

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	21
7 Chapter 7	29

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
5.1	1174002 Habib Abdul Rasyid	9
5.1.1	Teori	9
5.1.2	Praktek	11
6	Chapter 6	21
6.1	Habib Abdul Rasyid/ 1174002	21
6.1.1	Teori	21
6.1.2	Praktikum	24
7	Chapter 7	29
7.1	1164013 - Ikrima Ningrumsari mulyana	29
7.1.1	Teori	29
7.1.2	Praktek	34
7.1.3	Penanganan Error	42
7.1.4	Bukti Tidak Plagiat	43
	Daftar Pustaka	45
	Index	47

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

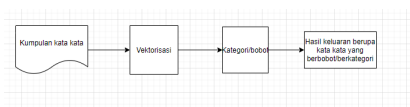
CHAPTER 5

5.1 1174002 Habib Abdul Rasyid

5.1.1 Teori

1. Jelaskan Kenapa Kata-Kata harus dilakukan vektorisasi lengkapi dengan ilustrasi gambar.

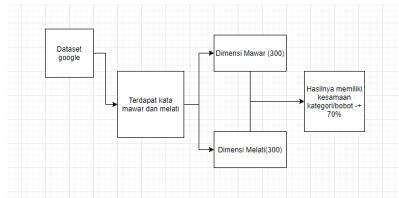
Karena Bertujuan untuk memprediksi kemunculan suatu kata sebuah kalimat. atau vektorisasi kata ini juga dapat memprediksikan suatu nilai,bobot, atau kategori yang sama, misalnya ada kategori kendaraan, motor dan mobil termasuk kedalam kategori kendaraan sehingga kita dapat memprediksi kemunculan kendaraan motor dan mobil pada kategori kendaraan.



Gambar 5.1 Teori 1

2. Jelaskan Mengapa dimensi dari vektor dataset google bisa mencapai 300 dilengkapi dengan ilustrasi gambar.

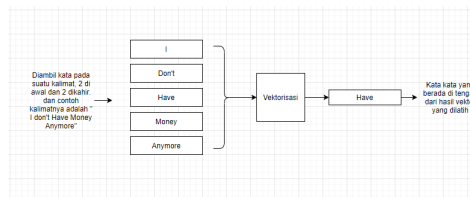
Dimensi dataset dari google bisa mencapai 300 karena dimensi dari vektor tersebut digunakan untuk membandingkan bobot dari setiap kata, misalkan terdapat kata mawar dan melati pada dataset google tersebut setiap kata tersebut di buat dimensi vektor 300 untuk kata mawar dan 300 dimensi vektor juga untuk kata melati kemudian kata tersebut di bandingkan bobot kesamaan katanya maka akan muncul akurasi sekitar 70 persen kesamaan bobot dikarenakan kata mawar dan melati sama sama di gunakan untuk jenis bunga.



Gambar 5.2 Teori 2

3. Jelaskan Konsep vektorisasi untuk kata . dilengkapi dengan ilustrasi atau gambar.

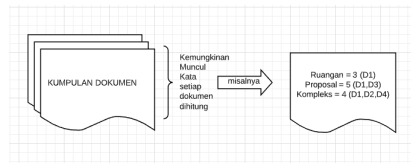
Vektorisasi untuk kata untuk mengetahui kata tengah dari suatu kalimat atau kata utama atau objek utama pada suatu kalimat contoh (Jangan lupa subscribe channel saya ya sekian treimakasih) kata tengah tersebut merupakan channel yang memiliki bobot sebagai kata tengah dari suatu kalimat atau bobot sebagai objek dari suatu kalimat. hal ini sangat berkaitan dengan dimensi vektor pada dataset google yang 300 tadi karena untuk mendapatkan nilai atau bobot dari kata tengah tersebut di dapatkan dari proses dimensi dari kata tersebut.



Gambar 5.3 Teori 3

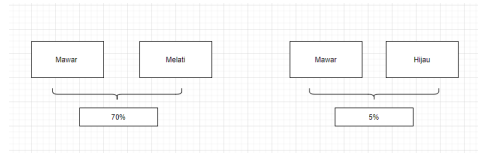
4. Jelaskan Konsep vektorisasi untuk dokumen. dilengkapi dengan ilustrasi atau gambar.

Vektorisasi untuk dokumen hampir sama seperti vektorisasi untuk kata hanya saja pemilihan kata utama atau kata tengah terdapat pada satu dokumen jadi mesin akan membuat dimensi vektor 300 untuk dokumen dan nanti kata tengahnya akan di sandingkan pada dokumen yang terdapat pada dokumen tersebut.



Gambar 5.4 Teori 4

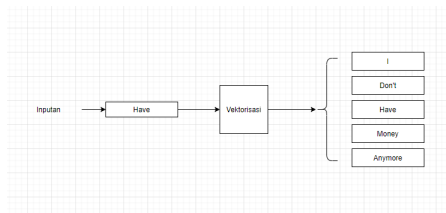
5. Jelaskan apa mean dan standar deviasi, lengkapi dengan ilustrasi atau gambar.
 mean merupakan petunjuk terhadap kata-kata yang di olah jika kata kata itu akurasi tinggi berarti kata tersebut sering muncul begitu juga sebaliknya, sedangkan standar deviation merupakan standar untuk menimbang kesalahan. sehingga kesalahan tersebut di anggap wajar misalkan kita memperkirakan kedalaman dari dataset merupakan 2 atau 3 tapi pada kenyataanya merupakan 5 itu merupakan kesalahan tapi masih bisa dianggap wajar karna masih mendekati perkiraan awal.



Gambar 5.5 Teori 5

6. Jelaskan Apa itu Skip-Gram sertakan contoh ilustrasi.

Skip-Gram adalah kebalikan dari konsep vektorisasi untuk kata dimana kata tengah menjadi acuan terhadap kata kata pelengkap dalam suatu kalimat.



Gambar 5.6 Teori 6

5.1.2 Praktek

1. Mencoba datasets GoogleNews-vectors

- berikut adalah hasil dari code yang digunakan untuk memanggil data library GENSIM dengan menggunakan perintah import, lalu dari library tersebut

diambil data yang akan digunakan untuk memproses data dari GoogleNews-vector. ilustrasi dapat dilihat pada gambar

```
In [6]: import gensim
In [7]: gnmud = gensim.models.KeyedVectors.load_word2vec_format('GoogleNews-vectors-negative300.bin', binary=True)
```

Gambar 5.7 import gensim dan olah data GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data LOVE yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

[illegible]

Gambar 5.8 hasil olah data LOVE pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data FAITH yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [9]: genmod ['faith']
Out[9]:
array([[ 0.26367188, -0.04150391,  0.1953125,  0.13476562, -0.14646438,
         0.19628901, -0.08453073,  0.1951562,  0.12287031,  0.13476562,
        -0.06046825,  0.10895312, -0.16601562,  0.12769608, -0.27140438,
        -0.08453073,  0.18461953, -0.1951562, -0.1951562,  0.13476562,
        0.15320301, -0.1953125,  0.10253906, -0.01367188,  0.21245451,
        -0.09597301, -0.2284912, -0.09042438,  0.26171875,  0.19027274,
        -0.1328125,  0.16434375,  0.01139524,  0.02111816,  0.19554688,
        -0.04125977,  0.12011719,  0.17484609,  0.22136769, -0.13476562,
        -0.01367188,  0.06464062, -0.17677781, -0.01780954,  0.16484825,
        -0.02180438,  0.01527324, -0.09521406, -0.20066406, -0.148025
```

Gambar 5.9 hasil olah data FAITH pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data FALL yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```

In [10]: genmod ['vall']
array([[ -0.04272461,  -0.10742188,  -0.09277344,  -0.16804531,  -0.1328125,
         -0.06693395,  -0.04321289,  -0.01904277,  -0.14648438,  -0.15906062,
         -0.08951406,  -0.04402188,  -0.014574,    -0.08091406,  -0.19024219,
         -0.11035165,  -0.01092529,  -0.08308071,  -0.0189209,  -0.1953125,
         -0.1615625,  -0.17610875,  -0.02928516,  -0.12109375,  -0.12695312,
         -0.037969,   -0.009375,   -0.01973739,  -0.125,    -0.01548438,
         -0.20951312,  -0.0098877,  -0.07763672,  -0.15527344,  -0.03939555,
         -0.04199219,  -0.29082812,  -0.18554688,  -0.00460604,  -0.02087042,
         -0.13574219,  -0.22558504,  -0.33789062,  -0.03564553,  -0.03839844])

```

Gambar 5.10 hasil olah data FALL pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data SICK yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [51]: genmed ['sick']
Out[51]:
array([-0.82017188e-01,  1.49414062e-01, -0.95277438e-02,  1.64802500e-01,
        -1.39705025e-01,  3.22050025e-01,  1.73828125e-01, -1.47680938e-01,
        1.91874219e-01,  5.46879800e-02,  1.66992188e-01, -1.68945312e-01,
        2.24084350e-01,  0.05760750e-02, -1.60015625e-01, -1.12704600e-01,
        1.66015625e-01,  1.79687500e-01,  5.92841816e-03,  2.45117188e-01,
        7.04034350e-02,  2.56347550e-02,  2.47795075e-02,  4.50048075e-01,
        1.78718936e-01, -9.93821289e-04,  8.88678750e-02, -1.95312500e-01,
        1.81486250e-01,  2.05625000e-01, -1.45987125e-01,  1.80585938e-02,
        9.42382812e-02, -1.12500000e-02,  1.08976090e-02, -6.39648438e-02])
```

Gambar 5.11 hasil olah data SICK pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data CLEAR yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [52]: genmed ['clear']
Out[52]:
array([-2.44148025e-04, -1.02080781e-01, -1.49414062e-01, -4.24884688e-02,
        -1.47680938e-01, -1.46484375e-01,  1.70758125e-01,  1.46484375e-01,
        2.20562500e-01,  9.76562500e-02,  2.47578125e-01, -1.29882812e-01,
        1.24012175e-01,  2.25632812e-02, -2.13871350e-02,  3.18095938e-01,
        2.80195312e-01, -4.76074219e-02, -6.83593750e-02, -1.21895750e-01,
        1.22256250e-02,  2.54915125e-02, -1.11816000e-02,  8.00781250e-01,
        -2.75878986e-02, -6.04248847e-03,  7.77384688e-02, -1.72051562e-01,
        6.46786575e-02,  3.33138000e-01, -1.78718938e-01, -1.48180938e-01,
        7.91815625e-02,  1.07910156e-01, -1.18931562e-01, -0.34980938e-02])
```

Gambar 5.12 hasil olah data CLEAR pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data SHINE yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [53]: genmed ['shine']
Out[53]:
array([-0.12402344,  0.25978562, -0.15917969,  0.27734375,  0.30273438,
        0.89068938,  0.36257812, -0.22846125, -0.18350175,  0.3071875 ,
        -0.18382734,  0.13671875,  0.25398625,  0.87128886,  0.82539862,
        0.21777344,  0.24025438,  0.5234375 ,  0.12384688, -0.19335938,
        -0.95883789,  0.8612793 ,  0.81940918,  0.87617188,  0.85182539,
        0.20019511,  0.30805938,  0.00162500,  0.08020597,  0.14648438,
        -0.34765625,  0.82563477, -0.23925781, -0.84516082, -0.00479128,
        -0.2411894, -0.18945312, -0.15234375, -0.05493184,  0.81434328,
        0.390625 , -0.1180175 ,  0.1484375 , -0.11183594,  0.24511719])
```

Gambar 5.13 hasil olah data SHINE pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data BAG yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [54]: genmed ['bag']
Out[54]:
array([-0.89515625,  0.15234375, -0.12402344,  0.13378986, -0.11328125,
        -0.8125667, -0.16113281,  0.14648438,  0.80595938,  0.148625 ,
        -0.86088559, -0.3846875 ,  0.20996094, -0.84345783, -0.2189375 ,
        -0.89578311, -0.85857111,  0.10253986,  0.19042659, -0.89421828,
        0.18817656, -0.87956984, -0.11891516,  0.87018156,  0.86347556,
        -0.15527344, -0.18945312,  0.11132812,  0.27539862, -0.86787189,
        0.81200641,  0.86689483,  0.2571125 ,  0.8324707 , -0.24609375,
        -0.89541992,  0.81811184,  0.24121894, -0.21875 ,  0.87588559,
        -0.89544453, -0.16113281,  0.15003986, -0.89521484, -0.16801562])
```

Gambar 5.14 hasil olah data BAG pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data CAR yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [55]: genmod ['car']
Out[55]:
array([[ 0.13085938,  0.00842285,  0.03344727, -0.05883769,  0.04083906,
        -0.14579812,  0.04931641, -0.16846521,  0.20890458,  0.11962093,
        0.18064406, -0.25      , -0.10480391, -0.10742168, -0.81879983,
        0.05208129, -0.08218875,  0.00455122,  0.14831225, -0.04543803,
        0.16113251, -0.05611329, -0.03089379,  0.08447266,  0.16218936,
        0.04467773, -0.25527344,  0.25396825,  0.13584375,  0.00756636,
        -0.25589538, -0.07731398, -0.03294889,  0.16380894, -0.12597066,
        -0.09912189,  0.16589388,  0.06884766, -0.18943312,  0.02812831,
        -0.0534668 , -0.63081905,  0.13081984,  0.24121894, -0.134379
```

Gambar 5.15 hasil olah data CAR pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data WASH yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [56]: genmod ['wash']
Out[56]:
array([[ 0.4684922e-03,  1.41681562e-01, -5.4687508e-02,  1.34705625e-02,
        -2.3828129e-01,  2.2431875e-01, -4.6478262e-02, -1.5983812e-01,
        1.07910156e-01,  2.53986250e-01,  1.13525391e-02, -1.66991288e-01,
        -2.7954815e-01,  2.30897121e-01, -0.27480458e-02,  1.06484795e-01,
        -7.4217500e-02,  3.04687588e-01,  2.11914062e-01, -0.88871875e-02,
        2.8778125e-02,  1.12896025e-01,  1.7450847e-02,  2.82681875e-01,
        6.2068121e-02,  1.62109375e-01,  1.93359375e-01,  2.17281550e-02,
        -2.6702880e-03, -0.13085938e-02, -2.38281250e-01,  2.29521812e-02,
```

Gambar 5.16 hasil olah data WASH pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data MOTOR yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [57]: genmod ['motor']
Out[57]:
array([[ 5.7378046e-02,  1.5890625e-01, -4.61425781e-02, -1.3281250e-01,
        -2.59705625e-01, -1.77734375e-01,  3.68521644e-02, -4.3750000e-01,
        2.3475000e-01,  2.5781250e-01,  2.74804688e-02,  2.46484625e-02,
        -2.51953125e-01, -5.78171875e-02,  0.15429688e-02,  1.86787575e-02,
        -0.8308075e-01,  1.96801125e-01,  0.0937500e-01,  1.1236800e-01,
        1.5625000e-01, -4.24804688e-02, -1.3281250e-01,  2.11914062e-01,
        1.2868075e-01,  1.09921875e-01,  1.53273438e-01,  4.388906375e-01,
        3.02734375e-01,  1.53208125e-01, -1.09921875e-01, -1.01874219e-01,
        -2.0528800e-01,  2.20515625e-01,  8.98437500e-02, -7.12899625e-02,
```

Gambar 5.17 hasil olah data MOTOR pada GoogleNews-vector

- lalu pada penggunaan code berikut ini akan mengolah data CYCLE yang terdapat pada file GoogleNews-vector, hasil dari pemrosesannya dapat dilihat pada gambar

```
In [58]: genmod ['cycle']
Out[58]:
array([[ 0.04541816,  0.21679688, -0.02709961,  0.12353516,  0.20703125,
        -0.1328125 ,  0.26367188, -0.1289625 ,  0.125  ,  0.15332081,
        -0.18027119, -0.1528012 ,  0.00170756,  0.21973756, -0.1503812 ,
        0.02654777, -0.07568359, -0.0625  ,  0.04614258, -0.31054688,
        -0.13379999, -0.11689922, -0.3359375 ,  0.0971125 ,  0.08447266,
        0.07220562, -0.06445312,  0.09519758,  0.14941406,  0.13071875,
        0.18082734,  0.02172852, -0.18093359,  0.02490234, -0.18649531,
        -0.0541892 ,  0.20462188, -0.40070662,  0.00347056, -0.06447266,
        0.17871894,  0.01165771, -0.01696777,  0.13671875, -0.16480625 ,
```

Gambar 5.18 hasil olah data CYCLE pada GoogleNews-vector

- dan pada hasil code berikut ini adalah hasil dari proses penggunaan perintah code similarity yang akan menghitung nilai value data yang dibandingkan dengan masing - masing kata seperti pada hasil dari perbandingan kata LOVE disandingkan dengan FAITH menghasilkan nilai 37 persen, sedangkan kata WASH dan SHINE menghasilkan nilai 27 persen dan kata CAR yang disandingkan dengan kata MOTOR menghasilkan 48 persen, dimana kita dapat menyimpulkan bahwa semakin data kata tersebut memiliki

tingkat kesamaan yang tinggi maka nilai hasil yang ditampilkanpun akan semakin tinggi. ilustrasi bisa dilihat pada gambar

```
In [19]: genmod.similarity('love', 'faith')
Out[19]: 0.3705347934587281

In [20]: genmod.similarity('wash', 'shine')
Out[20]: 0.2770128965426825

In [21]: genmod.similarity('car', 'motor')
Out[21]: 0.4810172832001571

In [22]: genmod.similarity('bag', 'cycle')
Out[22]: 0.040672609213443504

In [23]: genmod.similarity('shine', 'fall')
Out[23]: 0.27789493775772145
```

Gambar 5.19 hasil olah data pada GoogleNews-vector menggunakan SIMILARITY

2. extract_words dan PermutedSentences

pada penjelasan berikut ini akan menyangkut pembersihan data yang akan digunakan untuk diproses, dimana data akan di EXTRACT dari setiap katanya agar terbebas dari data TAG HTML, APOSTROPHES, TANDA BACA, dan SPASI yang berlebih. dengan menggunakan perintah code STRIP dan SPLIT. lalu penggunaan library random yang akan dibuat untuk melakukan KOCLOK data dengan acuan datanya adalah data yang terdapat pada variable KATA. untuk ilustrasi hasil dari codenya dapat dilihat pada gambar

```
In [28]: import re
...: def extract_words(kata):
...:     kata = kata.lower()
...:     kata = re.sub(r'<.*>', '', kata)
...:     kata = re.sub(r'(\w)\s+(\w)', '\1\2', kata)
...:     kata = re.sub(r'(\w)', '\1', kata)
...:     kata = re.sub(r'(\w)', '\1', kata)
...:     kata = kata.strip()
...:     return kata.split()
...:
...:
...: import random
...: class PermutedSentences(object):
...:     def __init__(self, lenght):
...:         self.lenght = lenght
...:
...:     def __iter__(self):
...:         req = list(range(self.lenght))
...:         random.shuffle(req)
...:         for kata in req:
...:             yield kata
```

Gambar 5.20 hasil olah data pada GoogleNews-vector menggunakan extract_words dan PermutedSentences

3. TaggedDocument dan Doc2Vec

gensim merupakan open-source model ruang vektor dan toolkit topic modeling, yang diimplementasikan dalam bahasa pemrograman Python. Untuk kinerja Gensim, digunakan NumPy, SciPy dan Cython (opsional). Gensim secara khusus ditujukan untuk menangani koleksi teks besar dengan menggunakan algoritma secara online. Gensim mengimplementasikan tf-idf, latent semantic analysis (LSA), Latent Dirichlet Analysis (LDA), dan lain-lain.

tagged document merupakan sebuah class yang terdapat pada pemrosesan data pada library gensim yang akan mengolah data teks yang ada pada dokumen - dokumen yang dipakai.

Doc2Vec merupakan algoritma doct embedding, yaitu pemetaan dari dokumen menjadi vektor, serta pemetaan data dokumen 1 dan dokumen lainnya. ilustrasi dari tagged document dan Word2Vec ada pada gambar

```
In [2]: from gensim.models.doc2vec import TaggedDocument
...: from gensim.models import Doc2Vec
```

Gambar 5.21 TaggedDocument dan Doc2Vec

4. Praktek data training

pertama buka data training yang akan diolah pada aplikasi python, import library OS dan membuat data variable `unsup_senteces` dengan nilai array kosong. buat data direktori untuk memanggil data yang akan diolah dan buat juga variable data nilai `fname` yang akan memproses data `dirname` untuk diisikan pada variable `unsup_sentences`. code yang digunakan dapat dilihat pada gambar

[illegible]

Gambar 5.22 data code praktik data training

data pada hasil code digambar berikut 5.23, menghasilkan data pada gambar 5.24 yang akan memunculkan data variable DIRNAME, FNAME, KATA dan unsup_sentences yang memiliki data sebanyak 55 kata dalam file yang diolah tersebut. hasil run dengan menggunakan code pada gambar 5.25, menghasilkan data nilai yang terdapat pada gambar 5.26. lalu pada code yang terdapat digambar 5.27, menghasilkan data 5.28.

```

import os
unsp_sentences = []

for dirname in ["train/pos", "train/neg", "train/unsp", "test/pos", "test/neg"]:
    for fname in sorted(os.listdir("%sImdb" % dirname)):
        if fname[-4:] == ".txt":
            with open("%sImdb" % dirname + "/" + fname, encoding="UTF-8") as f:
                kdata = f.read()
                words = extract_words(kdata)
                unsp_sentences.append(TaggedDocument(words, [dirname + "/" + fname]))

```

Gambar 5.23 data code praktek data training

Name	Type	Size	Value
dirname	str	1	test/neg
fname	str	1	9_4.txt
kata	str	1	David Bryce's comments nearby are exceptionally well written and infor ...
unsup_sentences	list	73293	[TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, Tagge ...
words	list	55	['.', ',', '!', '?', ':', ';', '"', '<', '>', '=', '~', '^', '*', '%', '\$', '@', '#', '& ...]

Gambar 5.24 data code praktek data training

```
for dirname in ["review_polarity/txt_sentences/pos", "review_polarity/txt_sentences/neg"]:
    for frame in sorted(os.listdir(dirname)):
        if frame[0:4] == ".txt":
            with open(dirname + "/" + frame, encoding='UTF-8') as f:
                for i, data in enumerate(f):
                    words = extract_words(data)
                    aux_sentences.append(TaggedDocument(words, ["%s-%d" % (dirname, frame), i]))
```

Gambar 5.25 data code praktek data training

Name	Type	Size	Value
dirname	str	1	review_polarity/txt_tokenized/neg
fname	str	1	cv999_14636.txt
i	int	1	24
kata	str	1	after watching a night at the roxbury, you'll be left with exactly
unsup_sentences	list	130013	[TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument]
words	list	3	['a', ' ', 'a']

Gambar 5.26 data code praktik data training

```
# In[23]:
with open("stanfordSentimentTreebank/original_rt_snippets.txt", encoding="UTF-8") as f:
    for i, line in enumerate(f):
        words = extract_words(kata)
        unsup_sentences.append(TaggedDocument(words, ["rt-%d" % i]))
```

Gambar 5.27 data code praktek data training

Name	Type	Size	Value
dirname	str	1	review_polarity/txt_sentoken/neg
fname	str	1	c\999_14636.txt
i	int	1	10604
kata	str	1	after watching _a_night_at_the_oxbury_, you'll be left with exactly ...
line	str	1	her face walked out smiling words like "horrible" and "terrible, ..."
unusp_sentences	list	146810	[TaggedDocument, TaggedDocument, TaggedDocument, TaggedDocument, Tagge ...
words	list	5	['.', ',', '.', '.']

Gambar 5.28 data code praktek data training

5. Why need Shuffled and Clean memory

dilakukan shuffled adalah agar datanya lebih mudah untuk diolah dan untuk menentukan tingkat tinggi akurasi dari hasil pemrosesan. dan dilakukan pembersihan memory adalah agar chace yang disimpan tidak membuat proses pada komputer mejadi lambat dan dapat digunakan untuk memproses data lainnya agar menjadi lebih ringan dan cepat. pada gambar 5.29 adlah proses untuk melakukan pengoclokkan data dan pada gambar 5.30 adalah proses untuk memasukan data unsup_sentences kedalam variable muter untuk diproses dengan class PermuterSentences. dan pada gambar 5.31 adalah code yang digunakan untuk membersihkan data memory.

```

In [28]: import re
...: def extract_words(kata):
...:     kata = kata.lower()
...:     kata = re.sub('<[^>]+>', '', kata)
...:     kata = re.sub('(\w)\s', '\12', kata)
...:     kata = re.sub('(\w)', ' ', kata)
...:     kata = re.sub('\s+', ' ', kata)
...:     kata = kata.strip()
...:     return kata.split()
...:
...:
...: import random
...: class PermuteSentences(object):
...:     def __init__(self, lenght):
...:         self.lenght = lenght
...:
...:     def __iter__(self):
...:         req = list(self.lenght)
...:         for kata in req:
...:             yield kata

```

Gambar 5.29 Shuffled dan Randomisasi data

```

In [10]: muter = PermuteSentences(unsup_sentences)
...: mod = Doc2Vec(muter, dm=0, hs=1, size=50)

```

Gambar 5.30 pembuatan variable muter untuk memuat data unsup_sentences

```

mod.delete_temporary_training_data(keep_inference=True)

```

Gambar 5.31 code untuk membersihkan data memory

6. Why model have to be saved

dalam pengolahan data dengan menggunakan proses yang panjang ditakutkan data yang sudah diproses tersebut dapat hilang jika terdapat kejadian atau emergency pada saat pengolahan dan pemrosesan data, misalnya harddisk error atau pun listrik yang padam. dan proses penyimpanan data juga dilakukan agar data yang sudah diolah data dipanggil lagi tanpa harus melakukan proses dari awal sehingga tidak memakan waktu. untuk code yang digunakan dapat dilihat pada gambar berikut ini adalah hasil file dari penggunaan code save tersebut. bisa dilihat pada gambar

```

In [33]: mod.infer_vector(extract_words("This Place is not worth your time, let alone Vegas."))
Out[33]:
array([-0.00174404, -0.00017116, -0.00647218, -0.00174012,  0.00063611,
        -0.003244,    -0.00033739, -0.00080851,  0.00724112,  0.0002009 ,
        -0.00137453, -0.00712116, -0.00925099,  0.00921014,  0.00184124,
        0.00024052, -0.00135045, -0.00040159, -0.00770017,  0.00422007,
        0.0000305 ,  0.00776206, -0.00712774,  0.00674206,  0.00411128,
        -0.00027394, -0.00046713,  0.00135007,  0.00033502, -0.00770078,
        -0.00070081,  0.00646778, -0.00071356, -0.00292108,  0.00391301,
        -0.00043065, -0.00030164, -0.00013665, -0.00029057,  0.00703703,
        0.00126318,  0.00027046, -0.00921578, -0.00080527,  0.0073308 ,
        0.00040559, -0.00027945,  0.00753258,  0.00640478, -0.00094211],
      dtype=float32)

```

Gambar 5.32 hasil file simpan

7. infer_vector

berfungsi untuk dokumen baru, dan bisa menggunakan data vektor yang dilatih secara massal, seperti yang disimpan dalam model, untuk dokumen yang merupakan bagian dari data training. untuk percobaannya dapat dilihat pada gambar

```
In [12]: from sklearn.metrics.pairwise import cosine_similarity
...: cosine_similarity(
...:     [mod.infer_vector(extract_words("Highly recommended."))],
...:     [mod.infer_vector(extract_words("Services sucks."))])
Out[12]: array([[0.280962]], dtype=float32)
```

Gambar 5.33 code dan hasil infer_vector

8. cosine_similarity

merupakan sebuah algoritma yang digunakan untuk membandingkan dari dua buah data yang bukan merupakan data vector untuk menguji nilai kemiripan data satu dengan data lainnya. hasil dari percobaan pada tugas no 8 ini dapat dilihat pada gambar 5.34 yang menghasilkan nilai akurasi sebesar 20 persen dan gambar 5.35 yang menghasilkan nilai akurasi sebesar 91 persen.

```
In [13]: from sklearn.metrics.pairwise import cosine_similarity
...: cosine_similarity(
...:     [mod.infer_vector(extract_words("tolong bantuan."))],
...:     [mod.infer_vector(extract_words("tolong bantuan."))])
Out[13]: array([[0.9143525]], dtype=float32)
```

Gambar 5.34 code dan hasil penggunaan cosine_similarity

```
In [16]: from sklearn.neighbors import KNeighborsClassifier
...: from sklearn.ensemble import RandomForestClassifier
...: from sklearn.model_selection import cross_val_score
...: import numpy as np
...:
...: clf = KNeighborsClassifier(n_neighbors=9)
...: clfrr = RandomForestClassifier()
```

Gambar 5.35 code dan hasil penggunaan cosine_similarity

9. Cross Validation

pertama melakukan import data dari library KNeighborsClassifier, RandomForestClassifier, cross_val_score dan numpy yang digunakan untuk membuat data cross validasi dapat dilihat pada gambar

```
In [40]: scores = cross_val_score(clf, sentvecs, sentiments, cv=5)
...: np.mean(scores), np.std(scores)
Out[40]: (0.5203333333333334, 0.08641746687223791)
```

Gambar 5.36 memasukkan code import library

lalu selanjutnya membuat data variable scores yang akan memuat nilai cross_val_score dengan datanya diambil dari KNeighborsClassifier yang terdiri dari sentvecs, sentiments dan clf dan mengolahnya menggunakan numpy untuk menampilkan data pada gambar yang menghasilkan nilai akurasi sebesar 53 persen.

```
In [41]: scores = cross_val_score(clfrr, sentvecs, sentiments, cv=5)
...: np.mean(scores), np.std(scores)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in version 0.20
to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in version 0.20
to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in version 0.20
to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in version 0.20
to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
Out[41]: (0.5319999999999999, 0.00644359888884536)
```

Gambar 5.37 perhitungan data KNeighborsClassifier dengan cross validasi

membuat data variable scores yang akan memuat nilai cross_val_score dengan datanya diambil dari RandomForestClassifier yang terdiri dari sentvecs, sentiments dan clrrf dan mengolahnya menggunakan numpy untuk menampilkan data pada gambar yang menghasilkan nilai akurasi sebesar 53 persen.

```
In [42]: from sklearn.pipeline import make_pipeline
...: from sklearn.feature_extraction.text import CountVectorizer,
TFidfTransformer
...: pipeline = make_pipeline(CountVectorizer(), TFidfTransformer(),
RandomForestClassifier())
...:
...: scores2 = cross_val_score(pipeline, sentiments, sentiments, cv=5)
...: np.mean(scores2), np.std(scores2)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in
version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in
version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in
version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in
version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
C:\Users\Fatih>C:\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in
version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
Out[42]: (0.7436868686868687, 0.023452879779011715)
```

Gambar 5.38 perhitungan data RandomForestClassifier dengan cross validasi

penggunaan make_pipeline adalah untuk membuat data dari KNeighborsClassifier, RandomForestClassifier dan Vectorizer digabungkan untuk menghasilkan data nilai pada gambar menghasilkan nilai akurasi sebesar 74 persen.

scores	float64 (5,)	[0.53199999 0.53199999 0.53199999 0.53199999 0.53199999]
score0	float64 (5,)	[0.53199999 0.53199999 0.53199999 0.53199999 0.53199999]
score1	float64 (5,)	[0.53199999 0.53199999 0.53199999 0.53199999 0.53199999]
score2	float64 (5,)	[0.53199999 0.53199999 0.53199999 0.53199999 0.53199999]
score3	float64 (5,)	[0.53199999 0.53199999 0.53199999 0.53199999 0.53199999]

Gambar 5.39 perhitungan data Cross Validasi untuk nilai keseluruhan dari KNeighborsClassifier, RandomForestClassifier dan Vectorizer

BAB 6

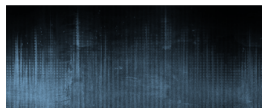
CHAPTER 6

6.1 Habib Abdul Rasyid/ 1174002

6.1.1 Teori

1. Jelaskan kenapa file suara harus dilakukan MFCC dilengkapi dengan ilustrasi atau gambar.

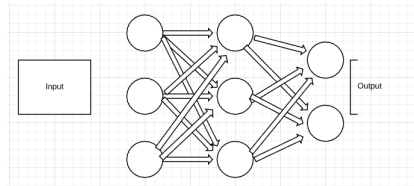
Karena MFCC digunakan untuk mengidentifikasi jenis suara. Sehingga dibutuhkan penggunaan MFCC untuk memproses data tersebut agar dapat dibaca oleh manusia.



Gambar 6.1 Ilustrasi gambar metode MFCC

2. Jelaskan konsep dasar neural network. dilengkapi dengan ilustrasi gambar.

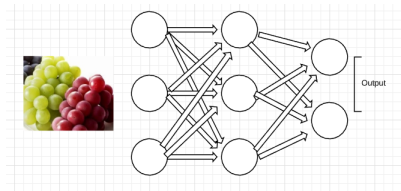
Konsep neural network itu sebenarnya mengadopsi dari kemampuan otak manusia yang mampu memberikan stimulasi/rangsangan, melakukan proses, dan memberikan output. Output diperoleh dari variasi stimulasi dan proses yang terjadi di dalam otak manusia. Neural Network ilustrasi Neural Network dapat dilihat pada gambar



Gambar 6.2 Ilustrasi Konsep dasar neural network

3. Jelaskan konsep pembobotan dalam neural network. dilengkapidengan ilustrasi gambar.

pembobotan dalam neural network yaitu digunakan untuk membedakan objek inputan atau variabel inputan untuk AI. Dimana data inputan yang masuk adalah 2 data "anggur merah" dan "anggur hijau" yang diolah dengan proses membandingkan data dan diolah melalui pembobotan sehingga menampilkan hasil output.



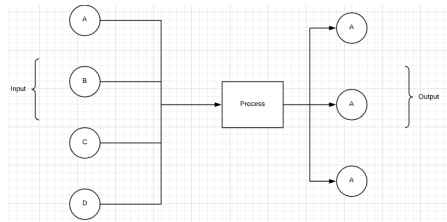
Gambar 6.3 Ilustrasi Konsep pembobotan pada neural network

4. Jelaskan konsep aktifitas dalam neural network. dilengkapi dengan ilustrasi gambar.

Dalam Neural Network cara aktifitas dilakukan terhadap input pada neural network inputan tersebut dimasukan kepada fungsi pada mesin sehingga di hasilkanlah output yang sesuai dengan fungsi tersebut.

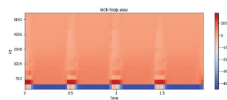
5. Jelaskan cara membaca hasil plot dari MFCC dilengkapi dengan ilustrasi gambar.

cara membaca hasil plotting dari MFCC yaitu tentukan terlebih dahulu batas minimal Hz dari gelombang suara dan batas maksimal dari suara tersebut. kemudian warna yang paling pekat merupakan hasil dari pengolahan data tersebut misalkan muncul warna orange pekat di bagian bawah dan orange muda di



Gambar 6.4 Gambar yang dibaca hasil plotnya

bagian atas yang berarti suara tersebut kuat bagian dasarnya dan biasanya juga antara warna yang pekat tersebut ada jarak.



Gambar 6.5 Ilustrasi Cara Membaca Hasil Plot

6. Jelaskan apa itu one-hot encoding, dilengkapi dengan ilustrasi kode atau gambar.

one-hot encoding merupakan pemberian nilai pada suatu variabel jika nilai itu positif maka nilainya satu dan jika negatif maka nilainya nol.

	pop	rock	blues	reggae	classical
Lagu 1	1	0	0	0	0
Lagu 2	1	0	0	0	0
Lagu 3	0	1	0	0	0
Lagu 4	0	0	1	0	0
Lagu 5	0	0	0	1	0
Lagu 6	0	0	0	0	1
Lagu 7	0	0	0	0	1

Gambar 6.6 Ilustrasi Konsep one-hot encoding

7. Jelaskan apa dari np.unique dan to_categorical dalam kode program, dilengkapi dengan ilustrasi atau gambar.

fungsi dari NP.UNIQUE adalah untuk membuat data elemen menjadi nilai yang bersifat unik dalam artian (Array). Sedangkan perintah to_categorical adalah untuk membuat data integer yang terdeteksi untuk diubah menjadi data matrix biner. ilustrasi dapat dilihat pada gambar

```
>>> np.unique([1, 1, 2, 2, 3, 3])
array([1, 2, 3])
>>> a = np.array([[1, 1], [2, 3]])
>>> np.unique(a)
array([1, 2, 3])
```

Gambar 6.7 Ilustrasi np.unique

8. Jelaskan apa fungsi dari Sequential dalam kode program, dilengkapi dengan ilustrasi atau gambar.


```
to_categorical

keras.utils.to_categorical(y, num_classes=None, dtype='float32')
```

Gambar 6.8 Ilustrasi `to_categorical`

fungsi dari `Sequential` dari code program adalah untuk membagi data - data agar dapat dianalisis oleh sistem lebih mudah, misalkan dari data 100 dibagi prosesnya menjadi 4 yaitu 25.

Robot 1	Robot 2	Robot 3	Robot 4	Robot 5
1-20	21-40	41-60	61-80	81-100

Gambar 6.9 Ilustrasi Konsep pembobotan pada neural network

6.1.2 Praktikum

1. Jelaskan isi dari data GTZAN Genre Collection dan data dari freesound. Buat kode program untuk meload data tersebut untuk digunakan pada MFCC. Jelaskan arti dari perbaris kode yang dibuat (harus beda dengan teman satukelas).

Isi data data merupakan datasets lagu atau suara yang tersiri dari 10 genre yang di simpan kedalam 10 folder yaitu folder blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, dan rock ke sepuluh folder tersebut masing-masing berisi 100 data suara sedangkan data freesound merupakan contoh data suara yang akan di gunakan untuk menguji hasil pengolahan data tersebut dengan menggunakan metode mfcc. apakah suara dari freesound termasuk kategori jazz pop atau sebagainya ?.

```
1 import librosa
2 import librosa.feature
3 import librosa.display
4 import glob
5 import numpy as np
6 import matplotlib.pyplot as plt
7 from keras.layers import Dense, Activation
8 from keras.models import Sequential
9 from keras.utils.np_utils import to_categorical
10
11 # In[1]: buat fungsi mfcc untuk ngetest ajah
12 def display_mfcc(song):
13     y, _ = librosa.load(song)
14     mfcc = librosa.feature.mfcc(y)
15
16     plt.figure(figsize=(10, 4))
17     librosa.display.specshow(mfcc, x_axis='time', y_axis='mel')
18     plt.colorbar()
19     plt.title(song)
20     plt.tight_layout()
21     plt.show()
```

dapat dilihat pada kode diatas pada baris kesatu dilakukan import librosa yang digunakan untuk fungsi mfcc pada suara. pada baris kedua dilakukan import librosa featuse dan pada baris ke tiga dilakukan librosa display selanjutnya pada baris ke empat dilakukan import glob kemudian insert numpy untuk pengolahan data menjadi vektor setelah itu dilakukan import matplotlib untuk melakukan ploting setelah itu dilakukan import librari keras.

Selanjutnya yaitu membuat fungsi mfcc dengan nama display_mfcc yang didalamnya terdapat variabel y yang berisi method librosa load kemudian variabel mfcc yang berisi method librosa featurea mfcc.

2. Jelaskan perbaris kode program dengan kata-kata dan di lengkapi ilustrasi gambar fungsi dari display_mfcc().

```

1 # In[2]: cek fungsi
2 display_mfcc('genres/disco/disco.00069.au')
3 # In[2]: cek fungsi
4 display_mfcc('genres/blues/blues.00069.au')
5 # In[2]: cek fungsi
6 display_mfcc('genres/classical/classical.00069.au')
7 # In[2]: cek fungsi
8 display_mfcc('genres/country/country.00069.au')
9 # In[2]: cek fungsi
10 display_mfcc('genres/hiphop/hiphop.00069.au')
11 # In[2]: cek fungsi
12 display_mfcc('genres/jazz/jazz.00069.au')
13 # In[2]: cek fungsi
14 display_mfcc('genres/pop/pop.00069.au')
15 # In[2]: cek fungsi
16 display_mfcc('genres/reggae/reggae.00069.au')
17 # In[2]: cek fungsi
18 display_mfcc('genres/rock/rock.00069.au')

```

pada baris ke dua program diatas digunakan untuk mendisplay tampilan gelombang suara dari file 266093__stereo-surgeon__kick-loop-5.wav menggunakan metode mfcc dengan menggunakan fungsi display_mfcc yang telah tadi di buat pada nomer dua begitu juga pada baris ke 4 6 8 sampai ke 22 secara teksis sama menggunakan fungsi display_mfcc hanyasaja beda peyimpanan data yang akan di tampilkan atau di eksekusi.

3. Jelaskan perbaris dengan kata-kata dan dilengkapi dengan ilustrasi gambar fungsi dari extract_features_song jelaskan kenapa data yangdiambil merupakan data 25.000 baris pertama ?

```

1 def extract_features_song(f):
2     y, _ = librosa.load(f)
3
4     # get Mel-frequency cepstral coefficients
5     mfcc = librosa.feature.mfcc(y)
6     # normalize values between -1,1 (divide by max)
7     mfcc /= np.amax(np.absolute(mfcc))
8
9     return np.ndarray.flatten(mfcc)[:25000]

```

pada baris ke tiga di definisikan nama `extract_features_song` yang nantinya akan di gunakan pada fungsi yang lainya kemudian dibuat variabel `y` dengan method `librosa load` setelah itu dibuat variabel baru `mfcc` dengan isi `librosa features mfcc` dengan isi variabel `y` tadi kemudian dibuat variabel `mfcc` dengan isian `np.max` dan variabel `mfcc` tadi terakhir di buat array dari data tersebut merupakan data 25000 data pertama. kenapa data 25000 pertama yang digunakan dikarenakan data tersebut digunakan sebagai data testing semakin besar data testing yang di gunakan maka semakin akurat hasil AI. tapi sebenarnya data tersebut relatif bisa lebih besar atau lebih kecil tergantung pada komputer masing masing.

4. Jelaskan Perbaris kode program dengan kata-kata dan di lengkapi ilustrasi gambar fungsi dari generate features and labels.

```

1 def generate_features_and_labels():
2     all_features = []
3     all_labels = []
4
5     genres = ['blues', 'classical', 'country', 'disco', 'hiphop',
6              'jazz', 'metal', 'pop', 'reggae', 'rock']
7     for genre in genres:
8         sound_files = glob.glob('genres/'+genre+'/*.au')
9         print('Processing %d songs in %s genre...' % (len(
10            sound_files), genre))
11         for f in sound_files:
12             features = extract_features_song(f)
13             all_features.append(features)
14             all_labels.append(genre)
15
16     # convert labels to one-hot encoding cth blues : 1000000000
17     classic 0100000000
18     label_uniq_ids, label_row_ids = np.unique(all_labels,
19        return_inverse=True)#ke integer
20     label_row_ids = label_row_ids.astype(np.int32, copy=False)
21     onehot_labels = to_categorical(label_row_ids, len(
22        label_uniq_ids))#ke one hot
23     return np.stack(all_features), onehot_labels

```

pada baris ke tiga merupakan pendefinisian nama fungsi yaitu `generate features and labels` kemudian membuat variabel baru dengan array kosong yaitu `all_features` dan `all_labels` kemudian mendefinisikan isian label untuk genre dengan cara membuat variabel `genres` kemudian di isi dengan 10 genre yang tadi setelah itu dilakukan fungsi `if else` dengan code `for` dan `in` setelah itu akan di buat encoding untuk data tiap tiap label contoh untuk blues 1000000000 dan untuk clasicl 0100000000.

5. Jelaskan dengan kata dan praktek kenapa penggunaan fungsi `generate features and labels` sangat lama saat meload dataset genre tunjukan keluarannya dari komputer sendiri dan artikan maksud dari luaran tersebut.

halnini menjadi lama dikarenakan mesin membaca satu persatu file yang ada pada folder dan dalam folder tersebut terdapat 100 file sehingga wajar menjadi

lama ditambah lagi mengolah data yang tadinya suara menjadi bentuk vektor. berikut merupakan codenya.

```
1 # In[3]: passing parameter dari fitur ekstraksi menggunakan mfcc
2 features , labels = generate_features_and_labels()
```

6. jelaskan kenapa harus dilakukan pemisahan data training dan data testing sebesar 80 persen praktekkan dengan kode dan tunjukan keluarannya dari komputer sendiri dan artikan maksud dari luaran tersebut. untuk code nya adalah sebagai berikut yang merupakan code untuk membagi data sebanyak 80 persen untuk data training maka data musik tadi yang total jumlahnya 1000 akan di bagi dua untuk data training sebanyak 800 dan 200 untuk data testing.

```
1 # In[3]: fitur ekstraksi
2 training_split = 0.8
```

7. praktekkan dan jelaskan masing masing parameter dari fungsi Sequential(). tunjukan keluarannya dari komputer sendiri dan artikan maksud dari luaran tersebut.

fungsi sequential digunakan untuk mengolah data inputan sesuai dengan fungsi yang ada pada fungsi sequential pada fungsi sequential kali ini menggunakan dua fungsi sequential mengkompile data dari 100 neuron atau dari 1 folder file dengan menggunakan fungsi relu dan softmax untuk menghasilkan outputan yang sesuai dengan kriteria.

```
1 # In[3]: membuat seq NN, layer pertama dense dari 100 neurons
2 model = Sequential([
3     Dense(100, input_dim=np.shape(train_input)[1]),
4     Activation('relu'),
5     Dense(10),
6     Activation('softmax'),
7     ])

```

8. praktekkan dan jelaskan masing masing parameter dari fungsi compile(). tunjukan keluarannya dari komputer sendiri dan artikan maksud dari luaran tersebut.

yaitu fungsi compile yang digunakan untuk mengetahui parameter yang digunakan dari data yang telah diolah untuk caranya dapat menggunakan codin-gan sebagai berikut.

```
1 # In[3]: fitur ekstraksi
2 model.compile(optimizer='adam',
3               loss='categorical_crossentropy',
4               metrics=['accuracy'])
5 print(model.summary())
```

9. praktekkan dan jelaskan masing masing parameter dari fungsi fit(). tunjukan keluarannya dari komputer sendiri dan artikan maksud dari luaran tersebut.

pada fungsi ini dilakukan pengolahan data dari 10 label tadi atau 10 file data sets tadi kemudian di hitung tingkat akurasi masing masing dan tingkat kegagalan atau loss data darisetiap file tersebut caranya dengan melakukan codingan berikut. pada gambar tersebut menunjukan 10 pengolahan data untuk menentukan nilai akurasi dan loss dari data tersebut dan selanjutnya dilakukan fungsi evaluasi.

```
1 # In[3]: fitur ekstraksi
2 model.fit(train_input , train_labels , epochs=10, batch_size=32,
3           validation_split=0.2)
```

10. praktekan dan jelaskan masing masing parameter dari fungsi evaluate(). tunjukan keluarannya dari komputer sendiri dan artikan maksud dari luaran tersebut.

pada fungsi ini dilakukan evaluasi terhadap datayang telah di runing sebelumnya untuk lebih jelasnya dapat di lihat codingan tersebut pada codingan tersebut dilakukan evaluasi pada tingkat kegagalan dan akurasi kebenaran maka hasilnya munculkan hasil evaluasi dari 10 proses dari setiap gendre yaitu akurasi sebesar 51 persen dan loss data sebesar 1.4105 data.

```
1 # In[3]: fitur ekstraksi
2 loss , acc = model.evaluate(test_input , test_labels , batch_size
3                             =32)
4 # In[3]: fitur ekstraksi
5 print("Done!")
6 print("Loss: %.4f, accuracy: %.4f" % (loss , acc))
```

11. praktekan dan jelaskan masing masing parameter dari fungsi predic tunjukan keluarannya dari komputer sendiri dan artikan maksud dari luaran tersebut.

fungsi predic merupakan fungsi untuk membandingkan tingkat akurasi pada setiap label yang sepuluh tadi maka data akan di sandingkan ke masing masing tingkat akurasinya, yang akurasinya paling tinggi maka itulah jawaban untuk setiap inputan yang dilakukan.

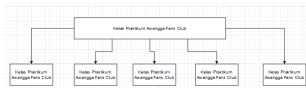
```
1 # In[3]: fitur ekstraksi
2 model.predict(test_input[:1])
```

CHAPTER 7

7.1 1164013 - Ikrima Ningrumsari mulyana

7.1.1 Teori

- 1. Jelaskan kenapa file teks harus di lakukan tokenizer. dilengkapi dengan ilustrasi atau gambar.



Gambar 7.1 Ilustrasi Tokenizer

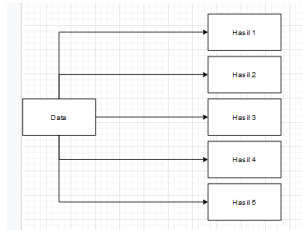
- 2. Jelaskan konsep dasar K Fold Cross Validation pada dataset komentar Youtube pada kode listing 7.1.dilengkapi dengan ilustrasi atau gambar.

```
1 kfold = StratifiedKFold(n_splits=5)
```

```
2 splits = kfold.split(d, d['CLASS'])
3
```

Listing 7.1 K Fold Cross Validation

Pada koding diatas terdapat variabel `kfold` yang didalamnya berisi parameter `split` yang diisikan nilai 5. hal tersebut dimaksudkan untuk membuat pengolahan data akan diulang setiap datanya sebanyak lima kali dengan atribut `class` sebagai acuan pengolahan datanya. Lalu kemudian akan di hasilkan akurasi dari pengulangan data tersebut sebesar sekian persen tergantung datanya



Gambar 7.2 Ilustrasi K Fold Cross Validation

3. Jelaskan apa maksudnya kode program *for train, test in splits*. dilengkapi dengan ilustrasi atau gambar.

For train digunakan untuk melakukan training atau pelatihan pada data yang sudah dideklarasikan sebelumnya. Sedangkan test in split digunakan untuk membatasi jumlah data yang akan diinputkan atau data yang akan digunakan.

```
In [3]: import numpy as np
...: from sklearn.model_selection import train_test_split
...: x = np.arange(10).reshape((10, 2))
...: y = range(5)
...: print(x)
[[0 1]
 [1 2]
 [2 3]
 [3 4]
 [4 5]
 [5 6]
 [6 7]
 [7 8]
 [8 9]
 [9 0]]

In [4]: train(x,
...:          y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0))
...: print(x_train)
[[0 1]
 [1 2]
 [2 3]
 [3 4]
 [4 5]
 [5 6]
 [6 7]
 [7 8]
 [8 9]
 [9 0]]
```

Gambar 7.3 Ilustrasi For train dan test in split

4. Jelaskan apa maksudnya kode program *train_content = d['CONTENT'].iloc[train_idx]* dan *test_content = d['CONTENT'].iloc[test_idx]*. dilengkapi dengan ilustrasi atau gambar.

Maksud dari kode program tersebut adalah membaca isian kolom pada field yang bernama `CONTENT` sebagai data training dan data testing untuk program

No	Nama	Content
1	Mobil	Kendaraan darat yang biasanya memiliki roda 4
2	Motor	Kendaraan darat yang biasanya memiliki roda 2
3	Traktor	Kendaraan darat yang dipakai untuk membajak sawah
4	Helikopter	Kendaraan udara yang memiliki baling-baling untuk terbang

Gambar 7.4 Ilustrasi penggunaan kolom Content

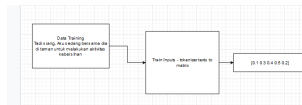
Jelaskan apa maksud dari fungsi `tokenizer = Tokenizer(num_words=2000)` dan `tokenizer.fit_on_texts(train_content)`, dilengkapi dengan ilustrasi atau gambar.

- `tokenizer = Tokenizer(num_words=2000)` digunakan untuk membaca kalimat yang telah dibuat menjadi token sebanyak 2000 kata
- `fit_on_texts` digunakan untuk membuat membaca data token teks yang telah dimasukan kedalam fungsi yaitu fungsi `train_konten`



Gambar 7.5 Ilustrasi fit tokenizer dan num_word=2000

5. Jelaskan apa maksud dari fungsi `d_train_inputs = tokenizer.texts_to_matrix(train_content, mode='tfidf')` dan `d_test_inputs = tokenizer.texts_to_matrix(test_content, mode='tfidf')`, dilengkapi dengan ilustrasi kode dan atau gambar. Untuk digunakan sebagai pengubah urutan teks yang tadi telah dilakukan tokenizer menjadi matriks yang berurutan seperti tf idf



Gambar 7.6 Ilustrasi `d_train_inputs = tokenizer.texts to matrix`

6. Jelaskan apa maksud dari fungsi `d_train_inputs = d_train_inputs/np.amax(np.absolute(d_train_inputs))` dan `d_test_inputs = d_test_inputs/np.amax(np.absolute(d_test_inputs))`, dilengkapi dengan ilustrasi atau gambar. Fungsi tersebut digunakan untuk membagi matriks tfidf dengan penentuan maksimum array sepanjang sumbu sehingga akan menimbulkan garis ke bawah dan ke atas yang membentuk gambar v. Lalu hasil tersebut akan dimasukkan ke variabel `d_train_input` dan `d_test_input` dengan metode `absolute`. Yang berarti tanpa bilangan negatif.
7. Jelaskan apa maksud fungsi dari `d_train_outputs = np_utils.to_categorical(d['CLASS'].iloc[train_idx])` dan `d_test_outputs = np_utils.to_categorical(d['CLASS'].iloc[test_idx])` dalam kode program, dilengkapi dengan ilustrasi atau gambar. Maksud dari fungsi tersebut yaitu untuk merubah nilai vektor yang ada pada atribut `class` menjadi bentuk `matrix` dengan pengurutan berdasarkan data index training dan testing.
8. Jelaskan apa maksud dari fungsi di listing 7.2. Gambarkan ilustrasi Neural Network nya dari model kode tersebut.

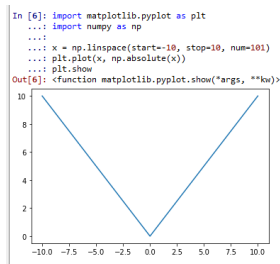

```

1 model = Sequential()
2 model.add(Dense(512, input_shape=(2000,)))
3 model.add(Activation('relu'))
4 model.add(Dropout(0.5))
5 model.add(Dense(2))
6 model.add(Activation('softmax'))
7

```

Listing 7.2 Membuat model Neural Network

model = sequential berarti variabel model berisi method sequential yang berguna untuk searching data dengan menerima parameter atau argumen kunci dengan langkah tertentu untuk mencari data yang telah diolah. Kemudian model akan ditambahkan method add dengan dense yang berarti data - data yang diinputkan akan terhubung, dengan data 612 dan 2000 data kata atau word kemudian model tersebut di masukan fungsi activation dengan rumus atau metode relu. setelah itu data akan di dropout 0.5 atau dipangkas sebanyak 50 persen dikarenakan pada pohon bobot terlalu akurat terhadap data.



Gambar 7.7 Ilustrasi d train inputs

```

In [10]: labels = [0,2,1,2,0,1]
...: keras.utils.to_categorical(labels)Out[11]:
array([[1., 0., 0.],
       [0., 0., 1.],
       [0., 1., 0.],
       [0., 0., 1.],
       [1., 0., 0.],
       [0., 1., 0.]])

```

Gambar 7.8 Ilustrasi train outputs = np utils.to categorical

9. Jelaskan apa maksud dari fungsi di listing 7.3 dengan parameter tersebut.

```

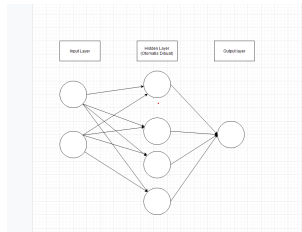
1 model.compile(loss='categorical_crossentropy', optimizer='
  adamax',
2               metrics=['accuracy'])
3

```

Listing 7.3 Compile model

model tersebut kemudian di compile atau di kembalikan kembali fungsi nilainya yangmana akan mengembalikan fungsi nilai loss nya berapa yang diambil dari

fungsi adamax yang berguna untuk mengetahui nilai lossnya kemudian metrics = accuracy merupakan akurasi dari nilai matrixnya. kemudian terdiri atas beberapa layer atau hidden layer. perbedaan antara deep learning dan DNN atau Deep Neural Network yaitu deep learning merupakan pemakai algoritma dari DNN dan DNN merupakan algoritma yang ada pada deep learning.



Gambar 7.9 Ilustrasi Neural Network

10. Jelaskan apa itu Deep Learning

Deep learning merupakan salah satu algoritma yang seperti Neural Network yang menggunakan meta data sebagai inputan dan mengolahnya menggunakan layer layer yang tersembunyi.

11. Jelaskan apa itu Deep Neural Network, dan apa bedanya dengan Deep Learning

Deep Neural Network merupakan algoritma jaringan syaraf yang melakukan pembobotan terhadap data yang sudah ada sebagai acuan untuk data inputan selanjutnya.

12. Jelaskan dengan ilustrasi gambar buatan sendiri(langkah per langkah) bagaimana perhitungan algoritma konvolusi dengan ukuran stride $(NPM \bmod 3 + 1) \times (NPM \bmod 3 + 1)$ yang terdapat max pooling.(nilai 30)

sebelum membuat ilustrasi perlu di ketahui apa itu stride, stride adalah acuan atau parameter yang menentukan pergeseran pada filter fixcel. sebagai contoh nilai stride 1 yang berarti filter akan bergeser sebanyak satu fixcel secara vertikal dan horizontal. selanjutnya apa itu max pooling contoh pada suatu gambar di tentukan Max Pooling dari 3×3 dengan stride 1 yang berarti setiap pergeseran 1 pixel akan diambil nilai terbesar dari pixel 3×3 tersebut.


```

13     # dengan ketentuan jika i lebih kecil daripada o
14     if i > 0:
15         # dibuat variabel img dengan isi keras untuk aktivasi
16         # neural network fungsi yang membaca data yang berada dalam
17         # folder HASYv2 dengan input nilai -1.0 dan 1.0
18         img = keras.preprocessing.image.img_to_array(
19             pil_image.open("HASYv2/" + row[0]))
20         # Pembagian data yang ada pada fungsi img sebanyak
21         255.0
22         img /= 255.0
23         # Penambahan nilai baru pada imgs pada row ke 1 2 dan
24         # dilanjutkan dengan variabel img
25         imgs.append((row[0], row[2], img))
26         # Penambahan nilai pada row ke 2 pada variabel
27         classes
28         classes.append(row[2])
29         # penambahan nilai satu pada variabel i
30         i += 1

```

3. Jelaskan kode program pada blok # In[3]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[3]: shuffle the data, split into 80% train, 20% test
2 # Melakukan import library random
3 import random
4 # melakukan random pada vungsi imgs
5 random.shuffle(imgs)
6 # Menginisiasi variabel split_idx dengan nilai integer 80 persen
7 # dikali dari pengembalian jumlah dari variabel imgs
8 split_idx = int(0.8*len(imgs))
9 # Menginisiasi variabel train dengan isi lebih besar split idx
10 train = imgs[:split_idx]
11 # Menginisiasi variabel test dengan isi lebih kecil split idx
12 test = imgs[split_idx:]

```

4. Jelaskan kode program pada blok # In[4]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[4]:
2 # Melakukan import library numpy dengan inisial np
3 import numpy as np
4 # Menginisiasi variabel train input dengan np method asarray yang
5 # mana membuat array dengan isi row 2 dari data train
6 train_input = np.asarray(list(map(lambda row: row[2], train)))
7 # membuat test input input dengan np method asarray yang mana
8 # membuat array dengan isi row 2 dari data test
9 test_input = np.asarray(list(map(lambda row: row[2], test)))
10 # Menginisiasi variabel train_output dengan np method asarray
11 # yang mana membuat array dengan isi row 1 dari data train
12 train_output = np.asarray(list(map(lambda row: row[1], train)))
13 # Menginisiasi variabel test_output dengan np method asarray yang
14 # mana membuat array dengan isi row 1 dari data test
15 test_output = np.asarray(list(map(lambda row: row[1], test)))

```

5. Jelaskan kode program pada blok # In[5]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```
1 # In[5]: import encoder and one hot
2 # Melakukan import library LabelEncode dari sklearn
3 from sklearn.preprocessing import LabelEncoder
4 # Melakukan import library OneHotEncoder dari sklearn
5 from sklearn.preprocessing import OneHotEncoder
```

6. Jelaskan kode program pada blok # In[6]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```
1 # In[6]:convert class names into one-hot encoding
2
3 # Menginisiasi variabel label_encoder dengan isi LabelEncoder
4 label_encoder = LabelEncoder()
5 # Menginisiasi variabel integer_encoded yang berfungsi untuk
   Menconvert variabel classes kedalam bentuk integer
6 integer_encoded = label_encoder.fit_transform(classes)
```

7. Jelaskan kode program pada blok # In[7]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```
1 # In[7]:then convert integers into one-hot encoding
2 # Menginisiasi variabel onehot_encoder dengan isi OneHotEncoder
3 onehot_encoder = OneHotEncoder(sparse=False)
4 # mengisi variabel integer_encoded dengan isi integer_encoded
   yang telah di convert pada fungsi sebelumnya
5 integer_encoded = integer_encoded.reshape(len(integer_encoded),
   1)
6 # Menconvert variabel integer_encoded kedalam onehot_encoder
7 onehot_encoder.fit(integer_encoded)
```

8. Jelaskan kode program pada blok # In[8]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```
1 # In[8]:convert train and test output to one-hot
2 # Menconvert data train output menggunakan variabel
   label_encoder kedalam variabel train_output_int
3 train_output_int = label_encoder.transform(train_output)
4 # Menconvert variabel train_output_int kedalam fungsi
   onehot_encoder
5 train_output = onehot_encoder.transform(train_output_int.reshape(
   len(train_output_int), 1))
6 # Menconvert data test-output menggunakan variabel label_encoder
   kedalam variabel test_output_int
7 test_output_int = label_encoder.transform(test_output)
8 # Menconvert variabel test_output_int kedalam fungsi
   onehot_encoder
```

```

9 test_output = onehot_encoder.transform(test_output_int.reshape(
    len(test_output_int), 1))
10 # Menginisiasi variabel num_classes dengan isi variabel
    label_encoder dan classess
11 num_classes = len(label_encoder.classes_)
12 # mencetak hasil dari nomer Class berupa persen
13 print("Number of classes: %d" % num_classes)

```

9. Jelaskan kode program pada blok # In[9]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[9]: import sequential
2 # Melakukan import library Sequential dari Keras
3 from keras.models import Sequential
4 # Melakukan import library Dense, Dropout, Flatten dari Keras
5 from keras.layers import Dense, Dropout, Flatten
6 # Melakukan import library Conv2D, MaxPooling2D dari Keras
7 from keras.layers import Conv2D, MaxPooling2D

```

10. Jelaskan kode program pada blok # In[10]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[10]: desain jaringan
2 # Menginisiasi variabel model dengan isian library Sequential
3 model = Sequential()
4 # variabel model di tambahkan library Conv2D tigapuluh dua bit
    dengan ukuran kernel 3 x 3 dan fungsi penghitungan relu dang
    menggunakan data train_input
5 model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
6    input_shape=np.shape(train_input[0])))
7 # variabel model di tambahkan dengan lib MaxPooling2D dengan
    ketentuan ukuran 2 x 2 pixel
8 model.add(MaxPooling2D(pool_size=(2, 2)))
9 # variabel model di tambahkan dengan library Conv2D 32bit dengan
    kernel 3 x 3
10 model.add(Conv2D(32, (3, 3), activation='relu'))
11 # variabel model di tambahkan dengan lib MaxPooling2D dengan
    ketentuan ukuran 2 x 2 pixel
12 model.add(MaxPooling2D(pool_size=(2, 2)))
13 # variabel model di tambahkan library Flatten
14 model.add(Flatten())
15 # variabel model di tambahkan library Dense dengan fungsi tanh
16 model.add(Dense(1024, activation='tanh'))
17 # variabel model di tambahkan library dropout untuk memangkas
    data tree sebesar 50 persen
18 model.add(Dropout(0.5))
19 # variabel model di tambahkan library Dense dengan data dari
    num_classes dan fungsi softmax
20 model.add(Dense(num_classes, activation='softmax'))
21 # mengcompile data model untuk mendapatkan data loss akurasi dan
    optimasi
22 model.compile(loss='categorical_crossentropy', optimizer='adam',
23    metrics=['accuracy'])

```

```

24 # mencetak variabel model kemudian memunculkan kesimpulan berupa
    data total parameter, trainable parameter dan bukan trainable
    parameter
25 print(model.summary())

```

11. Jelaskan kode program pada blok # In[11]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[11]: import sequential
2 # Melakukan import library keras callbacks
3 import keras.callbacks
4 # Menginisiasi variabel tensorboard dengan isi lib keras
5 tensorboard = keras.callbacks.TensorBoard(log_dir='./logs/mnist-
    style')

```

12. Jelaskan kode program pada blok # In[12]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[12]: 5menit kali 10 epoch = 50 menit
2 # fungsi model titambahkan metod fit untuk mengetahui perhitungan
    dari train_input train_output
3 model.fit(train_input, train_output,
4 # dengan batch size 32 bit
5         batch_size=32,
6         epochs=10,
7         verbose=2,
8         validation_split=0.2,
9         callbacks=[tensorboard])
10
11 score = model.evaluate(test_input, test_output, verbose=2)
12 print('Test loss:', score[0])
13 print('Test accuracy:', score[1])

```

13. Jelaskan kode program pada blok # In[13]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[13]: try various model configurations and parameters to find
    the best
2 # Melakukan import library time
3 import time
4 # Menginisiasi variabel result dengan array kosong
5 results = []
6 # melakukan looping dengan ketentuan konvolusi 2 dimensi 1 2
7 for conv2d_count in [1, 2]:
8     # menentukan ukuran besaran fixcel dari data atau konvert 1
        fixcel mnjadi data yang berada pada codigan dibawah.
9     for dense_size in [128, 256, 512, 1024, 2048]:
10         # membuat looping untuk memangkas masing-masing data
            dengan ketentuan 0 persen 25 persen 50 persen dan 75 persen.
11         for dropout in [0.0, 0.25, 0.50, 0.75]:
12             # Menginisiasi variabel model Sequential

```

```

13     model = Sequential()
14     #membuat looping untuk variabel i dengan jarak dari
    hasil konvolusi.
15     for i in range(conv2d_count):
16         # syarat jika i samadengan bobotnya 0
17         if i == 0:
18             # Penambahan method add pada variabel model
    dengan konvolusi 2 dimensi 32 bit didalamnya dan membuat
    kernel dengan ukuran 3 x 3 dan rumus aktivasi relu dan data
    shape yang di hitung dari data train.
19             model.add(Conv2D(32, kernel_size=(3, 3),
    activation='relu', input_shape=np.shape(train_input[0])))
20             # jika tidak
21         else:
22             # Penambahan method add pada variabel model
    dengan konvolusi 2 dimensi 32 bit dengan ukuran kernel 3 x3
    dan fungsi aktivasi relu
23             model.add(Conv2D(32, kernel_size=(3, 3),
    activation='relu'))
24             # Penambahan method add pada variabel model
    dengan isian method Max pooling berdimensi 2 dengan ukuran
    fixcel 2 x 2.
25             model.add(MaxPooling2D(pool_size=(2, 2)))
26             # merubah feature gambar menjadi 1 dimensi vektor
27             model.add(Flatten())
28             # Penambahan method dense untuk pemadatan data dengan
    ukuran dense di tentukan dengan rumus fungsi tanh.
29             model.add(Dense(dense_size, activation='tanh'))
30             # membuat ketentuan jika pemangkasan lebih besar dari
    0 persen
31             if dropout > 0.0:
32                 # Penambahan method dropout pada model dengan
    nilai dari dropout
33                 model.add(Dropout(dropout))
34                 # Penambahan method dense dengan fungsi num
    classs dan rumus softmax
35                 model.add(Dense(num_classes, activation='softmax'))
36                 # mongkcompile variabel model dengan hasi loss
    optimasi dan akurasi matrix
37                 model.compile(loss='categorical_crossentropy',
    optimizer='adam', metrics=['accuracy'])
38                 # melakukan log pada dir
39                 log_dir = './logs/conv2d-%d-dense-%d-dropout-%.2f' %
    (conv2d_count, dense_size, dropout)
40                 # Menginisiasi variabel tensorboard dengan isian dari
    library keras dan nilai dari log_dir
41                 tensorboard = keras.callbacks.TensorBoard(log_dir=
    log_dir)
42                 # Menginisiasi variabel start dengan isian dari
    library time menggunakan method time
43
44                 start = time.time()
45                 # Penambahan method fit pada model dengan data dari
    train input train output nilai batch nilai epoch verbose
    nilai 20 persen validation split dan callback dengan nilai
    tnsorboard.

```



```

46         model.fit(train_input , train_output , batch_size=32,
47                   epochs=10,
48                   verbose=0, validation_split=0.2, callbacks
49                   =[tensorboard ])
50         # Menginisiasi variabel score dengan nilai evaluasi
51         dari model menggunakan data tes input dan tes output
52         score = model.evaluate(test_input , test_output ,
53                                verbose=2)
54         # Menginisiasi variabel end
55         end = time.time()
56         # Menginisiasi variabel elapsed
57         elapsed = end - start
58         # mencetak hasil perhitungan
59         print("Conv2D count: %d, Dense size: %d, Dropout: %.2
60               f - Loss: %.2f, Accuracy: %.2f, Time: %d sec" % (conv2d_count
61               , dense_size , dropout , score[0], score[1], elapsed))
62         results.append((conv2d_count , dense_size , dropout ,
63                         score[0], score[1], elapsed))

```

14. Jelaskan kode program pada blok # In[14]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1  # In[14]:rebuild/retrain a model with the best parameters (from
2  the search) and use all data
3  # Menginisiasi variabel model dengan isian library Sequential
4  model = Sequential()
5  # variabel model di tambahkan library Conv2D tigapuluh dua bit
6  dengan ukuran kernel 3 x 3 dan fungsi penghitungan relu dang
7  menggunakan data train_input
8  model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
9                  input_shape=np.shape(train_input[0])))
10 # variabel model di tambahkan dengan lib MaxPooling2D dengan
11 ketentuan ukuran 2 x 2 pixel
12 model.add(MaxPooling2D(pool_size=(2, 2)))
13 # variabel model di tambahkan dengan library Conv2D 32bit dengan
14 kernel 3 x 3
15 model.add(Conv2D(32, (3, 3), activation='relu'))
16 # variabel model di tambahkan dengan lib MaxPooling2D dengan
17 ketentuan ukuran 2 x 2 pixel
18 model.add(MaxPooling2D(pool_size=(2, 2)))
19 # variabel model di tambahkan library Flatten
20 model.add(Flatten())
21 # variabel model di tambahkan library Dense dengan fungsi tanh
22 model.add(Dense(128, activation='tanh'))
23 # variabel model di tambahkan library dropout untuk memangkas
24 data tree sebesar 50 persen
25 model.add(Dropout(0.5))
26 # variabel model di tambahkan library Dense dengan data dari
27 num_classes dan fungsi softmax
28 model.add(Dense(num_classes , activation='softmax'))
29 # mengcompile data model untuk mendapatkan data loss akurasi dan
30 optimasi
31 model.compile(loss='categorical_crossentropy', optimizer='adam',
32               metrics=[ 'accuracy' ])

```

```

22 # mencetak variabel model kemudian memunculkan kesimpulan berupa
    data total parameter, trainable parameter dan bukan trainable
    parameter
23 print(model.summary())

```

15. Jelaskan kode program pada blok # In[15]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[15]:join train and test data so we train the network on all
    data we have available to us
2 # melakukan join numpy menggunakan data train_input test_input
3 model.fit(np.concatenate((train_input, test_input)),
4           # kelanjutan data yang di gunakan pada join
            train_output test_output
5           np.concatenate((train_output, test_output)),
6           #menggunakan ukuran 32 bit dan epoch 10
7           batch_size=32, epochs=10, verbose=2)

```

16. Jelaskan kode program pada blok # In[16]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[16]:save the trained model
2 #menyimpan model atau mengeksport model yang telah di jalantadi
3 model.save("mathsymbols.model")

```

17. Jelaskan kode program pada blok # In[17]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[17]:save label encoder (to reverse one-hot encoding)
2 # menyompan label encoder dengan nama classes.npy
3 np.save('classes.npy', label_encoder.classes_)

```

18. Jelaskan kode program pada blok # In[18]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[18]:load the pre-trained model and predict the math symbol
    for an arbitrary image;
2 # the code below could be placed in a separate file
3 # mengimpor library keras model
4 import keras.models
5 # Menginisiasi variabel model2 untuk meload model yang telah di
    simpan tadi
6 model2 = keras.models.load_model("mathsymbols.model")
7 # mencetak hasil model2
8 print(model2.summary())

```

19. Jelaskan kode program pada blok # In[19]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[19]: restore the class name to integer encoder
2 # Menginisiasi variabel label encoder ke 2 dengan isian fungsi
   label_encoder.
3 label_encoder2 = LabelEncoder()
4 # Penambahan method classess dengan data classess yang di eksport
   tadi
5 label_encoder2.classes_ = np.load('classes.npy')
6 # membuat fungsi predict dengan path img
7 def predict(img_path):
8     # Menginisiasi variabel newimg dengan membuaay image menjadi
       array dan membuka data berdasarkan img path
9     newimg = keras.preprocessing.image.img_to_array(pil_image.
       open(img_path))
10    # membagi data yang terdapat pada variabel newimg sebanyak
       255
11    newimg /= 255.0
12
13    # do the prediction
14    # Menginisiasi variabel predivtion dengan isian variabel
       model2 menggunakan fungsi predic dengan syarat variabel
       newimg dengan data reshape
15    prediction = model2.predict(newimg.reshape(1, 32, 32, 3))
16
17    # figure out which output neuron had the highest score, and
       reverse the one-hot encoding
18    # Menginisiasi variabel inverted denagan label encoder2 dan
       menggunakan argmax untuk mencari skor luaran tertinggi
19    inverted = label_encoder2.inverse_transform([np.argmax(
       prediction)])
20    # mencetak prediksi gambar dan confidence dari gambar.
21    print("Prediction: %s, confidence: %.2f" % (inverted[0], np.
       max(prediction)))

```

20. Jelaskan kode program pada blok # In[20]. Jelaskan arti dari setiap baris kode yang dibuat(harus beda dengan teman sekelas) dan hasil luarannya dari komputer sendiri.

```

1 # In[20]: grab an image (we'll just use a random training image
   for demonstration purposes)
2 # mencari prediksi menggunakan fungsi prediksi yang di buat tadi
   dari data di HASYv2/hasy-data/v2-00010.png
3 predict("HASYv2/hasy-data/v2-00010.png")
4 # mencari prediksi menggunakan fungsi prediksi yang di buat tadi
   dari data di HASYv2/hasy-data/v2-00500.png
5 predict("HASYv2/hasy-data/v2-00500.png")
6 # mencari prediksi menggunakan fungsi prediksi yang di buat tadi
   dari data di HASYv2/hasy-data/v2-00700.png
7 predict("HASYv2/hasy-data/v2-00700.png")

```

7.1.3 Penanganan Error

1. SS Error

```
File "D:/Kuliah/Semester 4/Kecerdasan Buatan/Saban/src/11740027.py", line 9,
in <module>
  import librosa
ModuleNotFoundError: No module named 'librosa'
```

Gambar 7.11 No Module Name error

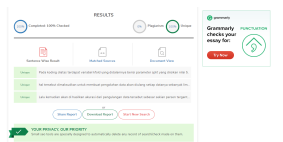
2. Jenis Error

- No Module

3. Cara Penanganan

Dengan cara melakukan instalasi module yang bersangkutan / menginstal library yang digunakan

7.1.4 Bukti Tidak Plagiat



Gambar 7.12 Tidak Melakukan Plagiat Pada Ch 7

DAFTAR PUSTAKA

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

Index

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