

CERDAS MENGUASAI GIT

CERDAS MENGUASAI GIT

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

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

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

CONTRIBUTORS

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

CONTENTS IN BRIEF

1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
5 Chapter 5	9
6 Chapter 6	11
7 Chapter 7	13
8 Chapter 8	15
9 Chapter 9	29
10 Chapter 10	31
11 Chapter 11	33
12 Chapter 12	35
13 Chapter 13	37
14 Chapter 14	39

DAFTAR ISI

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

5	Chapter 5	9
6	Chapter 6	11
7	Chapter 7	13
8	Chapter 8	15
8.1	1174006 - Kadek Diva Krishna Murti	15
8.1.1	Teori	15
8.1.2	Praktek	20
8.1.3	Penanganan Error	28
9	Chapter 9	29
10	Chapter 10	31
11	Chapter 11	33
12	Chapter 12	35
13	Chapter 13	37
14	Chapter 14	39
	Daftar Pustaka	41

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

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

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

R. M. A.

ACRONYMS

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

GLOSSARY

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

SYMBOLS

A Amplitude

$\&$ Propositional logic symbol

a Filter Coefficient

\mathcal{B} Number of Beats

INTRODUCTION

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

Informatics Research Center
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[1].

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

BAB 1

CHAPTER 1

BAB 2

CHAPTER 2

BAB 3

CHAPTER 3

BAB 4

CHAPTER 4

BAB 5

CHAPTER 5

BAB 6

CHAPTER 6

BAB 7

CHAPTER 7

BAB 8

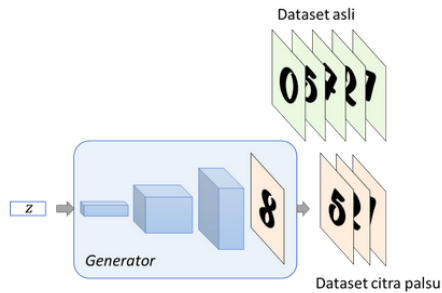
CHAPTER 8

8.1 1174006 - Kadek Diva Krishna Murti

8.1.1 Teori

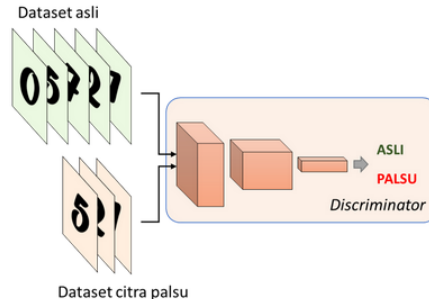
1. Jelaskan dengan ilustrasi gambar sendiri apa itu generator dengan perumpamaan anda sebagai mahasiswa sebagai generatornya.

Generator merupakan jaringan yang menggunakan data yang ada (seperti gambar, video, audio, atau teks) untuk dijadikan data yang baru. Misalnya menggunakan gambar yang sudah ada untuk menghasilkan gambar yang baru. Tujuan utama generator adalah untuk menghasilkan data (seperti gambar, video, audio, atau teks) dari vektor angka yang dihasilkan secara acak. Perumpamaannya saya sebagai mahasiswa atau generator mengubah beberapa baris kodingan teman yang sudah ada. Sehingga menghasilkan kodingan yang baru.

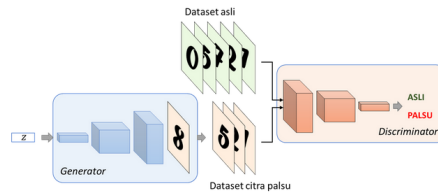


2. Jelaskan dengan ilustrasi gambar sendiri apa itu diskriminator dengan perumpamaan dosen anda sebagai diskriminatornya.

Diskriminator merupakan jaringan yang mencoba membedakan antara data yang ada dengan data baru yang dihasilkan oleh generator. Diskriminator mencoba memasukkan data inputan (data yang sudah ada) ke dalam kategori yang telah ditentukan. Perumpamaannya dosen atau diskriminator mencoba membedakan kodingan yang dibuat oleh saya dengan teman saya untuk menemukan kodingan yang mana yang sudah ada dan kodingan yang hasil perubahan.



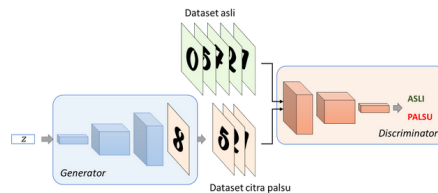
3. Jelaskan dengan ilustrasi gambar sendiri bagaimana arsitektur generator dibuat. Arsitektur generator:
- Jaringan generator mencoba membuat data yang terlihat seperti data yang nyata dari data yang sudah ada.
 - Ulangi langkah pertama hingga beberapa iterasi untuk mengecoh diskriminator dengan membuat data yang senyata mungkin.
 - Hingga akhirnya, diskriminator melatih generator hingga ia tidak bisa lagi membedakan data nyata dan data yang palsu.



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

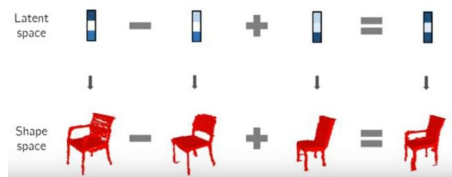
Arsitektur diskriminator:

- Jaringan diskriminator mencoba mengidentifikasi apakah data tersebut asli atau palsu.
- Ulangi langkah pertama hingga beberapa iterasi untuk mengecoh generator dengan memperbaiki kriterianya untuk menentukan palsu tidaknya.
- Hingga akhirnya, diskriminator melatih generator hingga ia tidak bisa lagi membedakan data nyata dan data yang palsu.



5. Jelaskan dengan ilustrasi gambar apa itu latent space.

Latent space adalah data yang dibuat dari beberapa vektor secara acak.



- Jelaskan dengan ilustrasi gambar apa itu adversarial play.
Adversarial play adalah proses pelatihan dimana jaringan discriminator dan jaringan generator berusaha saling memebodohi satu sama lain.
- Jelaskan dengan ilustrasi gambar apa itu Nash equilibrium.
Nash Equilibrium adalah keadaan dimana jaringan discriminator tidak dapat membedakan lagi data asli dan data palsu.

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

- (a) Image Generation, dimana GAN digunakan untuk mengenerate data baru dari perpaduan data asli yang ada.
- (b) Text to image synthesis, dimana GAN digunakan untuk mengenerate gambar berdasarkan deskripsi teks yang ada.
- (c) Face aging, dimana GAN digunakan untuk mengenerate beberapa gambar wajah berbeda usia.
- (d) Image to image translation, dimana GAN digunakan untuk mengenerate gambar baru dari gambar yang sudah ada disertai dengan penambahan style.
- (e) Video synthesis, dimana GAN digunakan untuk mengenerate sebuah video.
- (f) High resolution image generation, dimana GAN digunakan untuk meningkatkan resolusi atau mempertajam gambar.
- (g) Completing missing parts of images, dimana GAN digunakan untuk menambal bagian gambar yang hilang.

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).

Input Layer

Pada input layer diambil sampel vektor 100 dimensi dan dilempar ke hidden layer pertama berupa tensor.

```
1 input_shape=(batch_size , 100), output_shape=(batch_size , 100)
2
```

Hidden Layer

Pada hidden layer akan ada proses konversi dari tensor yang dilempar dari layer sebelumnya. Pada hidden layer terdiri dari beberapa dense layer dengan beberapa unit juga.

```
1 neurons=500, input_shape=(batch_size , 100), output_shape=(
2 batch_size , 500)
```

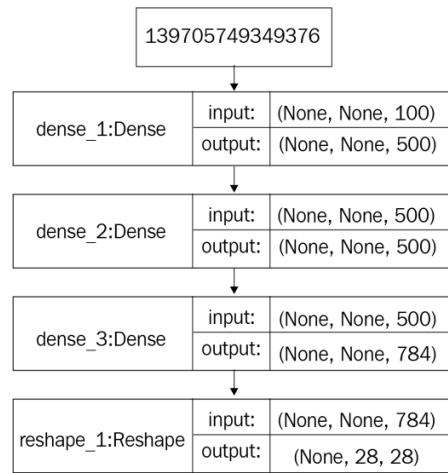
```
1 neurons=500, input_shape=(batch_size , 500), output_shape=(
2 batch_size , 500)
```

```
1 neurons=784, input_shape=(batch_size , 500), output_shape=(
2 batch_size , 784)
```

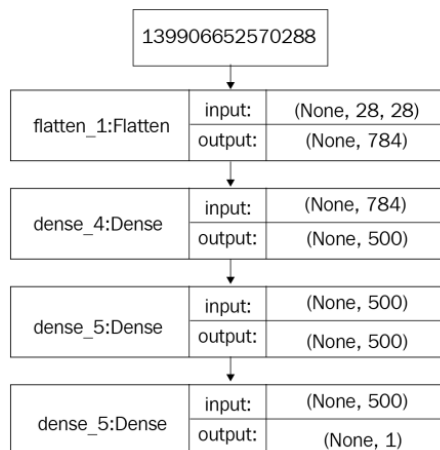
Output Layer

Pada output layer akan digenerate gambar yang memiliki bentuk 28 x 28 unit.

```
1 input_shape=(batch_size , 784), output_shape=(batch_size , 28,
2 28)
```



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



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

Inputan pada jaringan generator akan menerima input dari gambar asli, kemudian akan diolah oleh jaringan generator untuk dibuatkan gambar palsunya. Setelah dibuat gambar palsunya, kemudian akan diolah oleh jaringan diskriminator untuk dibedakan apakah gambar yang diolah sebelumnya oleh jaringan generator adalah asli atau palsu. Setelah diolah oleh jaringan discriminator akan mengembalikan value 0 atau 1

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.
Kullback-Leibler divergence tidak menerapkan symmetric nature untuk mengukur jarak antara dua probabilitas distribusi, sedangkan Jensen-Shannon divergence menerapkan symmetric nature.
13. Jelaskan apa itu fungsi objektif yang berfungsi untuk mengukur kesamaan antara gambar yang dibuat dengan yang asli.
Fungsi objektif berfungsi untuk mengukur 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.
Scoring algoritma adalah algoritma yang digunakan untuk menghitung akurasi GAN.
15. Jelaskan kelebihan dan kekurangan GAN.
Kelebihan
 - (a) Data hasil generate GAN terlihat mirip dengan data asli.
 - (b) Gampang dalam mengenali suatu objek dan bisa menghitung jarak antar objek
 Kekurangan
 - (a) Susah untuk dilatih

8.1.2 Praktek

1. Jelaskan apa itu 3D convolutions
3D convolution adalah operasi menerapkan filter 3D ke data input sepanjang tiga arah, yaitu x, y, z. Operasi akan membuat daftar 3D fitur secara bertumpuk, outputnya nanti berupa kotak.
2. Jelaskan dengan kode program arsitektur dari generator networknya, beserta penjelasan input dan output dari generator network.

Input Layer

Pada input layer diambil sampel vektor 100 dimensi dan dilempar ke hidden

layer pertama berupa tensor.

```
1 input_shape=(batch_size , 100) , output_shape=(batch_size , 100)
2
```

Hidden Layer

Pada hidden layer akan ada proses konversi dari tensor yang dilempar dari layer sebelumnya. Pada hidden layer terdiri dari beberapa dense layer dengan beberapa unit juga.

```
1 neurons=500, input_shape=(batch_size , 100) , output_shape=(
2 batch_size , 500)
```

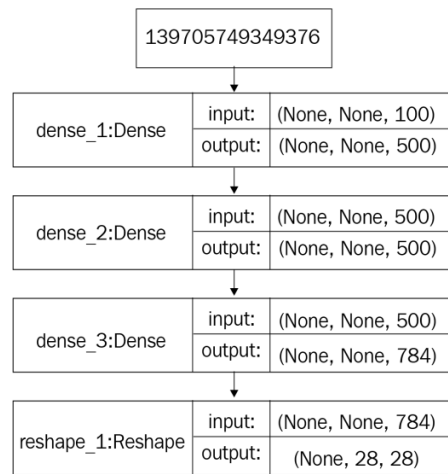
```
1 neurons=500, input_shape=(batch_size , 500) , output_shape=(
2 batch_size , 500)
```

```
1 neurons=784, input_shape=(batch_size , 500) , output_shape=(
2 batch_size , 784)
```

Output Layer

Pada output layer akan digenerate gambar yang memiliki bentuk 28 x 28 unit.

```
1 input_shape=(batch_size , 784) , output_shape=(batch_size , 28,
2 28)
```



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

Input Layer

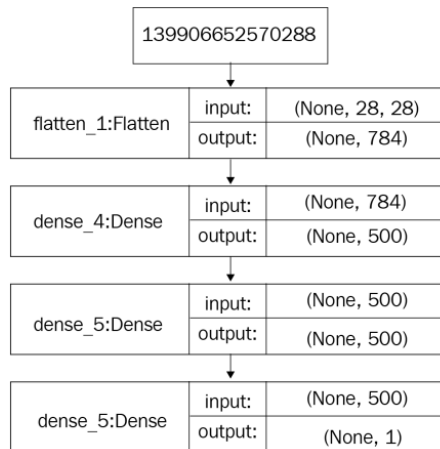
Pada input layer diambil sampel vektor 100 dimensi dan dilempar ke hidden layer pertama berupa tensor.

```
1 input_shape=(batch_size , 100) , output_shape=(batch_size , 100)
2
```

Hidden Layer

Pada hidden layer akan ada proses konversi dari tensor yang dilempar dari layer sebelumnya. Pada hidden layer terdiri dari beberapa dense layer dengan beberapa unit juga.

Pertama dicriminator akan menerima inputan bentuk 28 x 28. Lalu diteruskan ke hidden layer tanpa dimodifikasi. Pada hidden layer akan dimodifikasi. Lalu pada layer terakhir atau output layer ditentukan apakah data tadi adalah asli atau palsu



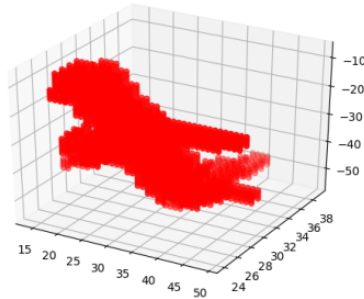
4. Jelaskan proses training 3D-GANs.

- Siapkan sample 200 dimensi berbentuk vektor.
- Generate sebuah gambar palsu dengan model generator.
- Latih jaringan generator dengan gambar asli dan gambar palsu.
- Gunakan adversial model untuk melatih generator model.
- Ulangi langkah diatas sampai epoch tertentu.

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

- Download terlebih dahulu data yang akan diolah

- (b) Siapkan IDE yang akan dipakai untuk mengolah data.
 - (c) Panggil data yang telah didownload tadi.
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.
- (a) Pertama download dataset dengan mengetik `wget http://3dshapenets.cs.princeton.edu/`
 - (b) Lalu extract dataset tadi dengan mengetik `unzip 3DShapeNetsCode.zip`
 - (c) Panggil data yang telah didownload tadi.
7. Jelaskan apa itu voxel dengan ilustrasi dan bahasa paling awam.
Voxel adalah point pada bentuk 3 dimesional, biasanya berupa posisi dari 3 koordinat yaitu x, y, dan z.
8. Visualisasikan dataset tersebut dalam tampilan visual plot, jelaskan cara melakukan visualisasinya.
- (a) Pertama load data gambar 3D.
 - (b) Lalu visualisasikan data gambar 3D tadi.



9. Buka file `run.py` jelaskan perbaris kode pada fungsi untuk membuat generator yaitu `build_generator`.
Pada kode fungsi `build_generator()` dibuat untuk nantinya dipakai sebagai model generator. Di fungsi tersebut berisikan layer-layer yang terla di definisikan sesuai keperluan generator.
10. Jelaskan juga fungsi untuk membangun diskriminator pada fungsi `build_discriminator`.
Pada kode fungsi `build_discriminator()` dibuat untuk nantinya dipakai sebagai model discriminator. Di fungsi tersebut berisikan layer-layer yang terla di definisikan sesuai keperluan discriminator.
11. Jelaskan apa maksud dari kode program `__name__ == '__main__'`


```

1  if __name__ == '__main__':
2

```

Listing 8.1 Kode program utama

Kode program tersebut untuk mengecek apakah module yang dipakai adalah main.

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
10

```

Listing 8.2 Setting Parameter

Kode program diatas tentang mendefinisikan variable yang akan dipakai.

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

```

Listing 8.3 Setting Parameter

Kode program diatas tentang pembuatan dan kompilasi model generator dan discriminator yang nantinya dipakai.

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=[
7  validity])
8  adversarial_model.compile(loss='binary_crossentropy',
9  optimizer=gen_optimizer)

```

Listing 8.4 Membuat dan Kompilasi Model Adversarial

Kode program diatas tentang pembuatan dan kompilasi model adversial yang nantinya dipakai.

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 ...")
5

```

Listing 8.5 Ekstraksi dan load dataset

Kode program diatas tentang meload data gambar 3D.

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

```

1      tensorboard = TensorBoard(log_dir="{}/{}".format(time.
2      time()),
3      tensorboard.set_model(generator)
4      tensorboard.set_model(discriminator)
5

```

Listing 8.6 Instansiasi tensorboard

Kode program diatas tentang mengevaluasi model yang dibuat dengan menggunakan TensorBoard.

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

```

1      labels_real = np.reshape(np.ones((batch_size,)), (-1, 1,
2      1, 1, 1))
3      labels_fake = np.reshape(np.zeros((batch_size,)), (-1, 1,
4      1, 1, 1))
5

```

Listing 8.7 Pelabelan dataset

Kode program diatas tentang pelabelan data asli dan palsu.

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 = []
6

```

Listing 8.8 Setting Epoch

Epoch dilakukan untuk mendapat hasil yang akurat. Kode program diatas tentang menampung tingkat loss dari generator dan discriminator.

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         print("Number of batches:", number_of_batches)
3         for index in range(number_of_batches):
4             print("Batch:", index + 1)
5

```

Listing 8.9 Setting Batch

Batch dipakai terkait pembagian proses pada tiap epoch yang dilakukan.

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 + 1) * batch_size, :, :, :]
3

```

Listing 8.10 Set real images dan vektor noise

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
    network
2         gen_volumes = generator.predict_on_batch(
    z_sample)
3

```

Listing 8.11 Generator Gambar Palsu

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(
    volumes_batch, labels_real)
4             loss_fake = discriminator.train_on_batch(
    gen_volumes, labels_fake)
5
6         d_loss = 0.5 * np.add(loss_real,
    loss_fake)
7         print("d_loss:{}".format(d_loss))

```

```

8
9         else :
10             d_loss = 0.0
11

```

Listing 8.12 Training Diskriminator

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=[
2             batch_size, 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                 gen_losses.append(g_loss)
7                 dis_losses.append(d_loss)

```

Listing 8.13 Training adversarial model

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 :

```

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,
5                 size=[batch_size, 1, 1, 1, z_size]).astype(np.float32)
6             generated_volumes = generator.predict(
7                 z_sample2, verbose=3)
8             for i, generated_volume in enumerate(
9                 generated_volumes[:5]):
10                 voxels = np.squeeze(generated_volume)
11                 voxels[voxels < 0.5] = 0.
12                 voxels[voxels >= 0.5] = 1.
13                 saveFromVoxels(voxels, "results/img_
14                 {}-{}-{}".format(epoch, index, i))

```

Listing 8.14 Buat dan simpan gambar 3D

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(
3             gen_losses), epoch)
4         write_log(tensorboard, 'd_loss', np.mean(
5             dis_losses), epoch)

```

4

Listing 8.15 Simpan Average losses setiap epoch

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"))
3

```

Listing 8.16 Simpan model

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, 1, z_size]).astype(np.float32)
11        generated_volumes = generator.predict(z_sample,
verbose=3)
12
13        for i, generated_volume in enumerate(
generated_volumes[:2]):
14            voxels = np.squeeze(generated_volume)
15            voxels[voxels < 0.5] = 0.
16            voxels[voxels >= 0.5] = 1.
17            saveFromVoxels(voxels, "results/gen-{}".format(i)
18        )

```

Listing 8.17 Testing model**8.1.3 Penanganan Error**

1. Screenshot Error
2. Tuliskan kode eror dan jenis errornya
3. Solusi pemecahan masalah error tersebut

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.

