loss)

```

Kode di atas befungsi untuk melakukan print gloss untuk generator dan juga dloss untuk diskriminatot.

24. Berikut adalah kode generate dan menyimpan gambar 3D setelah beberapa saat setiap epoch. Jelaskan mengapa ada perulangan dengan parameter tersebut, serta jelaskan arti setiap variabel beserta perlihatkan isinya dan artikan isinya :

```

1          # Every 10th mini-batch, generate volumes and
2          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.
-{}".format(epoch, index, i))

```

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

25. Berikut adalah kode program menyimpan average losses setiap epoch. Jelaskan apa itu tensorboard dan setiap parameter yang digunakan pada kode program ini :

```

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

```

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

26. Berikut adalah kode program menyimpan model. Jelaskan apa itu format h5 dan penjelasan dari kode program berikut :

```

1      generator.save_weights(os.path.join("models", "generator_weights.h5"))
2      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. Berikut adalah kode program testing model. Jelaskan dengan ilustrasi gambar dari mulai meload hingga membuat gambar 3D dengan menggunakan z sample, bisakah parameter z sample tersebut diubah? :

```

1          # Create models
2          generator = build_generator()
3          discriminator = build_discriminator()
4
5          # Load model weights

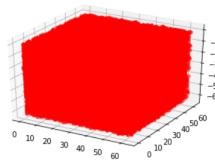
```

```

6     generator.load_weights(os.path.join("models", "generator_weights.h5"), True)
7     discriminator.load_weights(os.path.join("models", "discriminator_weights.h5"), True)
8
9     # Generate 3D models
10    z_sample = np.random.normal(0, 1, size=[batch_size, 1, 1,
11        z_size]).astype(np.float32)
12    generated_volumes = generator.predict(z_sample, verbose
13        =3)
14
15    for i, generated_volume in enumerate(generated_volumes
16        [:2]):
17        voxels = np.squeeze(generated_volume)
18        voxels[voxels < 0.5] = 0.
19        voxels[voxels >= 0.5] = 1.
20        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 diskriminat. Untuk ilustrasi gambar sebagai berikut :



**Gambar 8.53** Hasil Soal 27.

#### 8.4.3 Penanganan Error

1. `AttributeError: 'Model' object has no attribute 'get_distribution_strategy'`

```

In [15]: runfile('D:/ot/ch8/run.py', wdir='D:/ot/ch8')
Data loaded...
Traceback (most recent call last):
  File "C:\Users\Otto\PycharmProjects\ch8\run.py", line 11, in <module>
    tensorboard.set_model(generator)
  File "C:\Users\Otto\PycharmProjects\ch8\tensorboard\models.py", line 11, in set_model
    super(Tensorboard, self).set_model(model)
  File "C:\Users\Otto\PycharmProjects\ch8\tensorboard\models.py", line 532, in set_model
    self.log_dir, self.model.get_distribution_strategy())
  File "C:\Users\Otto\PycharmProjects\ch8\models.py", line 532, in get_distribution_strategy
    pyprint(disable=protected_access)
AttributeError: 'Model' object has no attribute '_get_distribution_strategy'

```

**Gambar 8.54** `AttributeError`

2. Tuliskan Kode Error dan Jenis Error

- `AttributeError`

### 3. Cara Penangan Error

- **AttributeError**

Error terjadi karena versi tensorflow yang digunakan tidak sama dengan versi tensorflow yang disarankan oleh program sehingga tidak terdapat atribut untuk membangun model. Cara mengatasinya adalah dengan menginstall versi tensorflow yang sama dengan versi yang dibutuhkan oleh program.

#### 8.4.4 Bukti Tidak Plagiat



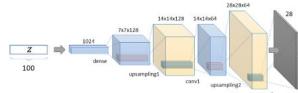
**Gambar 8.55** Bukti Tidak Melakukan Plagiat Chapter 8

### 8.5 1174008 - Arjun Yuda Firwanda

#### 8.5.1 Teori

1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.

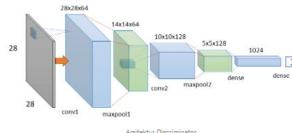
Generator merupakan generator yang biasanya menggunakan data yang ada untuk menghasilkan data baru, serta generator sendiri bertujuan untuk menghasilkan data berupa video,gambar, audio dan teks.



**Gambar 8.56** Generator

2. Jelaskan dengan ilustrasi gambar sendiri apa itu diskriminator dengan Dosen sebagai Diskriminator.

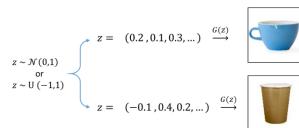
Diskriminator merupakan untuk membedakan antara data nyata dan data yang dihasilkan oleh generator. pada jaringan diskriminator sendiri mencoba memasukan data ke dalam kategori yang sudah ditentukan.



**Gambar 8.57** Diskriminasi

3. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur generator dibuat

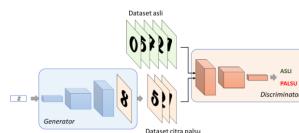
Dilihat berkebalikan dengan struktur jaringan saraf pada umumnya. Pada Generator dinyatakan menerima input vektor  $z$  dan kemudian mengubahnya menjadi gambar tiga dimensi.



**Gambar 8.58** Arsitektuk Generator

4. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur diskriminasi dibuat

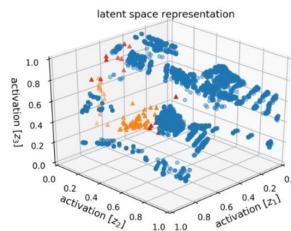
Diskriminasi merupakan jaringan klasifikasi biner yang menerima input gambar tiga dimensi dan mengeluarkan klasifikasi gambar asli. Diskriminasi dilatih dengan sekumpulan data, dan sekumpulan dataset, dan dilatih untuk bisa membedakan keduanya.



**Gambar 8.59** Arsitektur Diskriminasi

5. Jelaskan dengan ilustrasi gambar apa itu latent space.

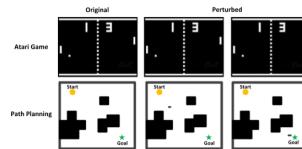
Latent space merupakan vektor angka yang dihasilkan secara acak.



**Gambar 8.60** Arsitektur Diskriminatör

6. Jelaskan dengan ilustrasi gambar apa itu adversarial play

Adversarial Play, berbagi data pelestarian privasi, dan vaksin rancangan. Saya menjelaskan bagaimana aspek konseptual kedua dari game keamanan menawarkan pemodelan alami paradigm untuk ini.



**Gambar 8.61** Adversarial Play

7. Jelaskan dengan ilustrasi gambar apa itu Nash equilibrium

Nash Equilibrium sendiri menggambarkan keadaan tertentu dalam teori permainannya. Sehingga dalam permainan non kooperatif ini dimana setiap pemainnya memiliki strategi untuk hasil yang terbaik berdasarkan apa yang mereka harapkan apa yang dilakukan oleh pemain lainnya.

		Perusahaan B	
		Diferensiasi	Homogen
Perusahaan A	Diferensiasi	100	200
	Homogen	0	200

**Gambar 8.62** Nash equilibrium

8. Sebutkan dan jelaskan contoh-contoh implementasi dari GAN

Cara membuat GAN dengan citra beresolusi tinggi. Kita mulai dengan data sederhana yang jauh lebih mudah, yaitu dataset citra angka tulisan tangan MNIST. Membangun GAN dengan library Keras dan Tensorflow.

```

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential, load_model
from keras.layers import Dense, Conv2D, Flatten
from keras.layers import BatchNormalization, Dropout
from keras.layers import UpSampling4, UpSampling2D
from keras.layers import MaxPooling2D, Activation
from keras.datasets import mnist
from keras.optimizers import SGD
from keras.utils import to_categorical
from PIL import Image

```

**Gambar 8.63** GAN

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

Inputan seed 1x100. Generator akan mengubah menjadi sebuah gambar 28x28 yang menggunakan Convolutional Neural Network.

```

def generator_model():
    model = Sequential([
        Dense(1024, input_dim=100, activation='tanh'),
        Dense(128*7*7),
        Reshape((7, 7, 128)),
        UpSampling2D(size=(2, 2)),
        Conv2D(64, (5, 5), padding='same', activation='tanh'),
        UpSampling2D(size=(2, 2)),
        Conv2D(1, (5, 5), padding='same', activation='tanh')
    ])
    return model

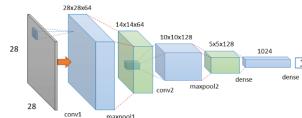
generator_model().summary()

```

**Gambar 8.64** Hasil Arsitektuk Generator

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

Diskriminatator merupakan CNN yang menerima input 28,28 dan menghasilkan angka biner, jika kelas 1 maka gambar asli, jika kelas 0 maka gambar palsu.

**Gambar 8.65** Hasil Arsitektur Generator

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

Generator, Di sini kami tetapkan ukuran input seed adalah 1x100. Diskriminatator, yang menerima input 28,28 dan menghasilkan angka biner, jika kelas 1 maka gambar asli, jika kelas 0 maka gambar palsu.

12. Jelaskan apa perbedaan antara KullbackLeibler divergence KL divergence relative entropy, Jensen-Shannon JS divergence information radius iRaD total divergence to the average dalam mengukur kualitas dari model.

Kullback-Leibler divergence untuk menghitung skor yang mengukur divergensi satu distribusi probabilitas dari yang lain. Jensen-Shannon divergence metode untuk mengukur kesamaan antara dua distribusi probabilitas. Ia juga dikenal sebagai radius informasi IRad atau total divergensi ke rata-rata.

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

Fungsi obyektif digunakan untuk membuat jaringan gerator yang menghasilkan gambar yang mirip dengan gambar yang aslinya, serta meningkatkan kesamaan data yang dihasilkan oleh generator ke data aslinya.

$$\min_{\theta_G} \max_{\theta_D} v(\theta_G, \theta_D) = \mathbf{E}_{x, y \sim p_{\text{data}}} [\log D(x, y)] \\ + \mathbf{E}_{z \sim p_z(z), \tilde{y} \sim p_y} [\log (1 - D(G(z, \tilde{y}), \tilde{y}))] \quad (1)$$

### Gambar 8.66 Hasil Arsitektuk Generator

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

The Inception Score memperkirakan kualitas koleksi gambar sintetis berdasarkan seberapa baik model klasifikasi gambar berkinerja terbaik Inception v3 mengklasifikasikannya sebagai salah satu dari 1.000 objek yang dikenal. Skor menggabungkan kepercayaan prediksi kelas bersyarat untuk setiap gambar sintetis kualitas dan integral dari probabilitas marginal dari kelas prediksi keragaman. The Frechet Inception Distance meringkas jarak antara vektor fitur Inception untuk gambar nyata dan yang dihasilkan dalam domain yang sama.

- Jelaskan kelebihan dan kekurangan GAN

Kelebihan GAN yang pertama dapat dijadikan sebagai pembelajaran tanpa pengawasan, kemudian yang kedua gan menghasilkan data yang mirip dengan aslinya, dan selanjutnya yang ketiga sebagai belajar distribusi kepadatan data.

Kekurangannya sulit dilatih, tidak stabil, serta terdapat masalah pada mode tutup.

## 8.5.2 Praktek

- Jelaskan apa itu 3D convolutions Konvolusi 3D menerapkan filter 3 dimensi ke dataset dan filter bergerak 3 arah x y z untuk menghitung representasi fitur level rendah. Bentuk output mereka adalah ruang volume 3 dimensi seperti kubus.
- Jelaskan dengan kode program arsitektur dari generator networknya, beserta penjelasan input dan output dari generator network. Pada kode program ini, menyebutkan seed atau ketetapan ukuran input. Yakni 1x100 dan mengubah outputnya menjadi 28x28 yang menggunakan Convolution Neural Network.

```

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

```

3. Jelaskan dengan kode program arsitektur dari diskriminator network, beserta penjelasan input dan outputnya. Pada program ini, input gambar berupa 28x28 dan menghasilkan angka biner yang menyatakan bahwa kelas 1 merupakan data asli, dan jika kelas 0 maka gambar palsu.

```

1 def build_generator():
2     """
3         Create a Generator Model with hyperparameters values defined
4         as follows
5         """
6
7     z_size = 200
8     gen_filters = [512, 256, 128, 64, 1]
9     gen_kernel_sizes = [4, 4, 4, 4, 4]
10    gen_strides = [1, 2, 2, 2, 2]
11    gen_input_shape = (1, 1, 1, z_size)

```

```

10     gen_activations = ['relu', 'relu', 'relu', 'relu', 'sigmoid']
11     gen_convolutional_blocks = 5
12
13     input_layer = Input(shape=gen_input_shape)
14
15     # First 3D transpose convolution(or 3D deconvolution) block
16     a = Deconv3D(filters=gen_filters[0],
17                   kernel_size=gen_kernel_sizes[0],
18                   strides=gen_strides[0])(input_layer)
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')(a)
27         a = BatchNormalization()(a, training=True)
28         a = Activation(activation=gen_activations[i + 1])(a)
29
30     gen_model = Model(inputs=[input_layer], outputs=[a])
31     return gen_model

```

4. Jelaskan proses training 3D-GANs Proses Training disini melatih data gambar menggunakan Generator Bangkitkan data citra palsu menggunakan Generator sejumlah dataset citra asli. Latih Discriminator untuk bisa membedakan dataset citra asli dari dataset citra palsu. Gunakan Discriminator yang sudah dilatih untuk melatih Generator agar bisa membangkitkan dataset citra palsu yang dinilai asli oleh Discriminator.
5. Jelaskan bagaimana melakukan settingan awal chapter 02 untuk memenuhi semua kebutuhan sebelum melanjutkan ke tahapan persiapan data. load dataset file csv. menghasilkan label biner lulus gagal berdasarkan G1 G2 G3 nilai tes, masing-masing 0 20 poin ambang batas untuk kelulusan adalah jumlah 30. gunakan one-hot encodin pada kolom kategorikal. gunakan shuffle row. menggunakan decision tree. menggunakan visualisasi pohon.
6. Jelaskan tentang dataset yang digunakan, dari mulai tempat unduh, cara membuka dan melihat data. Sampai deskripsi dari isi dataset dengan detail penjelasan setiap folder yang membuat orang awam paham. Isi dari dataset 3DShapeNets ada 6 folder. Dan masing-masing folder terdapat isi yang berbeda. Folder 3D untuk gambar 3D Folder bp untuk mendeskripsikan dan meneruskan program. Folder generative untuk mengenerate data gambar. Folder util untuk input json. Folder volumetric data untuk mengetahui metrik volume data. Folder voxelization untuk menampilkan hasil gambar berupa plot.
7. Jelaskan apa itu voxel dengan ilustrasi dan bahasa paling awam Voxel diibaratkan seperti pixel di bidang tiga dimensi, memiliki panjang, lebar dan tinggi. Volume pixel atau voxel adalah titik dalam ruang tiga dimensi. Sebuah voxel mendefin-

isikan posisi dengan tiga koordinat dalam arah x, y, dan z. Sebuah voxel adalah unit dasar untuk mewakili gambar 3D

8. Visualisasikan dataset tersebut dalam tampilan visual plot, jelaskan cara melakukan visualisasinya import library, load data file .mat, lalu read memakai matplotlib.
  9. buka file run.py jelaskan perbaris kode pada fungsi untuk membuat generator yaitu build generator. Membuat generator yaitu dengan ketentuan gen sebagai variabel dan membuat fungsi atau variabel gen model lalu dilakukan return
  10. jelaskan juga fungsi untuk membangun diskriminator pada fungsi build discriminator. Membangun diskriminator berfungsi untuk mendefenisikan seluruh gambar yang sudah di load generator sebagai gambar fake dan real
  11. jelaskan apa maksud dari kode program name main itu menetapkan beberapa variabel khusus seperti name itu mengeksekusi semua kode yang ditemukan dalam file. Jika interpreter python menjalankan if name main sebagai program utama, itu ialah menetapkan variabel name untuk memiliki nilai main. Jika file ini sedang diimpor dari modul lain, name akan ditetapkan ke nama modul. Nama modul tersedia sebagai nilai untuk name variabel global.

```
| if __name__ == '__main__':
|     ...
|         saveFromVoxels(voxels,
|             Loading data...
|             Data loaded...
|             Epoch: 0
|             Number of batches: 10
|             Batch: 1
|                 d_loss:0.6931722164154053
|                 g_loss:0.6931138038635254
|             Batch: 2
|                 g_loss:0.6931138038635254
|             Batch: 3
|                 d_loss:0.6931579113006592
|                 g_loss:0.6931090354919434
|             Batch: 4
|                 g_loss:0.6931090354919434
|             Batch: 5
|                 d_loss:0.6931560039520264
|                 g_loss:0.693108081817627
```

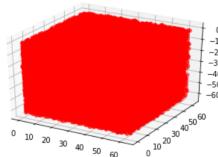
Gambar 8.67 File

12. jelaskan secara detil perbaris dan per parameter apa arti dari kode program :  
Artinya adalah load dataset yang hanya dalam folder chair data train

```

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

```

**Gambar 8.68** Parameter

13. Jelaskan secara detil dari kode program pembuatan dan kompilasi arsitektur berikut Disini menggunakan Adam sebagai algoritma pengoptimalan dan binary crossentropy sebagai kerugian loss.

```

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

```

**Gambar 8.69** Kompilasi Arsitektur

14. Jelaskan secara detil kode program untuk membuat dan melakukan kompilasi model adversarial berikut Ini artinya ialah kita memasukkan random vector kedalam generator model lalu bagi 2 yaitu generated example dan real example, dan meneruskan ke diskriminator model sebagai real atau fake.

```

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

```

Name	Date modified	Type	Size
discriminator_weights.h5	5/8/2020 5:27 PM	HS File	43,209 KB
generator_weights.h5	5/8/2020 5:27 PM	HS File	68,686 KB

**Gambar 8.70** Kompilasi Model Adversarial

15. Jelaskan Ekstrak dan load data kursi dengan menggunakan fungsi getVoxels-Format dan get3DImages yang digunakan pada kode program berikut : Ini melakukan load data pada dataset.

```

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

```

16. Jelaskan maksud dari kode program instansiasi TensorBoard yang menambahkan generator dan diskriminator pada program berikut: Ini berfungsi untuk membuat tensorflow yang nantinya bisa diakses melalui localhost

```

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

```

17. Jelaskan apa fungsi dari np.reshape ones zeros pada kode program berikut dengan parameternya: Fungsi ini ialah untuk melakukan reshape agar shape yang dihasilkan tidak terlalu besar.

```

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

```

18. Jelaskan kenapa harus ada perulangan dalam meraih epoch. Dan jelaskan apa itu epoch terkait kode program berikut: Karena jika epoch semakin banyak maka kualitas training yang dihasilkan akan semakin baik

```

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

```

19. Jelaskan apa itu batches dan kaitannya dengan kode program berikut, dan kenapa berada di dalam epoch: Batch adalah jumlah file yang akan di training

```

1
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)

```

20. Berikut adalah kode program pengambilan gambar dan noise. Jelaskan apa fungsi np.random.normal serta astype, serta jelaskan apa arti parameter titik dua dan jelaskan isi dari z sample dan volumes batch: Ini adalah untuk membuat gambar bersih dari noise dan juga menyesuaikan shape

```

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

```

21. Berikut adalah kode program generator gambar palsu. Jelaskan apa fungsi generator.predict\_on\_batch, serta jelaskan apa arti parameter z sample: Ialah membuat sample gambar palsu yang akan diteruskan ke diskriminator.

```

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

```

22. Berikut adalah kode program training diskriminator dengan gambar palsu dari generator dan gambar asil. Jelaskan apa maksudnya harus dilakukan training diskriminator secara demikian dan jelaskan apa isi loss fake dan loss real serta d loss dan fungsi train\_on\_batch. 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.

```

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

```

23. Berikut adalah kode program training model adversarial yang terdapat generator dan diskriminator. Jelaskan apa bagaimana proses terbentuknya parameter z dan g loss: Dengan melakukan print g loss untuk generator dan juga d loss untuk diskriminator

```

1         z = np.random.normal(0, 0.33, size=[batch_size,
2             1, 1, 1, z_size]).astype(np.float32)
3             g_loss = adversarial_model.train_on_batch(z,
4                 labels_real)
5                 print("g_loss:{}".format(g_loss))
6
7                 gen_losses.append(g_loss)
8                 dis_losses.append(d_loss)

```

24. Berikut adalah kode program generate dan menyimpan gambar 3D setelah beberapa saat setiap epoch. Jelaskan mengapa ada perulangan dengan parameter

tersebut, serta jelaskan arti setiap variabel beserta perlihatkan isinya dan artikan isinya: Mengapa ada perulangan ? karena untuk melakukan perbandingan dari hasil yang sudah didapat.

```

1           # Every 10th mini-batch, generate volumes and
2           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))

```

25. Berikut adalah kode program menyimpan average losses setiap epoch. Jelaskan apa itu tensorboard dan setiap parameter yang digunakan pada kode program ini: TensorBoard adalah sebuah aplikasi web localhost untuk memeriksa dan menyelesaikan grafik dari hasil TensorFlow.

```

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

```

26. Berikut adalah kode program menyimpan model. Jelaskan apa itu format h5 dan penjelasan dari kode program berikut: File H5 adalah file data yang disimpan dalam Format Data Hirarki. Ini berisi array multidimensi data ilmiah.

```

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

```

27. Berikut adalah kode program testing model. Jelaskan dengan ilustrasi gambar dari mulai meload hingga membuat gambar 3D dengan menggunakan z sample, bisakah parameter z sample tersebut diubah2: 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.

```

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

```

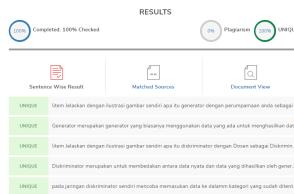
```

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

```

### 8.5.3 Penanganan Error

### 8.5.4 Bukti Tidak Plagiat



**Gambar 8.71**   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, [xxv](#)  
modern, [xxv](#)