

CERDAS MENGUASAI GIT

CERDAS MENGUASAI GIT

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisia

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasisih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

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

*'Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.'*

Imam Syafi'i

CONTRIBUTORS

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

CONTENTS IN BRIEF

1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
5 Chapter 5	9
6 Chapter 6	21
7 Chapter 7	47

DAFTAR ISI

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

5 Chapter 5	9
5.1 1174026- Felix Setiawan Lase	9
5.1.1 Teori	9
5.1.2 Praktek	11
6 Chapter 6	21
6.1 1174021 - Muhammad Fahmi	21
6.1.1 Soal Teori	21
6.1.2 Praktek Program	25
6.1.3 Penanganan Error	32
6.1.4 Bukti Tidak Plagiat	33
6.2 1174026 - Felix Lase	33
6.2.1 Soal Teori	33
6.2.2 Praktek Program	37
6.2.3 Penanganan Error	44
6.2.4 Bukti Tidak Plagiat	45
7 Chapter 7	47
Daftar Pustaka	49
Index	51

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

Bandung, Jawa Barat

Februari, 2019

ACKNOWLEDGMENTS

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

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

R. M. A.

ACRONYMS

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

GLOSSARY

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

SYMBOLS

A Amplitude

$\&$ Propositional logic symbol

a Filter Coefficient

B Number of Beats

INTRODUCTION

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

Informatics Research Center
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[1].

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

BAB 1

CHAPTER 1

BAB 2

CHAPTER 2

BAB 3

CHAPTER 3

BAB 4

CHAPTER 4

BAB 5

CHAPTER 5

5.1 1174026- Felix Setiawan Lase

5.1.1 Teori

1. Mengapa kata-kata dilakukan Vektorisasi

dikarenakan kata-kata yang biasanya digunakan untuk proses data supaya dapat menjadi bagian dari kumpulan beberapa data yang dapat di baca oleh sistem, sehingga sistem tersebut dapat memproses data text secara langsung tanpa di convert. Dapat dilihat seperti gambar dibawah ini



Gambar 5.1 Vektorisasi Kata

2. Mengapa dimensi dari vektor dataset google bisa sampai 300

dikarenakan dimensi pada dataset google bisa mencapai 300 tersebut digunakan untuk membandingkan bobot dari setiap hasil data yang diproses. Dapat dilihat seperti gambar dibawah ini



Gambar 5.2 Dataset Google

3. Jelaskan konsep vektorisasi untuk kata

pada vektorisasi tersebut digunakan word2vec yang memiliki keunggulan yang dapat dibedakan dengan penggunaan bag of word yang biasanya. Dapat dilihat seperti gambar dibawah ini



Gambar 5.3 konsep untuk kata

4. jelaskan konsep vektorisasi untuk dokumen

vektorisasi pada Doc2Vec dimana data yang terdapat pada file document tersebut diolah dengan melakukan pemrosesan yang mengutamakan nilai data filenamenya atau atribut utama dimana nilai data inputnya tidak terlalu diproses. Dapat dilihat seperti gambar dibawah ini



Gambar 5.4 konsep untuk dokumen

5. Jelaskan apa mean dan standar deviasi

Mean adalah nilai rata-rata dari beberapa buah data. Nilai mean dapat ditentukan dengan membagi jumlah data dengan banyaknya data.

sedangkan Standar deviasi adalah nilai statistik yang digunakan untuk menentukan bagaimana sebaran data dalam sampel, dan seberapa dekat titik data individu ke mean atau rata-rata nilai sampel.

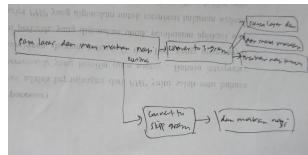
Dapat dilihat seperti gambar dibawah ini

Contoh	
namn	ålder
björn	70
dina	50
ina	40

Gambar 5.5 Mean and Deviation Standard

6. Apa itu skip-gram

Skip-gram merupakan teknik yang digunakan di area speech processing, dimana n-gram yang dibentuk kemudian ditambahkan juga dengan tindakan skip pada token-tokennya. Dapat dilihat seperti gambar dibawah ini



Gambar 5.6 Skip-Gram

5.1.2 Praktek

1. Try datasets GoogleNews-vectors

- berikut adalah hasil dari code yang digunakan untuk memanggil data library GENSIM dengan menggunakan perintah import, lalu dari library tersebut diambilah data yang akan digunakan untuk memproses data dari GoogleNews-vector. ilustrasi dapat dilihat pada gambar

```
In [6]: import gensim  
In [7]: gennod = 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

```
In [8]: genomed['love']
Out[8]:
array([ 0.10302374, -0.15234375,  0.02587091,  0.16503906, -0.16503906,
       0.06584545,  0.29269765, -0.26350718, -0.140625,  0.28117188,
      -0.02642541, -0.08203125, -0.02770996, -0.04394525,  0.23553155,
       0.16992188,  0.12896596,  0.15722656,  0.05765636, -0.06982422,
      -0.03857242,  0.07958984,  0.22949219, -0.14355506,  0.17698675,
      -0.04325441,  0.07550001,  0.05000001,  0.05000001,  0.05000001,
       0.11151566,  0.13594464,  0.15552222,  0.17382667,  0.15286286])
```

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]: genomed['faith']
Out[9]:
array([ 0.26361788, -0.04150391,  0.1953125 ,  0.13476562, -0.14648483,
       0.11962891,  0.04345703,  0.10351562,  0.12070371,  0.13476562,
       0.06646045,  0.18945312, -0.16601562,  0.21679689,  0.27142418,
       0.3202125 ,  0.08419449,  0.16361812,  0.1937125 ,  0.16162562,
       0.13330215,  0.08419449,  0.16361812,  0.1937125 ,  0.16162562,
       0.05895783,  0.22949219, -0.00604248,  0.26171675,  0.10302734,
      -0.1328125 ,  0.21484375,  0.01135254,  0.02111815,  0.01854686])
```

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]: gennod['fall']
Out[10]:
array([-0.04724211,  0.10742188, -0.09277344,  0.16894531, -0.1328125,
       0.06913359,  0.04321289,  0.01904297,  0.14468438,  0.15039862,
       0.08691468,  0.04492188,  0.0145874,  0.08691468,  0.19824219,
      -0.11351556,  0.01092529, -0.08309781,  0.01953125,  0.1953125,
      -0.1015625,   0.13671875,  0.09228156, -0.12109375,  0.12695131,
      -0.03417969,  0.2109375,  0.01977539,  0.125,  0.01544189,
```

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 [11]: genmod['sick']
Out[11]:
array([ 1.82617188e-01,  1.49414062e-01, -4.05273438e-02,
       1.64062500e-01, -2.59756526e-01,  3.22265256e-01,  1.73821258e-01,
       1.47469388e-01,  1.01874219e+01,  4.68758000e-02,  1.69921820e-01,
       1.68945312e-01,  2.2434919e-02,  9.66798750e-02, -1.66815625e-01,
       1.12304688e-01,  1.66015265e-01,  1.79687500e-01,  5.92841916e-03,
       2.45117188e-01,  1.00000000e+00])
```

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 [12]: genmod['clear']
Out[12]:
array([-2.44140625e-04, -1.02050701e-01, -1.49414062e-01, -4.248064868e-02,
       1.67986758e-01, -1.46484375e-01, 1.76757812e-01, 1.46484375e-01,
       2.26581250e-01, 9.15625000e-02, -2.07287500e-01, -1.20811250e-01,
       1.36528125e-01, 1.36528125e-01, 1.36528125e-01, 1.36528125e-01,
       2.001195312e-01, 4.76074219e-02, -6.83595750e-02, -1.21093750e-01,
       2.236255625e-01, 3.14453125e-01, -1.11816400e-01, 8.09712500e-02,
       2.593515625e-01, 2.593515625e-01, 2.593515625e-01, 2.593515625e-01,
       9.66758675e-02, -8.11330800e-01, -1.77178385e-01, 4.88398950e-01])
```

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 [93]: genmod ['shine']
Out[93]:
array([-0.124802344,  0.25970562, -0.10017969, -0.2773475,  0.3857348,
       0.13352771,  0.22549252,  0.13352771,  0.22549252, -0.2773475,
      -0.10017969,  0.25970562,  0.13352771,  0.22549252, -0.2773475,
     -0.10017969,  0.13352771,  0.22549252,  0.07128986, -0.02539862,
     0.27777344,  0.24023398,  0.523475,  0.12394688, -0.19355918,
     0.38805351,  0.38805351,  0.08162586, -0.0582937,  0.14648458,
    -0.04816252,  0.02563477,  0.2395181,  0.04541652, -0.04541652,
    -0.04212052,  0.04212052,  0.2395181,  0.04541652, -0.04541652,
     0.390625, -0.2109375,  0.1484375, -0.13183594,  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 [94]: genmod ['bag']
Out[94]:
array([-0.0515625,  0.15234875, -0.12602344,  0.1379896, -0.11378125,
       0.12094125,  0.12094125,  0.0623925,  0.14002125,
      -0.06005599, -0.3046875,  0.20969694, -0.04345783, -0.2109375,
     -0.05957331, -0.05957331,  0.10253985,  0.19042969, -0.09423829,
     0.11132812,  0.11132812,  0.11132812,  0.11132812,  0.11132812,
    -0.15527344, -0.18945132,  0.11132812,  0.27539862, -0.06787109,
     0.05414453,  0.05414453,  0.05414453,  0.24121094,  0.21875479,
    -0.055414453,  0.05414453,  0.24121094,  0.21875479, -0.055414453,
     -0.09814453, -0.16113281,  0.16503986, -0.09521484, -0.16601562,
```

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 [95]: genmod ['car']
Out[95]:
array([ 0.13085938,  0.00842285,  0.03344277, -0.05883789,  0.04003986,
       0.18864469, -0.25,  0.13085938,  0.00842285,  0.03344277, -0.05883789,
      -0.18864469, -0.18864469, -0.18864469, -0.18864469, -0.18864469,
     0.05200195, -0.00216705,  0.06445312,  0.14453125, -0.04541816,
     0.05200195, -0.00216705,  0.06445312,  0.14453125, -0.04541816,
     0.04467773, -0.15527344,  0.25398625,  0.33984375,  0.08756836,
    -0.25585938, -0.01733398,  0.02795886,  0.16389584, -0.12957956,
     0.05346668,  0.05346668,  0.05346668,  0.05346668,  0.05346668,
     -0.05346668, -0.05346668,  0.11083984,  0.24121094, -0.134375,
```

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 [96]: genmod ['wash']
Out[96]:
array([ 9.46044724e-05,  1.4108152e-01, -5.46875000e-02,  1.34765625e-01,
       1.39532125e-01,  1.24315750e-01, -0.44735524e-01, -1.39532125e-01,
      1.07910156e-01,  2.53986250e-01,  1.13523511e-02, -1.66992105e-01,
     -2.795116e-02,  2.08001612e-01, -4.20003224e-02,  1.85462190e-01,
     -7.07875000e-02,  2.07875000e-01,  1.13523511e-02, -1.66992105e-01,
     2.6778125e-01,  2.12098025e-01,  1.74508547e-02,  2.02981954e-03,
     8.00000000e-03,  1.00000000e-01,  2.00000000e-01,  2.00000000e-01,
     -2.67028800e-03, -0.13085938e-02, -2.30212538e-01,  2.23632812e-03],
```

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 [17]: genmod(['motor'])
Out[17]:
array([-1.73734608e-02, 1.58398625e-02, -4.01429781e-02, -1.32812580e-01,
       3.59795625e-01, -1.77734375e-01, 3.68052354e-02, -4.37900000e-01,
       2.34757600e-02, 2.57812500e-01, 1.74884688e-01, 2.41446252e-02,
       1.29525000e-01, -1.29525000e-01, 1.29525000e-01, 1.29525000e-01,
       -1.8338081e-02, 1.58203125e-01, -5.8937500e-02, 1.12384588e-01,
       1.56250000e-01, -4.2489468e-02, -1.58125000e-01, 2.11934802e-01,
       1.29525000e-01, -1.29525000e-01, 1.29525000e-01, 1.29525000e-01,
       3.02734175e-01, 1.53320312e-01, -1.09218754e-01, -1.10794219e-01,
       -2.06388000e-02, 2.28156250e-01, 8.98437500e-02, -7.12396254e-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 [18]: genmod(['cycle'])
Out[18]:
array([ 0.104541016,  0.21679688, -0.02709961,  0.12353516, -0.20703125,
       0.1128125,   0.26367188, -0.12890625, -0.125,   0.15332031,
       -0.1128125,   0.26367188, -0.12890625,  0.125,   0.15332031,
       -0.02563477, -0.07568359, -0.0521875,  0.04614258, -0.21954688,
       -0.11379986, -0.11669922, -0.359377,   0.078125,   0.08447266,
       0.1093275,   0.21827852, -0.10893359,  0.02402034, -0.18644531,
       -0.05541992, -0.29492188, -0.40839682,  0.06347956, -0.08447266,
       0.17871094,  0.01165771, -0.81696777,  0.13618775, -0.1649625,
```

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 ditampilkan 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]: 0.040672609213443584
In [23]: genmod.similarity('shine', 'fall')
Out[23]: 0.2778949377572145
```

Gambar 5.19 hasil olah data pada GoogleNews-vector menggunakan SIMILARITY

2. extract_words dan PermutatedSentences

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, APOSTROPES, 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+)(\w*)', ' \g<1>\g<2> ', kata)
            ...:     kata = re.sub(r'\s+', ' ', kata)
            ...:     kata = re.sub(r'^\s+', '', kata)
            ...:     return kata.strip()
            ...: 
            ...: 
...:     import random
...:     class PermuteSentences(object):
...:         def __init__(self, length):
...:             self.length = length
...: 
...:         def __iter__(self):
...:             req = list(self.length)
...:             req[0] = None
...:             for kata in req:
...:                 yield kata
```

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

3. TaggedDocument dan Doc2Vec

gensim merupakan open-source model ruang vektor dan toolkit topic modeling, yang diimplementasikan dalam bahasa pemrograman Python. Untuk kerja 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 doc 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_sentences dengan nilai array kosong. buatkan data direktori untuk memanggil data yang akan diolah dan buatkan juga variable data nilai fname yang akan memproses data dirname untuk diisikan pada variable unsup_sentences. code yang digunakan dapat dilihat pada gambar

```

    untag_sentences = []

    for dirname in ["train", "test", "trainval", "testval", "test-test"]:
        for file in os.listdir(dirname):
            if file.endswith(".txt") == True:
                if file[0] == "t":
                    f = open(os.path.join(dirname, file), "r")
                    words = extract_words(f)
                    words.append(extract_sentence(file))
                    untag_sentences.append(appendDocument(words, [dirname + " " + file]))
                else:
                    f = open(os.path.join(dirname, file), "r")
                    words = extract_words(f)
                    words.append(extract_sentence(file))
                    untag_sentences.append(appendDocument(words, [file + " " + file]))
```

Test

```

    for dirname in ["train", "test", "trainval", "testval", "test-test"]:
        for file in os.listdir(dirname):
            if file.endswith(".txt") == True:
                if file[0] == "t":
                    f = open(os.path.join(dirname, file), "r")
                    words = extract_words(f)
                    words.append(extract_sentence(file))
                    untag_sentences.append(appendDocument(words, [f"t {file} " + file]))
```

Extract

```

    for i, line in enumerate(untag_sentences):
        if line[0] == "t":
            f = open(os.path.join("train", line[-1]), "r")
            words = extract_words(f)
            words.append(extract_sentence(line[-1]))
            untag_sentences.append(appendDocument(words, [f"t {line[-1]} " + line[-1]]))
```

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.

```
[10]:  
import os  
unisup_sentences = []  
  
for dirname in ["train/neg", "train/pos", "train/neut", "test/neg", "test/pos"]:  
    for fname in os.listdir(dirname):  
        if fname[-4:] == ".txt":  
            with open(os.path.join(dirname, fname), encoding="UTF-8") as f:  
                katas = f.read()  
                words = extract_words(katas)  
                unisup_sentences.append(TaggedDocument(words, [dirname + "/" + fname]))
```

Gambar 5.23 data code praktik data training

Gambar 5.24 data code praktik data training

```
[ 56/123] for dirname in ["review_polarity/txt_sentoken/pos", "review_polarity/txt_sentoken/neg"]:
    for fname in os.listdir(dirname):
        if fname[-4:] == ".txt":
            with open(os.path.join(dirname, fname), encoding='UTF-8') as f:
                for i, kata in enumerate(f):
                    words = extract_words(kata)
                    print("%s\t%s\t%s\t%s\t%s" % (os.path.join(dirname, fname), i, kata, words, len(words)))
```

Gambar 5.25 data code praktik data training

Gambar 5.26 data code praktik data training

```
with open('rtf2sentences/reback/original_rtf_snippets.txt', encoding='UTF-8') as f:
    for line in enumerate(f):
        words = extract_words(line)
        unsup_sentences.append(TaggedDocument(words, ['rtf-%s' % i]))
```

Gambar 5.27 data code praktek data training

Name	Type	Size	Value
dname	str	1	review_polarity/txt_sentoken/neg
fname	str	1	cv995_14036.txt
i	int	1	100004
kata	str	1	after watching _a_night_at_the_rosbury_ , you'll be able with ease to recall the words like permissible and sensible .
line	str	1	Her fans walked out muttering words like permissible and sensible .
unsup_sentences	list	148016	[TaggedDocument, TaggedDocument, TaggedDocument, ...]
words	list	3	['.', ',', '']

Gambar 5.28 data code praktek data training

5. Mengapa dibutuhkan Shuffled Dan 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 menjadi lambat dan dapat digunakan untuk memproses data lainnya agar menjadi lebih ringan dan cepat. pada gambar 5.29 adlah proses untuk melakukan pengocokan 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|\_)(\w|_)*)\1\12', kata)
....:     kata = re.sub('^\s+', ' ', 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)
....:         random.shuffle(req)
....:         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

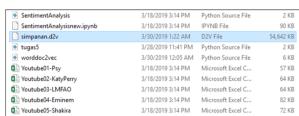
6. Mengapa Model diperlukan Penyimpanan

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

```
mod.save('simpanan.d2v')
```

Gambar 5.32 data code save data

berikut ini adalah hasil file dari penggunaan code save tersebut. bisa dilihat pada gambar



Gambar 5.33 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 [13]: mod.infer_vector(extract_words("This Place is not worth your time.  
Let's go to Vegas."))
Out[13]: [-0.0002094, -0.0001711, -0.0004218, -0.00174012, 0.00050611,  
-0.0003240, 0.0003709, -0.0000682, 0.00724112, 0.00000079,  
-0.00033745, -0.0072118, -0.0025599, 0.00921614, 0.00114124,  
0.00000005, 0.00000001, 0.00000001, 0.00000001, 0.00000001,  
0.0008306, 0.0077520, -0.00713274, 0.00674286, 0.00414208,  
-0.0002354, -0.00848763, 0.00395007, 0.00035502, -0.00737978,  
-0.00000001, 0.00000001, 0.00000001, 0.00000001, 0.00000001,  
-0.00054596, -0.00201164, -0.0091866, -0.00929807, 0.00763393,  
0.00000001, 0.00027064, -0.0092178, -0.00050527, 0.00733351,  
0.00069505, -0.00729494, -0.0092178, -0.00050527, 0.00733351,  
dtype='float32']
```

Gambar 5.34 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 5.35 yang menghasilkan nilai akurasi sebesar 20 persen dan gambar 5.36 yang menghasilkan nilai akurasi sebesar 91 persen.

```
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([[0.206962]], dtype=float32)
```

Gambar 5.35 code dan hasil penggunaan cosine_similarity

```
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([[0.91143525]], dtype=float32)
```

Gambar 5.36 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 [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)  
...: rfc = RandomForestClassifier()
```

Gambar 5.37 memasukan 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 [40]: scores = cross_val_score(clf, sentvecs, sentiments, cv=5)
...: np.mean(scores), np.std(scores)
Out[40]: (0.5283333333333334, 0.006411794687223791)
```

Gambar 5.38 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 clfrf dan mengolahnya menggunakan numpy untuk menampilkan data pada gambar yang menghasilkan nilai akurasi sebesar 53 persen.

Gambar 5.39 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.

```
In [42]: from sklearn.pipeline import make_pipeline
... from sklearn.feature_extraction.text import CountVectorizer,
TfidfTransformer
... pipeline = make_pipeline(CountVectorizer(), TfidfTransformer(),
RandomForestClassifier())
... scores3 = cross_val_score(pipeline,sentences,sentiments, cv=5)
... print(scores3)
C:\Users\Fasti\K-Lancard\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\Fasti\K-Lancard\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\Fasti\K-Lancard\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\Fasti\K-Lancard\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\Fasti\K-Lancard\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\Fasti\K-Lancard\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]: [0.7406666666666667, 0.821428570791171]
```

Gambar 5.40 perhitungan data Cross Validasi untuk nilai keseluruhan dari KNeighborsClassifier, RandomForestClassifier dan Vectorizer

BAB 6

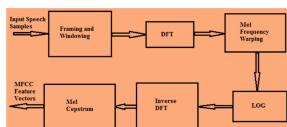
CHAPTER 6

6.1 1174021 - Muhammad Fahmi

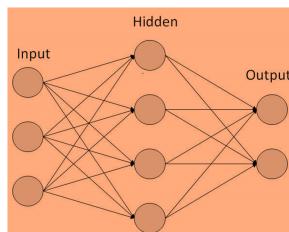
6.1.1 Soal Teori

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

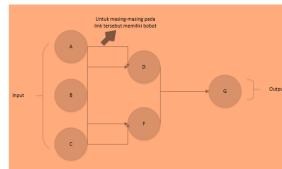
Karena MFCC adalah koefisien yang mewakili audio. Ekstraksi fitur dalam proses ini ditandai dengan konversi data suara menjadi gambar spektrum gelombang. File audio dilakukan oleh MFCC sehingga objek suara dapat dikonversi menjadi matriks. Suara akan menjadi vektor yang akan diproses sebagai output. Selain mempermudah mesin dalam bahasa ini karena mesin tidak dapat membaca teks, maka MFCC perlu mengubah suara menjadi vektor. Untuk ilustrasi, lihat gambar berikut:

**Gambar 6.1** Teori 1

2. Jelaskan konsep dasar neural network.dilengkapi dengan ilustrasi atau gambar.
- Konsep sederhana dari neural network atau jaringan saraf sederhana dengan proses pembelajaran pada anak-anak dengan memetakan pola-pola baru yang diperoleh dari input untuk membuat pola-pola baru pada output. Contoh sederhana ini menganalogikan kinerja otak manusia. Jaringan saraf itu sendiri terdiri dari unit pemrosesan yang disebut neuron yang berisi fungsi adder dan aktivasi. Fungsi aktivasi itu sendiri untuk publikasi diberikan oleh neuron. Jaringan saraf yang mendukung pemikiran sistem atau aplikasi yang melibatkan otak manusia, baik untuk mendukung berbagai elemen sinyal yang diterima, memeriksa kesalahan, dan juga memproses secara paralel. Karakteristik neural network atau jaringan saraf dilihat dari pola hubungan antar neuron, metode penentuan bobot setiap koneksi, dan fungsi aktivasi mereka.. Untuk ilustrasi, lihat gambar berikut:

**Gambar 6.2** Teori 2

3. Jelaskan konsep pembobotan dalam neural network.dilengkapi dengan ilustrasi atau gambar.
- Pembobotan di dalam neural network juga akan menentukan penanda konektivitas. Dalam proses neural network mulai dari input yang diterima oleh neuron bersama dengan nilai bobot masing-masing input. Setelah memasuki neuron, nilai input akan ditambahkan oleh fungsi penerima. Hasil penambahan ini akan diproses oleh masing-masing fungsi neuron, hasil penambahan ini akan dibandingkan dengan nilai ambang tertentu. Jika nilai jumlah ini melebihi nilai ambang batas, aktivasi neuron akan dibatalkan, tetapi sebaliknya jika jumlah hasil di bawah nilai ambang batas, neuron akan diaktifkan. Setelah neuron aktif maka akan mengirimkan nilai output melalui bobot outputnya ke semua neuron yang terkait. Untuk ilustrasi, lihat gambar berikut:



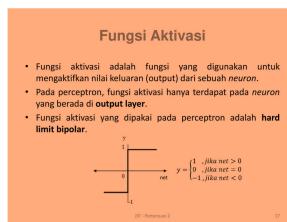
Gambar 6.3 Teori 3

4. Jelaskan konsep fungsi aktifasi dalam neural network. dilengkapi dengan ilustrasi atau gambar.

Fungsi aktivasi dalam neural network ialah merupakan suatu operasi matematik yang dikenakan pada sinyal output. Fungsi ini digunakan untuk mengaktifkan atau menonaktifkan neuron. Fungsi aktivasi ini terbagi setidaknya menjadi 6, adapun sebagai berikut:

- Fungsi Undak Biner Hard Limit, fungsi ini biasanya digunakan oleh jaringan lapiran tunggal untuk mengkonversi nilai input dari suatu variabel yang bernilai kontinu ke suatu nilai output biner 0 atau 1.
- Fungsi Undak Biner Threshold, fungsi ini menggunakan nilai ambang sebagai batasnya.
- Fungsi Bipolar Symetric Hard Limit, fungsi ini memiliki output bernilai 1, 0 atau -1.
- Fungsi Bipolar dengan Threshold, fungsi ini mempunyai output yang bernilai 1, 0 atau -1 untuk batas nilai ambang tertentu.
- Fungsi Linear atau Identitas.

- . Namun, untuk ilustrasi lihat gambar berikut:

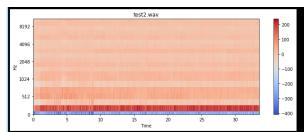


Gambar 6.4 Teori 4

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

Perhatikan gambar dibawah: Gambar menjelaskan bahwa pada waktu ke-5 kekuatan atau layak yang dikeluarkan pada nada ini paling keras pada 20 Hz, selain itu pada 40 -120 Hz daya atau nilai dikeluarkan pada musik yang telah diplot.

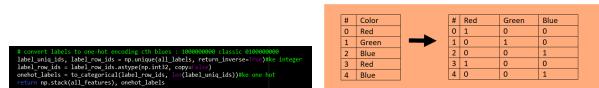
Demikian juga, warna tersebut adalah kekuatan atau nilai tertinggi dibandingkan dengan warna canggih. Yang merah terdengar di bawah pendengaran manusia, sehingga tidak bisa didengar secara langsung.



Gambar 6.5 Teori 5

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

Sederhananya, one-hot encoding ini adalah untuk mengubah hasil data vektorisasi menjadi angka biner 0 dan 1 dan membuat informasi tentang atribut yang diberi label. Untuk ilustrasi, lihat gambar berikut:



Gambar 6.6 Teori 6

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

Fungsi np.unique adalah untuk menemukan berbagai elemen atau array unik, dan dapat mengembalikan elemen unik dari array yang diurutkan. Sebagai ilustrasi, cukup bisa dilihat pada gambar di bawah ini. Gambar tersebut menjelaskan bahwa unik itu sendiri akan mengambil data yang berbeda dari variabel a dalam fungsi array.

```
>>> import numpy as fahmi
>>> x = fahmi.array([[1,2],[1,1],[2,3],[4,5]])
>>> fahmi.unique(x)
array([1, 2, 3, 4, 5])
>>>
```

Gambar 6.7 Teori 7

Fungsi dari to_categorical ialah untuk mengubah suatu vektor yang berupa integer menjadi matrix dengan kelas biner. Untuk ilustrasinya bisa dilihat pada gambar ??

```
>>> Categories ([], object): [a, f, h, i, n]
>>> #import pandas as pd
>>> fahmi = pd.Categorical(['c','l','e','c','o','p','a','s'])
>>> print (fahmi)
[c, l, e, c, o, p, a, s]
>>> Categories ([], object): [a, c, e, i, o, p, s]
>>>
```

Gambar 6.8 Teori 7

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

Fungsi dari Sequential sebagai salah satu jenis model yang digunakan dalam perhitungan. Sequential ini membangun tumpukan linear yang berurutan. Untuk ilustrasi gambar sebagai berikut :

```
In [29]: model = Sequential([
    ...,
    Dense(10, input_dim=shape(train_input)[1]),
    ...,
    Activation('relu'),
    ...,
    Dense(1),
    ...,
    Activation('softmax'),
    ...
])
```

Gambar 6.9 Teori 8

6.1.2 Praktek Program

1. Soal 1

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

Kode di atas menjelaskan isi data GTZAN. Ini adalah kumpulan data yang berisi 10 genre lagu dengan masing-masing genre memiliki 100 lagu yang akan kami lakukan proses MFCC dan juga freesound yang hanya berisi konten lagu, jika GTZAN memiliki beberapa genre jika freesound hanya untuk 1 lagu dan disini kita membuat fungsi untuk membaca file audio dan outputnya sebagai plot, hasilnya adalah sebagai berikut:

```

1 In [1]: import librosa
2 import librosa.feature
3 import librosa.display
4 import glob
5 import os
6 import matplotlib.pyplot as plt
7 from IPython.display import Audio
8 from IPython.display import HTML
9 from keras.models import Sequential
10 from keras.layers import Dense
11 from keras.utils import to_categorical
12
13 def display_mfcc(file):
14     Y, sr = librosa.load(file)
15     mfccs = librosa.feature.mfcc(y=Y)
16     librosa.display.specshow(mfccs, x_axis='time', y_axis='mfcc')
17     plt.title(file)
18     plt.colorbar()
19     plt.show()
20
21 using TensorFlow
22 C:\Users\Ampit\PycharmProjects\librosa\tensorflow\python\framework\types.py:546:
23 FutureWarning: In the future, division by zero will result in an inf instead of a Future
24 warning. This will change in a future version of numpy. It will be understood as (x_type, 1, 1) / (1, type).
25
26 C:\Users\Ampit\PycharmProjects\librosa\tensorflow\python\framework\types.py:547:
27 FutureWarning: In the future, division by zero will result in an inf instead of a Future
28 warning. This will change in a future version of numpy. It will be understood as (x_type, 1, 1) / (1, type).
29
30 C:\Users\Ampit\PycharmProjects\librosa\tensorflow\python\framework\types.py:548:
31 FutureWarning: In the future, division by zero will result in an inf instead of a Future
32 warning. This will change in a future version of numpy. It will be understood as (x_type, 1, 1) / (1, type).
33
34 In [1]: display('test1.wav')
35
36 In [2]: display('test2.wav')
37
38 In [3]: display('dataset/genres/disco/disco.00069.au')
39
40 In [4]: display('dataset/genres/blues/blues.00069.au')
41
42 In [5]: display('dataset/genres/classical/classical.00069.au')
43
44 In [6]: display('dataset/genres/country/country.00069.au')
45
46 In [7]: display('dataset/genres/hiphop/hiphop.00069.au')
47
48 In [8]: display('dataset/genres/jazz/jazz.00069.au')
49
50 In [9]: display('dataset/genres/pop/pop.00069.au')
51
52 In [10]: display('dataset/genres/reggae/reggae.00069.au')
53
54 In [11]: display('dataset/genres/rock/rock.00069.au')

```

Gambar 6.10 Hasil Soal 1.

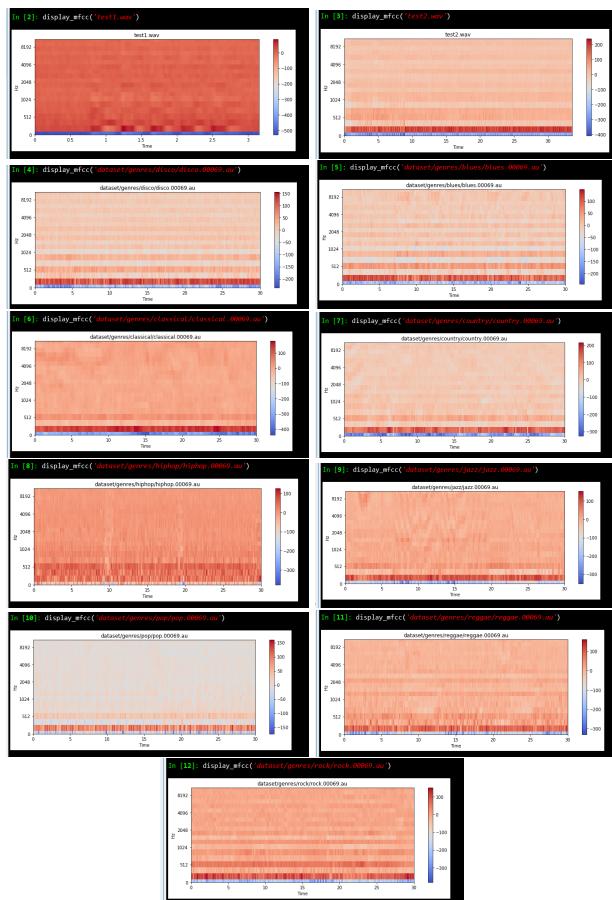
2. Soal 2

```

1 # In[2]: Soal Nomor 2
2 display_mfcc('test1.wav')
3 # In[2]: Soal Nomor 2
4 display_mfcc('test2.wav')
5 # In[2]: Soal Nomor 2
6 display_mfcc('dataset/genres/disco/disco.00069.au')
7 # In[2]: Soal Nomor 2
8 display_mfcc('dataset/genres/blues/blues.00069.au')
9 # In[2]: Soal Nomor 2
10 display_mfcc('dataset/genres/classical/classical.00069.au')
11 # In[2]: Soal Nomor 2
12 display_mfcc('dataset/genres/country/country.00069.au')
13 # In[2]: Soal Nomor 2
14 display_mfcc('dataset/genres/hiphop/hiphop.00069.au')
15 # In[2]: Soal Nomor 2
16 display_mfcc('dataset/genres/jazz/jazz.00069.au')
17 # In[2]: Soal Nomor 2
18 display_mfcc('dataset/genres/pop/pop.00069.au')
19 # In[2]: Soal Nomor 2
20 display_mfcc('dataset/genres/reggae/reggae.00069.au')
21 # In[2]: Soal Nomor 2
22 display_mfcc('dataset/genres/rock/rock.00069.au')

```

Kode di atas akan menampilkan hasil dari proses mfcc yang sudah dibuat fungsi pada soal 1, yaitu display_mfcc() dan akan menampilkan plot dari pembacaan file audio. Berikut adalah hasil setelah saya lakukan running dan pembacaan file audio :



Gambar 6.11 Hasil Soal 2.

3. Soal 3

```

1 # In [3]: Soal Nomor 3
2
3 def extract_features_song(f):
4     y, _ = librosa.load(f)
5
6     # get Mel-frequency cepstral coefficients
7     mfcc = librosa.feature.mfcc(y)
8     # normalize values between -1,1 (divide by max)
9     mfcc /= np.amax(np.absolute(mfcc))
10
11     return np.ndarray.flatten(mfcc)[:25000]

```

Baris pertama itu untuk membuat fungsi `extract_features_song(f)`. Pada baris kedua itu akan me-load data inputan dengan menggunakan `librosa`. Lalu selan-

jutnya untuk membuat sebuah fitur untuk mfcc dari y atau parameter inputan. Lalu akan me-return menjadi array dan akan mengambil 25000 data saja dari hasil vektorisasi dalam 1 lagu. Hasilnya adalah sebagai berikut :

```
[In [12]:] cell extract_features_song(f):
    .....
    # get Mel-frequency cepstral coefficients
    mfcc = librosa.feature.mfcc(y)
    # normalize values between -1,1 (divide by max)
    mfcc /= np.max(np.abs(mfcc))
    .....
    return np.ndarray.flatten(mfcc);
```

Gambar 6.12 Hasil Soal 3.

4. Soal 4

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

Kode di atas dapat digunakan untuk melakukan fungsi yang sebelumnya telah kita lakukan. Kemudian di bagian genre yang disesuaikan dengan dataset nama folder. Untuk baris berikutnya akan mengulang genre folder dengan ekstensi .au. Maka itu akan memanggil fungsi ekstrak lagu. Setiap file dalam folder itu akan diekstraksi menjadi vektor dan akan ditambahkan ke fitur. Dan fungsi yang ditambahkan adalah untuk menumpuk file yang telah di-vektor-kan. Hasil kode tidak menampilkan output. Hasilnya adalah sebagai berikut :

```

    # (4) generate_features_and_labels():
    all_labels = []
    all_features = []
    all_tracks = []

    genres = ['blues', 'classical', 'country', 'disco', 'hiphop', 'jazz', 'metal',
              'pop', 'reggae', 'rock']

    for genre in genres:
        for file in glob.glob(os.path.join('genres', genre, '*.mp3')):
            print("Extracting song in file: %s" % file)
            song = Song(file)
            for track in song.tracks:
                features = extract_features(song)
                all_tracks.append(track)
                all_labels.append(genre)
                all_features.append(features)

    # convert labels to one-hot encoding (0-10)
    label_ids = [label_to_id(label) for label in all_labels]
    return inverse_transform(label_ids), label_ids, label_to_id, np.argmax(all_labels), return_inverse=True

def label_row_idx(label, row_ids):
    label_idx = label_ids.index(label)
    stoprow = np.argmax(label_ids[:label_idx])
    copyrow = np.argmax(label_ids[label_idx+1:])
    os.environ['KMP_DUPLICATE_LIB_OK'] = '1'
    return (stoprow, copyrow, label_idx, label_ids[copyrow], label_ids[stoprow])

```

Gambar 6.13 Hasil Soal 4.

5. Soal 5

```
1 # In [5]: Soal Nomor 5
2
3 features, labels = generate_features_and_labels()
4 # In [5]: Soal Nomor 5
5 print(np.shape(features))
6 print(np.shape(labels))
```

Kode diatas berfungsi untuk melakukan load variabel features dan labels. Mengapa memakan waktu yang lama ? Karena mesin akan melakukan vektorisasi terhadap semua file yang berada pada setiap foldernya, di sini terdapat 10 folder dengan masing-masing folder terdiri atas 100 buah lagu, setiap lagu tersebut akan dilakukan vektorisasi atau ekstraksi data menggunakan mfcc. Oleh karena itu, proses cukup memakan waktu. Hasilnya adalah sebagai berikut :

```
In [15]: features, labels = generate_features_and_labels()
Processing 100 songs in blues genre...
Processing 100 songs in classical genre...
Processing 100 songs in country genre...
Processing 100 songs in disco genre...
Processing 100 songs in hiphop genre...
Processing 100 songs in jazz genre...
Processing 100 songs in metal genre...
Processing 100 songs in pop genre...
Processing 100 songs in reggae genre...
Processing 100 songs in rock genre...
In [16]: print(np.shape(features))
...: print(np.shape(labels))
(1000, 25000)
(1000, 10)
```

Gambar 6.14 Hasil Soal 5

6. Soal 6

```
1 # In [6]: Soal Nomor 6
2 training_split = 0.8
3 # In [6]: Soal Nomor 6
4 alldata = np.column_stack((features , labels))
5 # In [6]: Soal Nomor 6
6 np.random.shuffle(alldata)
7 splitidx = int(len(alldata) * training_split)
8 train , test = alldata [:splitidx ,:] , alldata [splitidx :,:]
9 # In [6]: Soal Nomor 6
10 print(np.shape(train))
11 print(np.shape(test))
12 # In [6]: Soal Nomor 6
```

```

13 train_input = train[:, :-10]
14 train_labels = train[:, -10:]
15 # In [6]: Soal Nomor 6
16 test_input = test[:, :-10]
17 test_labels = test[:, -10:]
18 # In [6]: Soal Nomor 6
19 print(np.shape(train_input))
20 print(np.shape(train_labels))

```

Kode diatas berfungsi untuk melakukan training split 80%. Karena supaya mesin dapat terus belajar tentang data baru, jadi ketika prediksi dibuat tentang data yang terlatih itu bisa mendapatkan persentase yang cukup bagus. Hasilnya adalah sebagai berikut :

In [17]: training_split = 0.8	In [18]: alldata = np.column_stack((features, labels))																																																
In [19]: np.random.shuffle(alldata)splitidx = int(len(alldata) * training_split)train, test = alldata[:splitidx], alldata[splitidx:, :]	In [20]: print(np.shape(train))print(np.shape(test)) (800, 25010) (200, 25010)																																																
In [21]: train_input = train[:, :-10]train_labels = train[:, -10:]	In [22]: test_input = test[:, :-10]test_labels = test[:, -10:]																																																
In [23]: print(np.shape(train_input))print(np.shape(train_labels)) (800, 25000) (800, 10)	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>alldata</td> <td>float32</td> <td>(8000, 25010)</td> <td>[1.47864508 -0.79930827 -0.77314489 ... 0.</td> </tr> <tr> <td>features</td> <td>float32</td> <td>(8000, 25000)</td> <td>[1.47864508 -0.81619907 -0.75164417 ... -0.01183104 0.49327732</td> </tr> <tr> <td>labels</td> <td>float32</td> <td>(8000, 10)</td> <td>[1. 0. 0. ... 0. 0. 0.]</td> </tr> <tr> <td>splitidx</td> <td>int</td> <td>1</td> <td>800</td> </tr> <tr> <td>train</td> <td>float32</td> <td>(800, 25000)</td> <td>[1.47864508 -0.81619907 -0.75164417 ... 0. 1.</td> </tr> <tr> <td>test</td> <td>float32</td> <td>(200, 25000)</td> <td>[1.47864508 -0.81619907 -0.75164417 ... -0.01183104 0.49327732</td> </tr> <tr> <td>train_input</td> <td>float32</td> <td>(800, 25000)</td> <td>[1.47864508 -0.81619907 -0.75164417 ... 0. 1.</td> </tr> <tr> <td>train_labels</td> <td>float32</td> <td>(800, 10)</td> <td>[1. 0. 0. ... 0. 0. 0.]</td> </tr> <tr> <td>train_input</td> <td>float32</td> <td>(800, 25000)</td> <td>[1.47864508 -0.81619907 -0.75164417 ... 0. 1.</td> </tr> <tr> <td>train_labels</td> <td>float32</td> <td>(800, 10)</td> <td>[1. 0. 0. ... 0. 0. 0.]</td> </tr> <tr> <td>training_split</td> <td>float</td> <td>1</td> <td>0.8</td> </tr> </tbody> </table>	Name	Type	Size	Value	alldata	float32	(8000, 25010)	[1.47864508 -0.79930827 -0.77314489 ... 0.	features	float32	(8000, 25000)	[1.47864508 -0.81619907 -0.75164417 ... -0.01183104 0.49327732	labels	float32	(8000, 10)	[1. 0. 0. ... 0. 0. 0.]	splitidx	int	1	800	train	float32	(800, 25000)	[1.47864508 -0.81619907 -0.75164417 ... 0. 1.	test	float32	(200, 25000)	[1.47864508 -0.81619907 -0.75164417 ... -0.01183104 0.49327732	train_input	float32	(800, 25000)	[1.47864508 -0.81619907 -0.75164417 ... 0. 1.	train_labels	float32	(800, 10)	[1. 0. 0. ... 0. 0. 0.]	train_input	float32	(800, 25000)	[1.47864508 -0.81619907 -0.75164417 ... 0. 1.	train_labels	float32	(800, 10)	[1. 0. 0. ... 0. 0. 0.]	training_split	float	1	0.8
Name	Type	Size	Value																																														
alldata	float32	(8000, 25010)	[1.47864508 -0.79930827 -0.77314489 ... 0.																																														
features	float32	(8000, 25000)	[1.47864508 -0.81619907 -0.75164417 ... -0.01183104 0.49327732																																														
labels	float32	(8000, 10)	[1. 0. 0. ... 0. 0. 0.]																																														
splitidx	int	1	800																																														
train	float32	(800, 25000)	[1.47864508 -0.81619907 -0.75164417 ... 0. 1.																																														
test	float32	(200, 25000)	[1.47864508 -0.81619907 -0.75164417 ... -0.01183104 0.49327732																																														
train_input	float32	(800, 25000)	[1.47864508 -0.81619907 -0.75164417 ... 0. 1.																																														
train_labels	float32	(800, 10)	[1. 0. 0. ... 0. 0. 0.]																																														
train_input	float32	(800, 25000)	[1.47864508 -0.81619907 -0.75164417 ... 0. 1.																																														
train_labels	float32	(800, 10)	[1. 0. 0. ... 0. 0. 0.]																																														
training_split	float	1	0.8																																														

Gambar 6.15 Hasil Soal 6.

7. Soal 7

```

1 # In [7]: Soal Nomor 7
2 model = Sequential([
3     Dense(100, input_dim=np.shape(train_input)[1]),
4     Activation('relu'),
5     Dense(10),
6     Activation('softmax'),
7 ])

```

fungsi Sequential() ialah sebuah model untuk menentukan izin pada setiap neuron, di sini adalah 100 dense yang merupakan 100 neuron pertama dari data pelatihan. Fungsi dari relay itu sendiri adalah untuk mengaktifkan neuron atau input yang memiliki nilai maksimum. Sedangkan untuk dense 10 itu adalah output dari hasil neuron yang telah berhasil diaktifkan, untuk dense 10 diaktifkan menggunakan softmax. Hasilnya adalah sebagai berikut :

```
In [24]: model = Sequential([
    ...
    Dense(100, input_dim=train_input[1]),
    Activation('relu'),
    ...
    Dense(10),
    Activation('softmax'),
])

```

Gambar 6.16 Hasil Soal 7.

8. Soal 8

```
1 # In [8]: Soal Nomor 8
2 model.compile(optimizer='adam',
3                 loss='categorical_crossentropy',
4                 metrics=[ 'accuracy' ])
5 print(model.summary())
```

Model Compile di perjelas dengan gambar dibawah, Hasil output pada kode tersebut seperti gambar menjelaskan bahwa dense pertama itu memiliki 100 neurons dengan parameter sekitar 2 juta lebih dengan aktviasi 100, jadi untuk setiap neurons memiliki masing-masing 1 aktivasi. Sama halnya seperti dense 2 memiliki jumlah neurons sebanyak 10 dengan parameter 1010 dan jumlah aktivasinya 10 untuk setiap neurons tersebut dan total parameternya sekitar 2.5 juta data yang akan dilatih pada mesin tersebut.

```
In [25]: model.compile(optimizer='adam',
...
)
print(model.summary())
Model: "sequential_1"
Layer (Type)          Output Shape       Param #
dense_1 (Dense)      (None, 100)        2500100
activation_1 (Activation) (None, 100)        0
dense_2 (Dense)      (None, 10)         1010
activation_2 (Activation) (None, 10)        0
=====
Total params: 2,501,110
Trainable params: 2,501,110
Non-trainable params: 0
None
```

Gambar 6.17 Hasil Soal 8.

9. Soal 9

```
1 # In [9]: Soal Nomor 9
2 model.fit(train_input , train_labels , epochs=10, batch_size=32,
3 validation_split=0.2)
```

Kode tersebut berfungsi untuk melatih mesin dengan data training input dan training label. Epochs ini merupakan iterasi atau pengulangan berapa kali data tersebut akan dilakukan. Batch_size ini adalah jumlah file yang akan dilakukan pelatihan pada setiap 1 kali pengulangan. Sedangkan validation_split itu untuk menentukan persentase dari cross validation atau k-fold sebanyak 20% dari masing-masing data pengulangan, hasilnya adalah sebagai berikut :

Gambar 6.18 Hasil Soal 9.

10. Soal 10

```
1 # In[10]: Soal Nomor 10
2 loss , acc = model.evaluate(test_input , test_labels , batch_size
   =32)
3 # In[10]: Soal Nomor 10
4 print("Done!")
5 print("Loss: %.4f, accuracy: %.4f" % (loss , acc))
```

Fungsi evaluate atau evaluasi ini ialah untuk menguji data pengujian setiap file. Di sini ada prediksi yang hilang, artinya mesin memprediksi data, sedangkan untuk keseluruhan perjanjian sekitar 55%, hasilnya adalah sebagai berikut :

```
[16/22] loss, acc = model.evaluate(text_input, test_labels, batch_size=10)
200/200 [=====] 0s - 200ms/step
[In 28]: print("Done!")
Done!
[In 29]: loss, acc = model.evaluate(text_input, test_labels, batch_size=10)
          accuracy: 0.4900
```

Gambar 6.19 Hasil Soal 10.

11. Soal 11

```
1 # In[11]: Soal Nomor 11  
2 model.predict(test_input[:1])
```

Fungsi Predict ialah untuk menghasilkan suatu nilai yang sudah di prediksi dari data training sebelumnya. Gambar dibawah ini menjelaskan file yang di jalankan tersebut termasuk ke dalam genre apa, hasilnya bisa dilihat pada gambar tersebut presentase yang paling besar yakni genre rock. Maka lagu tersebut termasuk ke dalam genre rock dengan perbandingan presentase hasil prediksi.

```
In [29]: model.predict(test_input[:])  
Out[29]:  
array([[ 6.8733118e-02,  8.2329191e-02,  6.3466556e-02,  1.3786874e-02,  
       2.5682386e-02,  6.7089211e-01,  2.7797744e-04,  0.2083577e-02,  
       2.5101696e-02,  0.32062517e-02]], dtype=float32)
```

Gambar 6.20 Hasil Soal 11.

6.1.3 Penanganan Error

1. ScreenShoot Error

```
[1]: In [1]: import librosa
... import librosa.feature
... import librosa.display
... import matplotlib.pyplot as plt
from collections import Counter
Facebook (most recent call last):
File "/usr/local/lib/python3.6/dist-packages/librosa/_util.py", line 1, in <module>
    import librosa
ModuleNotFoundError: No module named 'librosa'
```

Gambar 6.21 ModuleNotFoundError



Gambar 6.22 An error occurred while starting the kernel

2. Cara Penanganan Error

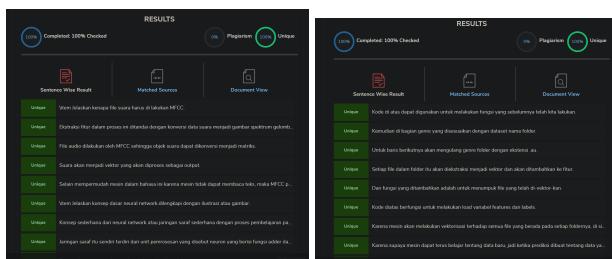
▪ ModuleNotFoundError

Error terdapat pada library yang tidak terbaca, karena library librosa belum di install, solusi nya ialah dengan menginstall library tersebut, pip install librosa.

▪ An error occurred while starting the kernel

Ini error yang tidak biasa karena error terdapat pada kernel, disini solusi nya ialah kita harus menginstall hdf5, yaitu dengan conda uninstall hdf5 dan conda install hdf5.

6.1.4 Bukti Tidak Plagiat



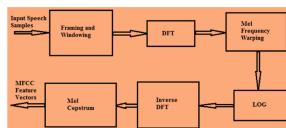
Gambar 6.23 Bukti Tidak Melakukan Plagiat Chapter 6

6.2 1174026 - Felix Lase

6.2.1 Soal Teori

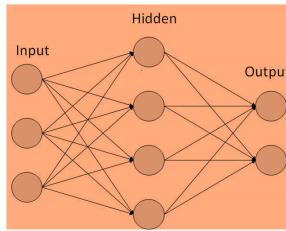
1. Jelaskan kenapa file suara harus di lakukan MFCC. dilengkapi dengan ilustrasi atau gambar.

Karena MFCC adalah koefisien yang mewakili audio. Ekstraksi fitur dalam proses ini ditandai dengan konversi data suara menjadi gambar spektrum gelombang. File audio dilakukan oleh MFCC sehingga objek suara dapat dikonversi menjadi matriks. Suara akan menjadi vektor yang akan diproses sebagai output. Selain mempermudah mesin dalam bahasa ini karena mesin tidak dapat membaca teks, maka MFCC perlu mengubah suara menjadi vektor. Untuk ilustrasi, lihat gambar berikut:



Gambar 6.24 Teori 1

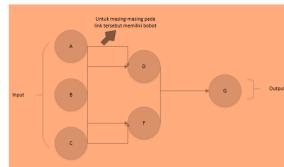
2. Jelaskan konsep dasar neural network.dilengkapi dengan ilustrasi atau gambar. Konsep sederhana dari neural network atau jaringan saraf sederhana dengan proses pembelajaran pada anak-anak dengan memetakan pola-pola baru yang diperoleh dari input untuk membuat pola-pola baru pada output. Contoh sederhana ini menganalogikan kinerja otak manusia. Jaringan saraf itu sendiri terdiri dari unit pemrosesan yang disebut neuron yang berisi fungsi adder dan aktivasi. Fungsi aktivasi itu sendiri untuk publikasi diberikan oleh neuron. Jaringan saraf yang mendukung pemikiran sistem atau aplikasi yang melibatkan otak manusia, baik untuk mendukung berbagai elemen sinyal yang diterima, memeriksa kesalahan, dan juga memproses secara paralel. Karakteristik neural network atau jaringan saraf dilihat dari pola hubungan antar neuron, metode penentuan bobot setiap koneksi, dan fungsi aktivasi mereka.. Untuk ilustrasi, lihat gambar berikut:



Gambar 6.25 Teori 2

3. Jelaskan konsep pembobotan dalam neural network.dilengkapi dengan ilustrasi atau gambar. Pembobotan di dalam neural network juga akan menentukan penanda koneksi. Dalam proses neural network mulai dari input yang diterima oleh neuron bersama dengan nilai bobot masing-masing input. Setelah memasuki neuron,

nilai input akan ditambahkan oleh fungsi penerima. Hasil penambahan ini akan diproses oleh masing-masing fungsi neuron, hasil penambahan ini akan dibandingkan dengan nilai ambang tertentu. Jika nilai jumlah ini melebihi nilai ambang batas, aktivasi neuron akan dibatalkan, tetapi sebaliknya jika jumlah hasil di bawah nilai ambang batas, neuron akan diaktifkan. Setelah neuron aktif maka akan mengirimkan nilai output melalui bobot outputnya ke semua neuron yang terkait. Untuk ilustrasi, lihat gambar berikut:



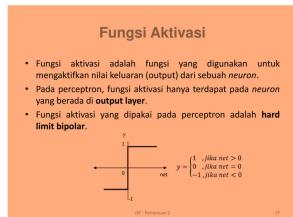
Gambar 6.26 Teori 3

4. Jelaskan konsep fungsi aktifasi dalam neural network. dilengkapi dengan ilustrasi atau gambar.

Fungsi aktivasi dalam neural network ialah merupakan suatu operasi matematik yang dikenakan pada sinyal output. Fungsi ini digunakan untuk mengaktifkan atau menonaktifkan neuron. Fungsi aktivasi ini terbagi setidaknya menjadi 6, adapun sebagai berikut:

- Fungsi Undak Biner Hard Limit, fungsi ini biasanya digunakan oleh jaringan lapiran tunggal untuk mengkonversi nilai input dari suatu variabel yang bernilai kontinu ke suatu nilai output biner 0 atau 1.
- Fungsi Undak Biner Threshold, fungsi ini menggunakan nilai ambang sebagai batasnya.
- Fungsi Bipolar Symetric Hard Limit, fungsi ini memiliki output bernilai 1, 0 atau -1.
- Fungsi Bipolar dengan Threshold, fungsi ini mempunyai output yang bernilai 1, 0 atau -1 untuk batas nilai ambang tertentu.
- Fungsi Linear atau Identitas.

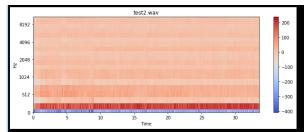
. Namun, untuk ilustrasi lihat gambar berikut:



Gambar 6.27 Teori 4

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

Perhatikan gambar dibawah: Gambar menjelaskan bahwa pada waktu ke-5 kekuatan atau layak yang dikeluarkan pada nada ini paling keras pada 20 Hz, selain itu pada 40 -120 Hz daya atau nilai dikeluarkan pada musik yang telah diplot. Demikian juga, warna tersebut adalah kekuatan atau nilai tertinggi dibandingkan dengan warna canggih. Yang merah terdengar di bawah pendengaran manusia, sehingga tidak bisa didengar secara langsung.



Gambar 6.28 Teori 5

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

Sederhananya, one-hot encoding ini adalah untuk mengubah hasil data vektorisasi menjadi angka biner 0 dan 1 dan membuat informasi tentang atribut yang diberi label. Untuk ilustrasi, lihat gambar berikut:

<pre>1. convert_label_to_one_hot_encoding(np_labels, num_classes=3, return_inverse=False) label1,unq_ids = np.unique(np_labels, return_inverse=True) np_labels = np.eye(num_classes)[label1,unq_ids]#one hot encoding onehot_labels = to_categorical(label1,unq_ids) #one hot encoding</pre>	<p>The diagram shows a mapping from a list of integers to a one-hot encoded matrix. On the left, a list of integers [0, 1, 2, 3, 4] is shown. An arrow points to the right, where a 5x3 matrix is displayed. The columns represent the integers 0, 1, 2, 3, and 4. The matrix has 1s at the positions corresponding to the input values and 0s elsewhere. This represents the one-hot encoding of the input labels.</p>
--	---

Gambar 6.29 Teori 6

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

Fungsi np.unique adalah untuk menemukan berbagai elemen atau array unik, dan dapat mengembalikan elemen unik dari array yang diurutkan. Sebagai ilustrasi, cukup bisa dilihat pada gambar di bawah ini. Gambar tersebut menjelaskan bahwa unik itu sendiri akan mengambil data yang berbeda dari variabel a dalam fungsi array.

```
>>> import numpy as fahmi
>>> x = fahmi.array([[1,2],[1,1],[2,3],[4,5]])
>>> fahmi.unique(x)
array([1, 2, 3, 4, 5])
>>>
```

Gambar 6.30 Teori 7

Fungsi dari `to_categorical` ialah untuk mengubah suatu vektor yang berupa integer menjadi matrix dengan kelas biner. Untuk ilustrasinya bisa dilihat pada gambar ??

```
[Categories (5, object): [a, f, h, i, n]
>>> pd.set_option('display.max_colwidth', 40)
>>> fahmi = pd.Categorical(['c','l','e','c','o','p','a','s'])
>>> print(fahmi)
[c, l, e, c, o, p, a, s]
Categories (7, object): [a, c, e, i, o, p, s]
>>>
```

Gambar 6.31 Teori 7

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

Fungsi dari Sequential sebagai salah satu jenis model yang digunakan dalam perhitungan. Sequential ini membangun tumpukan linear yang berurutan. Untuk ilustrasi gambar sebagai betikut :

```
In [29]: model = Sequential([
    Dense(10, input_dim=mpg.shape[1]),
    Activation('relu'),
    Dense(10),
    Activation('softmax'),
    ])
```

Gambar 6.32 Teori 8

6.2.2 Praktek Program

1. Soal 1

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

```

19     plt.colorbar()
20     plt.title(song)
21     plt.tight_layout()
22     plt.show()

```

Kode di atas menjelaskan isi data GTZAN. Ini adalah kumpulan data yang berisi 10 genre lagu dengan masing-masing genre memiliki 100 lagu yang akan kami lakukan proses MFCC dan juga freesound yang hanya berisi konten lagu, jika GTZAN memiliki beberapa genre jika freesound hanya untuk 1 lagu dan disini kita membuat fungsi untuk membaca file audio dan outputnya sebagai plot, hasilnya adalah sebagai berikut:

```

In [1]: import librosa
In [1]: import librosa.feature
In [1]: import librosa.display
In [1]: import glob
In [1]: import os
In [1]: import matplotlib.pyplot as plt
In [1]: import numpy as np
In [1]: from keras.models import Dense, Activation
In [1]: from keras.models import Sequential
In [1]: from keras.utils import to_categorical
.....
def display_mfcc(song):
    Ys = librosa.load(song, sr=16000)[1]
    plt.figure(figsize=(10, 10))
    librosa.display.specshow(Ys, x_axis='time', y_axis='mel')
    plt.title(song)
    plt.colorbar()
    plt.show()

Using TensorFlow backend.
C:\Users\Arafat\PC\Kணால்லி\librosa\librosa\feature\tsne.py:55:
DeprecationWarning: tsne() is deprecated. In a future version of numpy, it will be understood as (Ys, 1) / (1, Ys).shape[1].
  return np.dot(Ys, Ys.T) / (2 * np.sum(Ys ** 2, axis=1)[:, np.newaxis])
C:\Users\Arafat\PC\Kணால்லி\librosa\librosa\feature\tsne.py:57:
DeprecationWarning: tsne() is deprecated. In a future version of numpy, it will be understood as (Ys, 1) / (1, Ys).shape[1].
  return np.dot(Ys, Ys.T) / (2 * np.sum(Ys ** 2, axis=1)[:, np.newaxis])
C:\Users\Arafat\PC\Kணால்லி\librosa\librosa\feature\tsne.py:59:
DeprecationWarning: tsne() is deprecated. In a future version of numpy, it will be understood as (Ys, 1) / (1, Ys).shape[1].
  return np.dot(Ys, Ys.T) / (2 * np.sum(Ys ** 2, axis=1)[:, np.newaxis])
C:\Users\Arafat\PC\Kணால்லி\librosa\librosa\feature\tsne.py:61:
DeprecationWarning: tsne() is deprecated. In a future version of numpy, it will be understood as (Ys, 1) / (1, Ys).shape[1].
  return np.dot(Ys, Ys.T) / (2 * np.sum(Ys ** 2, axis=1)[:, np.newaxis])

```

Gambar 6.33 Hasil Soal 1.

2. Soal 2

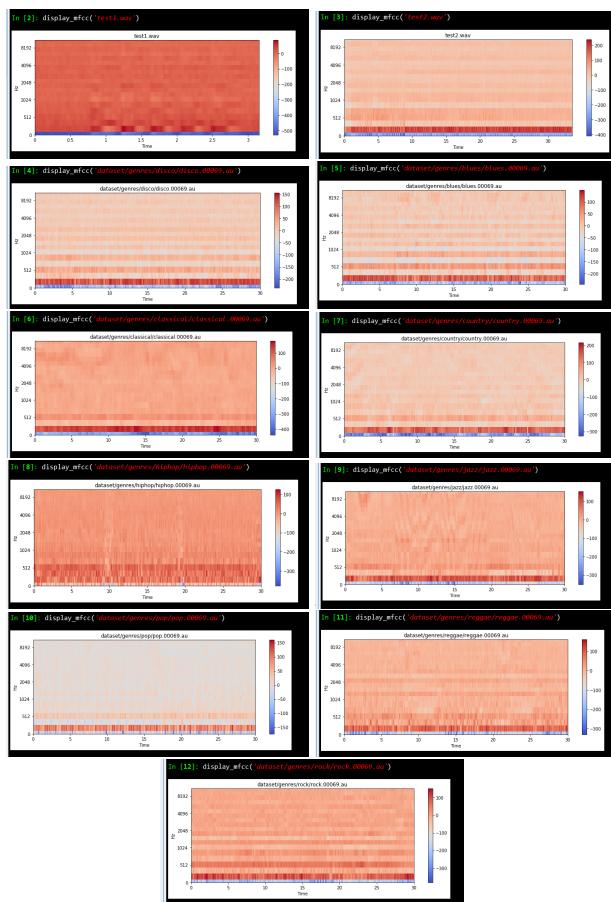
```

1 # In[2]: Soal Nomor 2
2 display_mfcc('test1.wav')
3 # In[2]: Soal Nomor 2
4 display_mfcc('test2.wav')
5 # In[2]: Soal Nomor 2
6 display_mfcc('dataset/genres/disco/disco.00069.au')
7 # In[2]: Soal Nomor 2
8 display_mfcc('dataset/genres/blues/blues.00069.au')
9 # In[2]: Soal Nomor 2
10 display_mfcc('dataset/genres/classical/classical.00069.au')
11 # In[2]: Soal Nomor 2
12 display_mfcc('dataset/genres/country/country.00069.au')
13 # In[2]: Soal Nomor 2
14 display_mfcc('dataset/genres/hiphop/hiphop.00069.au')
15 # In[2]: Soal Nomor 2
16 display_mfcc('dataset/genres/jazz/jazz.00069.au')
17 # In[2]: Soal Nomor 2
18 display_mfcc('dataset/genres/pop/pop.00069.au')
19 # In[2]: Soal Nomor 2
20 display_mfcc('dataset/genres/reggae/reggae.00069.au')
21 # In[2]: Soal Nomor 2
22 display_mfcc('dataset/genres/rock/rock.00069.au')

```

Kode di atas akan menampilkan hasil dari proses mfcc yang sudah dibuat fungsi pada soal 1, yaitu display_mfcc() dan akan menampilkan plot dari pembacaan

file audio. Berikut adalah hasil setelah saya lakukan running dan pembacaan file audio :



Gambar 6.34 Hasil Soal 2.

3. Soal 3

```

1 # In [3]: Soal Nomor 3
2
3 def extract_features_song(f):
4     y, _ = librosa.load(f)
5
6     # get Mel-frequency cepstral coefficients
7     mfcc = librosa.feature.mfcc(y)
8     # normalize values between -1,1 (divide by max)
9     mfcc /= np.amax(np.absolute(mfcc))
10
11    return np.ndarray.flatten(mfcc)[:25000]

```

Baris pertama itu untuk membuat fungsi extract_features_song(f). Pada baris kedua itu akan me-load data inputan dengan menggunakan librosa. Lalu selanjutnya untuk membuat sebuah fitur untuk mfcc dari y atau parameter inputan. Lalu akan me-return menjadi array dan akan mengambil 25000 data saja dari hasil vektorisasi dalam 1 lagu. Hasilnya adalah sebagai berikut :

```
In [13]: def extract_features_song(f):
    ....
    y, sr = librosa.load(f)
    ....
    # get Mel-frequency cepstral coefficients
    mfcc = librosa.feature.mfcc(y)
    mfcc -= np.mean(mfcc) # subtract mean
    mfcc /= np.max(np.abs(mfcc))
    ....
    return np.ndarray.flatten(mfcc)[:25000]
```

Gambar 6.35 Hasil Soal 3.

4. Soal 4

```
1 # In [4]: Soal Nomor 4
2
3 def generate_features_and_labels():
4     all_features = []
5     all_labels = []
6
7     genres = ['blues', 'classical', 'country', 'disco', 'hiphop',
8               'jazz', 'metal', 'pop', 'reggae', 'rock']
9     for genre in genres:
10         sound_files = glob.glob('dataset/genres/'+genre+'/*.au')
11         print('Processing %d songs in %s genre ... ' % (len(
12             sound_files), genre))
13         for f in sound_files:
14             features = extract_features_song(f)
15             all_features.append(features)
16             all_labels.append(genre)
17
18     # convert labels to one-hot encoding cth blues : 1000000000
19     # classic 0100000000
20     label_uniq_ids, label_row_ids = np.unique(all_labels,
21                                              return_inverse=True)#ke integer
22     label_row_ids = label_row_ids.astype(np.int32, copy=False)
23     onehot_labels = to_categorical(label_row_ids, len(
24         label_uniq_ids))#ke one hot
25
26     return np.stack(all_features), onehot_labels
```

Kode di atas dapat digunakan untuk melakukan fungsi yang sebelumnya telah kita lakukan. Kemudian di bagian genre yang disesuaikan dengan dataset nama folder. Untuk baris berikutnya akan mengulang genre folder dengan ekstensi .au. Maka itu akan memanggil fungsi ekstrak lagu. Setiap file dalam folder itu akan diekstraksi menjadi vektor dan akan ditambahkan ke fitur. Dan fungsi yang ditambahkan adalah untuk menumpuk file yang telah di-vektor-kan. Hasil kode tidak menampilkan output. Hasilnya adalah sebagai berikut :

```

In [14]: generate_features_and_labels():
    ...
    all_labels = []
    ...
    genres = ['blues', 'classical', 'country', 'disco', 'hiphop', 'jazz', 'metal',
    ...
    for genre in genres:
        song_files = glob.glob('dataset/genres/' + genre + '/*.mp3')
        for f in song_files:
            song = extract_features(f)
            all_labels.append(all_labels[-1])
            all_features.append(song)
    ...
    # convert labels to one-hot encoding (in blues : 10000000 classic 00000000
    label_rec_id = np.eye(10)[label_rec_ids[:, 0].copy()]
    label_rec_id = label_rec_id.astype(np.int16).copy()
    ...
    # convert labels to one-hot encoding (in blues : 10000000 classic 00000000
    labels = np.stack(all_labels), onehot_labels

```

Gambar 6.36 Hasil Soal 4.

5. Soal 5

```

1 # In [5]: Soal Nomor 5
2
3 features , labels = generate_features_and_labels()
# In [5]: Soal Nomor 5
4 print(np.shape(features))
5 print(np.shape(labels))

```

Kode diatas berfungsi untuk melakukan load variabel features dan labels. Mengapa memakan waktu yang lama ? Karena mesin akan melakukan vektorisasi terhadap semua file yang berada pada setiap foldernya, di sini terdapat 10 folder dengan masing-masing folder terdiri atas 100 buah lagu, setiap lagu tersebut akan dilakukan vektorisasi atau ekstraksi data menggunakan mfcc. Oleh karena itu, proses cukup memakan waktu. Hasilnya adalah sebagai berikut :

In [15]: features, labels = generate_features_and_labels() Processing 100 songs in blues genre... Processing 100 songs in classical genre... Processing 100 songs in country genre... Processing 100 songs in disco genre... Processing 100 songs in hiphop genre... Processing 100 songs in jazz genre... Processing 100 songs in metal genre... Processing 100 songs in pop genre... Processing 100 songs in reggae genre... Processing 100 songs in rock genre...	features float64 (1000, 2000) labels float64 (1000, 10)	Info	Type	Size	Value
					[0.0, 0.0, ..., 0.0, 0.0]
					[1.0, 0.0, ..., 0.0, 0.0]
					[0.0, 0.0, ..., 0.0, 0.0]

```

In [16]: print(np.shape(features))
In [16]: print(np.shape(labels))
(1000, 2000)
(1000, 10)

```

Gambar 6.37 Hasil Soal 5.

6. Soal 6

```

1 # In [6]: Soal Nomor 6
2 training_split = 0.8
# In [6]: Soal Nomor 6
3 alldata = np.column_stack(( features , labels ))
# In [6]: Soal Nomor 6
4 np.random.shuffle(alldata)
5 splitidx = int(len(alldata) * training_split)
6 train , test = alldata[:splitidx ,:], alldata[splitidx :,:]
# In [6]: Soal Nomor 6
7 print(np.shape(train))
8 print(np.shape(test))
# In [6]: Soal Nomor 6
9

```

```

13 train_input = train[:, :-10]
14 train_labels = train[:, -10:]
15 # In [6]: Soal Nomor 6
16 test_input = test[:, :-10]
17 test_labels = test[:, -10:]
18 # In [6]: Soal Nomor 6
19 print(np.shape(train_input))
20 print(np.shape(train_labels))

```

Kode diatas berfungsi untuk melakukan training split 80%. Karena supaya mesin dapat terus belajar tentang data baru, jadi ketika prediksi dibuat tentang data yang terlatih itu bisa mendapatkan persentase yang cukup bagus. Hasilnya adalah sebagai berikut :

In [17]: training_split = 0.8	In [18]: alldata = np.column_stack((features, labels))																																																
In [19]: np.random.shuffle(alldata)splitidx = int(len(alldata) * training_split)train, test = alldata[splitidx:,], alldata[splitidx:,:]	In [20]: print(np.shape(train))print(np.shape(test)) (800, 25010) (200, 25010)																																																
In [21]: train_input = train[:, :-10]train_labels = train[:, -10:]	In [22]: test_input = test[:, :-10]test_labels = test[:, -10:]																																																
	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>alldata</td> <td>float32</td> <td>(2000, 25010)</td> <td>[1.47864508 -0.79903827 -0.77314489 ... 0. 0.]</td> </tr> <tr> <td>features</td> <td>float32</td> <td>(2000, 2500)</td> <td>[1.47864508 -0.81619045 -0.75164417 ... -0.01833046 0.49327732]</td> </tr> <tr> <td>labels</td> <td>float32</td> <td>(2000, 10)</td> <td>[1. 0. 0. ... 0. 0. 0.]</td> </tr> <tr> <td>splitidx</td> <td>int</td> <td>1</td> <td>0</td> </tr> <tr> <td>train</td> <td>float32</td> <td>(200, 25010)</td> <td>[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]</td> </tr> <tr> <td>test</td> <td>float32</td> <td>(200, 25010)</td> <td>[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]</td> </tr> <tr> <td>train_input</td> <td>float32</td> <td>(200, 2500)</td> <td>[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]</td> </tr> <tr> <td>train_labels</td> <td>float32</td> <td>(200, 10)</td> <td>[1. 0. 0. ... 0. 0. 0.]</td> </tr> <tr> <td>test_input</td> <td>float32</td> <td>(200, 2500)</td> <td>[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]</td> </tr> <tr> <td>test_labels</td> <td>float32</td> <td>(200, 10)</td> <td>[1. 0. 0. ... 0. 0. 0.]</td> </tr> <tr> <td>training_split</td> <td>float</td> <td>1</td> <td>0.8</td> </tr> </tbody> </table>	Name	Type	Size	Value	alldata	float32	(2000, 25010)	[1.47864508 -0.79903827 -0.77314489 ... 0. 0.]	features	float32	(2000, 2500)	[1.47864508 -0.81619045 -0.75164417 ... -0.01833046 0.49327732]	labels	float32	(2000, 10)	[1. 0. 0. ... 0. 0. 0.]	splitidx	int	1	0	train	float32	(200, 25010)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]	test	float32	(200, 25010)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]	train_input	float32	(200, 2500)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]	train_labels	float32	(200, 10)	[1. 0. 0. ... 0. 0. 0.]	test_input	float32	(200, 2500)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]	test_labels	float32	(200, 10)	[1. 0. 0. ... 0. 0. 0.]	training_split	float	1	0.8
Name	Type	Size	Value																																														
alldata	float32	(2000, 25010)	[1.47864508 -0.79903827 -0.77314489 ... 0. 0.]																																														
features	float32	(2000, 2500)	[1.47864508 -0.81619045 -0.75164417 ... -0.01833046 0.49327732]																																														
labels	float32	(2000, 10)	[1. 0. 0. ... 0. 0. 0.]																																														
splitidx	int	1	0																																														
train	float32	(200, 25010)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]																																														
test	float32	(200, 25010)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]																																														
train_input	float32	(200, 2500)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]																																														
train_labels	float32	(200, 10)	[1. 0. 0. ... 0. 0. 0.]																																														
test_input	float32	(200, 2500)	[1.47864508 -0.5668732 -0.4586873 ... 0. 0. 1.]																																														
test_labels	float32	(200, 10)	[1. 0. 0. ... 0. 0. 0.]																																														
training_split	float	1	0.8																																														
In [23]: print(np.shape(train_input))print(np.shape(train_labels)) (800, 25000) (800, 10)																																																	

Gambar 6.38 Hasil Soal 6.

7. Soal 7

```

1 # In [7]: Soal Nomor 7
2 model = Sequential([
3     Dense(100, input_dim=np.shape(train_input)[1]),
4     Activation('relu'),
5     Dense(10),
6     Activation('softmax'),
7 ])

```

fungsi Sequential() ialah sebuah model untuk menentukan izin pada setiap neuron, di sini adalah 100 dense yang merupakan 100 neuron pertama dari data pelatihan. Fungsi dari relay itu sendiri adalah untuk mengaktifkan neuron atau input yang memiliki nilai maksimum. Sedangkan untuk dense 10 itu adalah output dari hasil neuron yang telah berhasil diaktifkan, untuk dense 10 diaktifkan menggunakan softmax. Hasilnya adalah sebagai berikut :

```
In [24]: model = Sequential([
    ...
    Dense(100, input_dim=train_input[1]),
    Activation('relu'),
    ...
    Dense(10),
    Activation('softmax'),
])

```

Gambar 6.39 Hasil Soal 7.

8. Soal 8

```
1 # In [8]: Soal Nomor 8
2 model.compile(optimizer='adam',
3                 loss='categorical_crossentropy',
4                 metrics=['accuracy'])
5 print(model.summary())
```

Model Compile di perjelas dengan gambar dibawah, Hasil output pada kode tersebut seperti gambar menjelaskan bahwa dense pertama itu memiliki 100 neurons dengan parameter sekitar 2 juta lebih dengan aktviasi 100, jadi untuk setiap neurons memiliki masing-masing 1 aktivasi. Sama halnya seperti dense 2 memiliki jumlah neurons sebanyak 10 dengan parameter 1010 dan jumlah aktivasinya 10 untuk setiap neurons tersebut dan total parameternya sekitar 2.5 juta data yang akan dilatih pada mesin tersebut.

```
In [25]: model.compile(optimizer='adam',
...
)
print(model.summary())
Model: "sequential_1"
Layer (Type)          Output Shape       Param #
dense_1 (Dense)      (None, 100)        2500100
activation_1 (Activation) (None, 100)        0
dense_2 (Dense)      (None, 10)         1010
activation_2 (Activation) (None, 10)        0
=====
Total params: 2,501,110
Trainable params: 2,501,110
Non-trainable params: 0
None
```

Gambar 6.40 Hasil Soal 8.

9. Soal 9

```
1 # In [9]: Soal Nomor 9
2 model.fit(train_input, train_labels, epochs=10, batch_size=32,
3            validation_split=0.2)
```

Kode tersebut berfungsi untuk melatih mesin dengan data training input dan training label. Epochs ini merupakan iterasi atau pengulangan berapa kali data tersebut akan dilakukan. Batch_size ini adalah jumlah file yang akan dilakukan pelatihan pada setiap 1 kali pengulangan. Sedangkan validation_split itu untuk menentukan persentase dari cross validation atau k-fold sebanyak 20% dari masing-masing data pengulangan, hasilnya adalah sebagai berikut :

```
In [10]: model.fit(train_input, train_labels, epochs=10, batch_size=10)
WARNING:tensorflow:From C:\Users\JAYA-MEPC\Anaconda\lib\site-packages\tensorflow\train\train.py:102: 
train_graph: deprecated_func_scope_wrapper is deprecated. Please use 
tf.function instead.
Train on 100 samples
Epoch 1/10
1/10 [0m0s] - loss: 2.0096 - accuracy: 0.2999 -
1/10 [0m0s] - loss: 1.8985 - val_accuracy: 0.2608 -
1/10 [0m0s] - loss: 1.7875 - val_accuracy: 0.2705 -
1/10 [0m0s] - loss: 1.6979 - val_accuracy: 0.4215 -
1/10 [0m0s] - loss: 1.5443 - val_accuracy: 0.4563 -
1/10 [0m0s] - loss: 1.4655 - val_accuracy: 0.4312 -
1/10 [0m0s] - loss: 1.4090 - val_accuracy: 0.4561 -
1/10 [0m0s] - loss: 1.3462 - val_accuracy: 0.4588 -
1/10 [0m0s] - loss: 1.2862 - val_accuracy: 0.5346 -
1/10 [0m0s] - loss: 1.2318 - val_accuracy: 0.5138 -
1/10 [0m0s] - loss: 1.1844 - accuracy: 0.4238 -
1/10 [0m0s] - loss: 1.1463 - val_accuracy: 0.4543 -
1/10 [0m0s] - loss: 1.1149 - val_accuracy: 0.4349 -
1/10 [0m0s] - loss: 1.0849 - val_accuracy: 0.4737 -
1/10 [0m0s] - loss: 1.0562 - val_accuracy: 0.4956 -
1/10 [0m0s] - loss: 1.0344 - accuracy: 0.4837 -
1/10 [0m0s] - loss: 1.0138 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.9949 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.9762 - accuracy: 0.4915 -
1/10 [0m0s] - loss: 0.9574 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.9402 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.9222 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.9042 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.8862 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.8682 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.8502 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.8322 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.8142 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.7962 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.7782 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.7602 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.7422 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.7242 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.7062 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.6882 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.6702 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.6522 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.6342 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.6162 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.5982 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.5802 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.5622 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.5442 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.5262 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.5082 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.4902 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.4722 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.4542 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.4362 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.4182 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.3992 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.3812 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.3632 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.3452 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.3272 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.3092 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.2912 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.2732 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.2552 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.2372 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.2192 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.2012 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.1832 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.1652 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.1472 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.1292 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.1112 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.0932 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.0752 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.0572 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.0392 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.0212 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.0032 - accuracy: 0.4956 -
1/10 [0m0s] - loss: 0.0000 - accuracy: 0.4956
```

Gambar 6.41 Hasil Soal 9.

10. Soal 10

```
1 # In[10]: Soal Nomor 10
2 loss, acc = model.evaluate(test_input, test_labels, batch_size=32)
3 # In[10]: Soal Nomor 10
4 print("Done!")
5 print("Loss: %.4f, accuracy: %.4f" % (loss, acc))
```

Fungsi evaluate atau evaluasi ini ialah untuk menguji data pengujian setiap file. Di sini ada prediksi yang hilang, artinya mesin memprediksi data, sedangkan untuk keseluruhan perjanjian sekitar 55%, hasilnya adalah sebagai berikut :

```
In [27]: print("Done!")
...: print("Loss: %.4f, accuracy: %.4f" % (loss, acc))
Done!
Loss: 1.4653, accuracy: 0.4900
```

Gambar 6.42 Hasil Soal 10.

11. Soal 11

```
1 # In[11]: Soal Nomor 11
2 model.predict(test_input[:1])
```

Fungsi Predict ialah untuk menghasilkan suatu nilai yang sudah di prediksi dari data training sebelumnya. Gambar dibawah ini menjelaskan file yang di jalankan tersebut termasuk ke dalam genre apa, hasilnya bisa dilihat pada gambar tersebut presentase yang paling besar yakni genre rock. Maka lagu tersebut termasuk ke dalam genre rock dengan perbandingan presentase hasil prediksi.

```
In [29]: model.predict(test_input[:1])
Out[29]: array([1.57373118e-02, 8.330395e-03, 6.3665556e-03, 1.3786874e-02,
   5.568286e-02, 6.7809211e-01, 2.7797746e-04, 2.0203577e-02,
   2.5181699e-02, 3.0326517e-02]), dtype=float32)
```

Gambar 6.43 Hasil Soal 11.

6.2.3 Penanganan Error

1. ScreenShoot Error

```
[1]: import librosa
... import librosa.feature
... import librosa.display
... import matplotlib.pyplot as plt
fb = librosa.FFmpegBackend()
fb.configure(logfile=None)
fb.start()

File "", line 1, in <module>
    import librosa
ModuleNotFoundError: No module named 'librosa'
```

Gambar 6.44 ModuleNotFoundError



Gambar 6.45 An error occurred while starting the kernel

2. Cara Penanganan Error

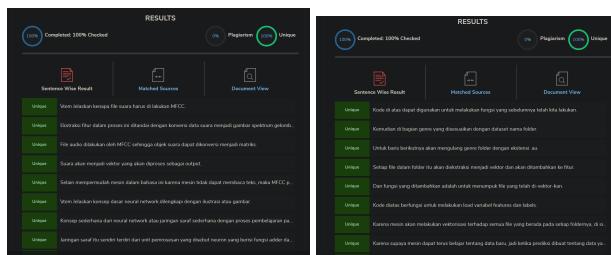
▪ ModuleNotFoundError

Error terdapat pada library yang tidak terbaca, karena library librosa belum di install, solusi nya ialah dengan menginstall library tersebut, pip install librosa.

▪ An error occurred while starting the kernel

Ini error yang tidak biasa karena error terdapat pada kernel, disini solusi nya ialah kita harus menginstall hdf5, yaitu dengan conda uninstall hdf5 dan conda install hdf5.

6.2.4 Bukti Tidak Plagiat



Gambar 6.46 Bukti Tidak Melakukan Plagiat Chapter 6

BAB 7

CHAPTER 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](#)