

CERDAS MENGUASAI PYTHON

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

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Listings

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan flask 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[?].

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

BAB 1

LIBRARY CSV DAN PANDAS

BAB 2

PRAKTEK LIBRARY CSV DAN PANDAS

BAB 3

PYSERIAL

BAB 4

PRAKTEK PYSERIAL

4.1 Kadek Diva Krishna Murti

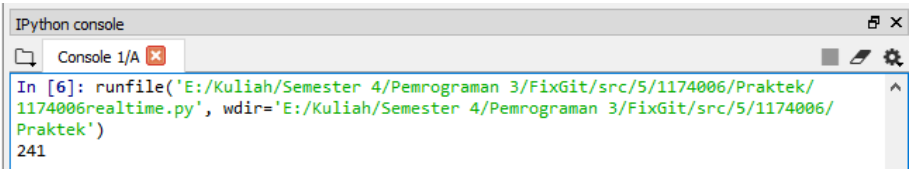
Ketrampilan Pemrograman

4.1.1 Soal No. 1

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data langsung dari arduino!

```
1 import serial
2
3 def getData():
4     ser = serial.Serial('COM5', 9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 getData()
```

Listing 4.1 Fungsi untuk mendapatkan data dari Arduino.



```
IPython console
Console 1/A
In [6]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006realtime.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
241
```

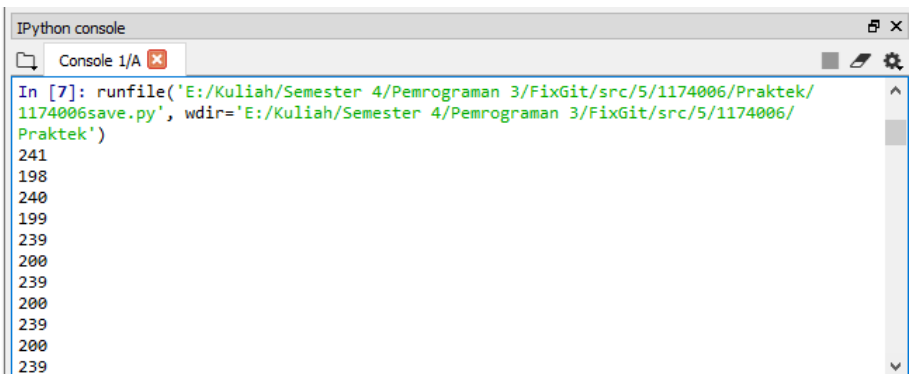
Gambar 4.1 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino.

4.1.2 Soal No. 2

Buatlah fungsi (file terpisah/library dengan nama NPMsave.py) untuk mendapatkan data langsung dari arduino dengan looping!

```
1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5', 9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getDataLoop()
```

Listing 4.2 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.



```
IPython console
Console 1/A
In [7]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006save.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
241
198
240
199
239
200
239
200
239
200
239
```

Gambar 4.2 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.

4.1.3 Soal No. 3

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv!

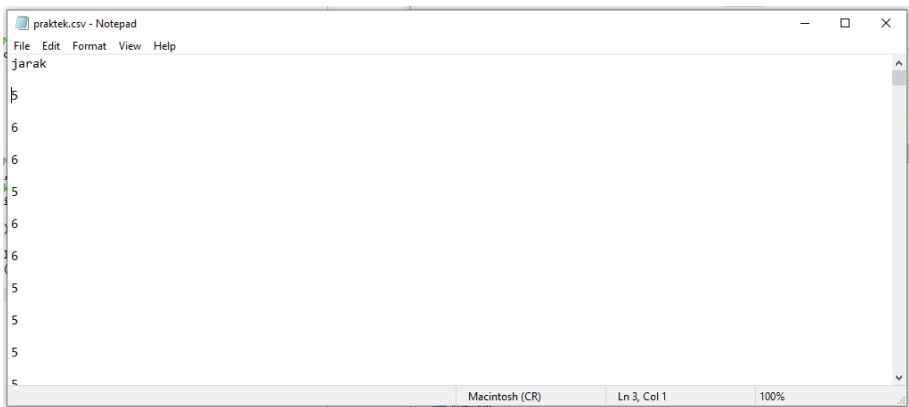
```
1 import serial
2 import csv
3
```

```

4 def writeCsv():
5     ser = serial.Serial('COM5', 9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = ser.readline().decode("utf-8").strip('\n').strip('\r')
13            writer.writerow({'jarak': data})
14
15 writeCsv()

```

Listing 4.3 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.



Gambar 4.3 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

4.1.4 Soal No. 4

Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi!

```

1 import csv
2
3 def readCsv():
4     with open('praktek.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['jarak'])
8
9 readCsv()

```

Listing 4.4 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.

```

IPython console
Console 1/A
In [12]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006csv.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
Out[12]:
5
6
6
5
6
6
5
5
5
5
5

```

Gambar 4.4 Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.

4.1.5 Kode Program Praktek

```

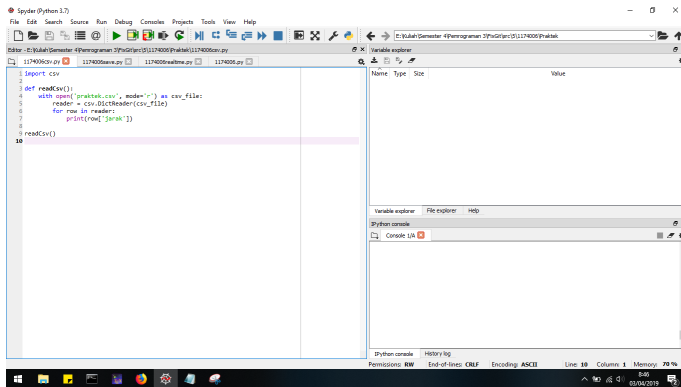
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
1 import serial
2
3 def getdata():
4     ser = serial.Serial('COM5', 9600)
5     print(ser.readline().decode("utf-8").strip("\n").strip("\r"))
6
7 getdata()
8
9 import csv
10
11 def writedata():
12     ser = serial.Serial('COM5', 9600)
13     with open('praktek.csv', 'w+', encoding='utf-8') as csv_file:
14         fieldnames = ['data']
15         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
16         writer.writeheader()
17         while (1):
18             data = ser.readline().decode("utf-8").strip("\n").strip("\r")
19             writer.writerow({'data': data})
20
21 writedata()
22
23
24

```

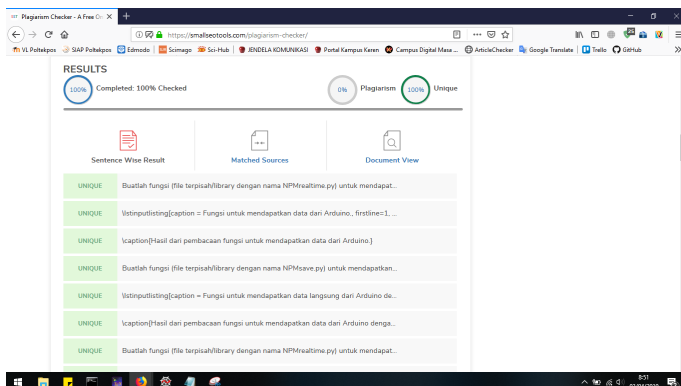
```

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
1 import serial
2
3 def getdata():
4     ser = serial.Serial('COM5', 9600)
5     print(ser.readline().decode("utf-8").strip("\n").strip("\r"))
6
7 getdata()
8
9

```



4.1.6 Cek Plagiat Praktek



Ketrampilan Penanganan Error

4.1.7 Soal No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Name Error** NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya

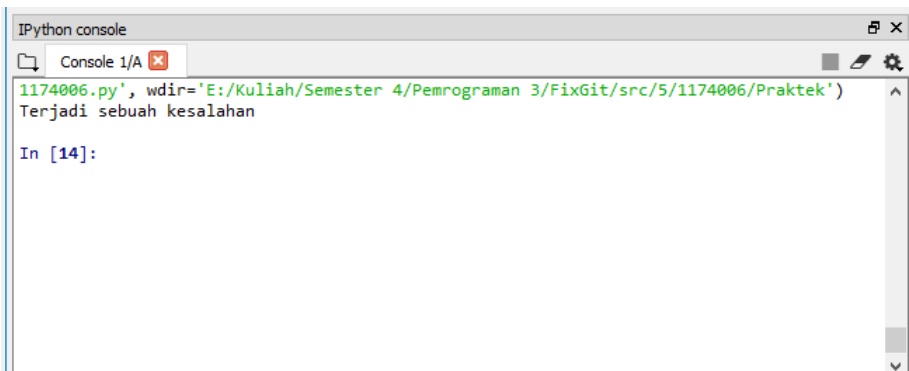
adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.

- Type Error TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

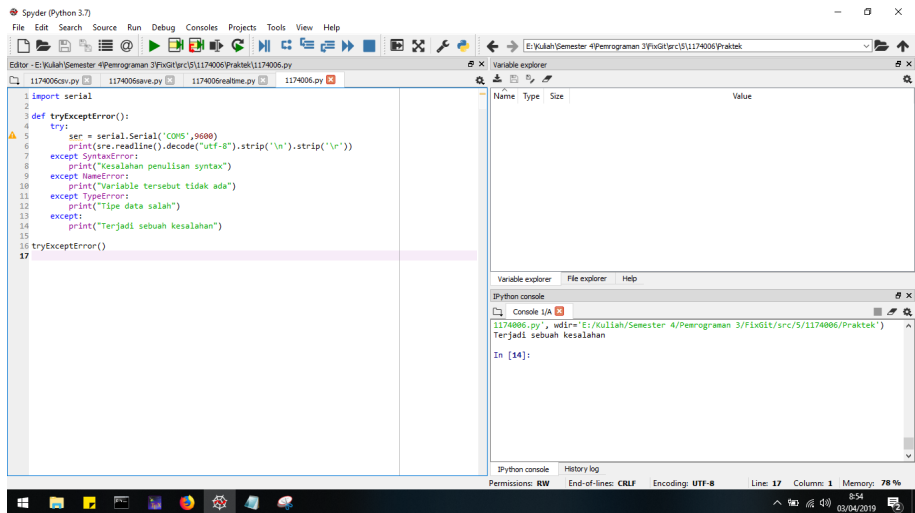
```
1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM5', 9600)
6         print(ser.readline().decode('utf-8').strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
9     except NameError:
10        print("Variable tersebut tidak ada")
11    except TypeError:
12        print("Tipe data salah")
13    except:
14        print("Terjadi sebuah kesalahan")
15
16 tryExceptError()
```

Listing 4.5 Fungsi untuk menanggulangi error menggunakan Try Except.

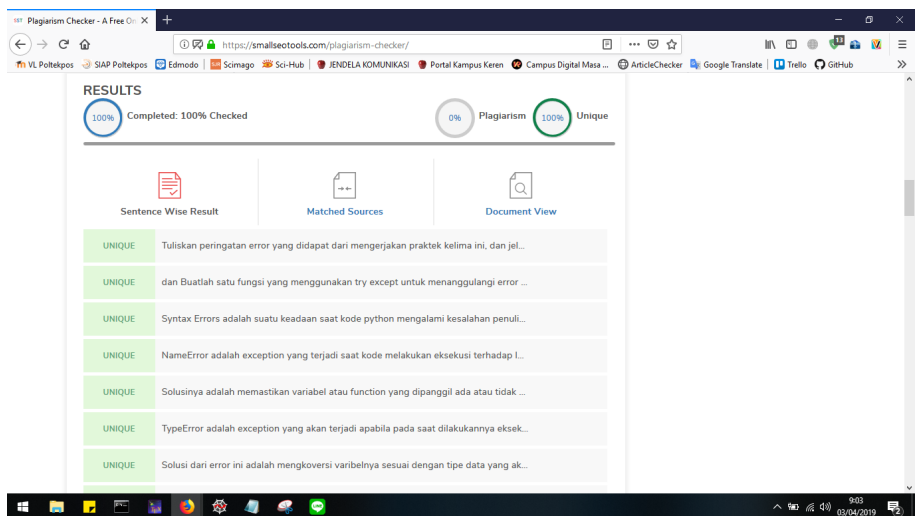


Gambar 4.5 Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.

4.1.8 Kode Program Penanganan Error



4.1.9 Cek Plagiat Penanganan Error



4.2 Muh. Rifky Prananda

Ketrampilan Pemrograman

4.2.1 Soal No. 1

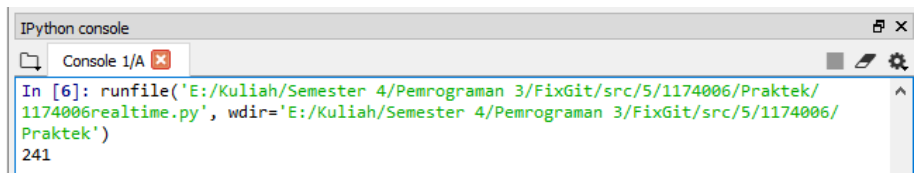
Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data langsung dari arduino!

```

1 import serial
2
3 def getData():
4     ser = serial.Serial('COM5',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7     getData()

```

Listing 4.6 Fungsi untuk mendapatkan data dari Arduino.



Gambar 4.6 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino.

4.2.2 Soal No. 2

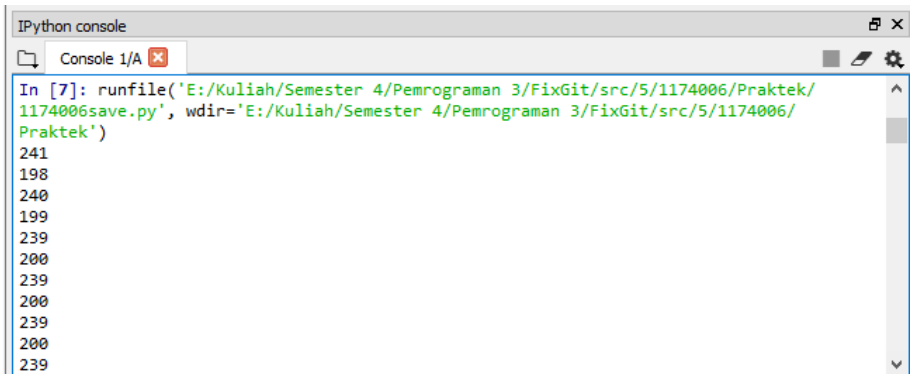
Buatlah fungsi (file terpisah/library dengan nama NPMsave.py) untuk mendapatkan data langsung dari arduino dengan looping!

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8     getDataLoop()

```

Listing 4.7 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.



```
IPython console
Console 1/A
In [7]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/
1174006save.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/
Praktek')
241
198
240
199
239
200
239
200
239
200
239
```

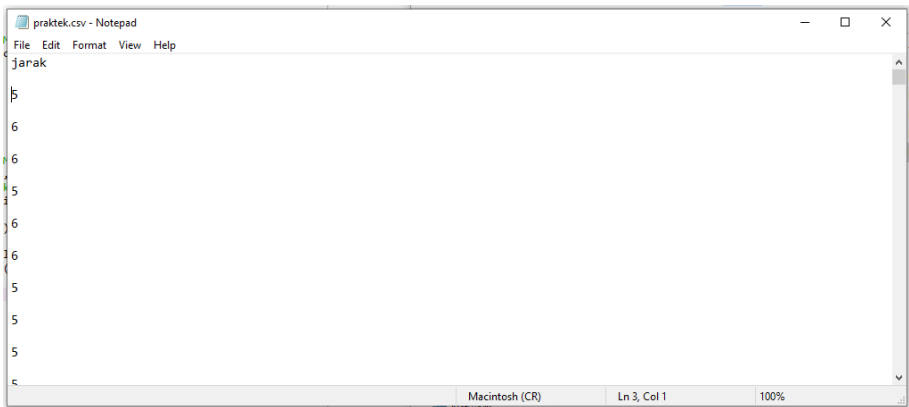
Gambar 4.7 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.

4.2.3 Soal No. 3

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv!

```
1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM5', 9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = ser.readline().decode("utf-8").strip('\n').strip('
\r')
13            writer.writerow({'jarak': data})
14
15 writeCsv()
```

Listing 4.8 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.



Gambar 4.8 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

4.2.4 Soal No. 4

Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi!

```

1 import csv
2
3 def readCsv():
4     with open('praktek.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['jarak'])
8
9 readCsv()
```

Listing 4.9 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.

```

IPython console
Console 1/A
In [12]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006.csv.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
Out[12]:
5
6
6
5
6
6
5
5
5
5
5

```

Gambar 4.9 Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.

4.2.5 Kode Program Praktek

```

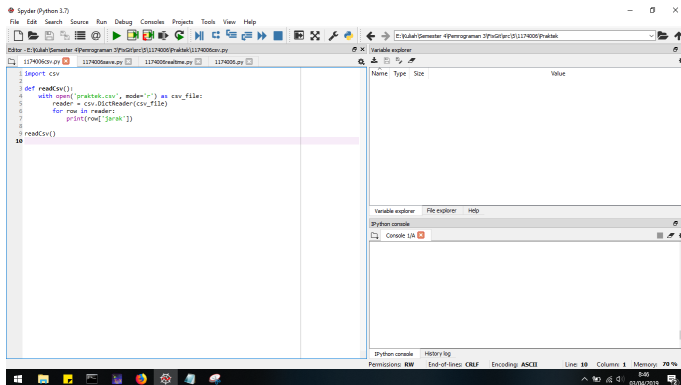
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Kuliah\Semester 4\Pemrograman 3\FixGit\src\5\1174006\Praktek
1174006.csv.py 1174006.csv.py 1174006.csv.py 1174006.csv.py
1 import serial
2
3 def getdata():
4     ser = serial.Serial('COM5', 9600)
5     print(ser.readline().decode("utf-8").strip("\n").strip("\r"))
6
7 getdata()
8
9 import csv
10
11 def writecsv():
12     ser = serial.Serial('COM5', 9600)
13     with open('praktek.csv', 'w+', encoding='utf-8') as csv_file:
14         fieldnames = ['data']
15         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
16         writer.writeheader()
17         while (1):
18             data = ser.readline().decode("utf-8").strip("\n").strip("\r")
19             writer.writerow({'data': data})
20             print(data)
21
22 writecsv()
23
24

```

```

Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Kuliah\Semester 4\Pemrograman 3\FixGit\src\5\1174006\Praktek
1174006.csv.py 1174006.csv.py 1174006.csv.py 1174006.csv.py
1 import serial
2
3 def getdata():
4     ser = serial.Serial('COM5', 9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip("\n").strip("\r"))
7
8 getdata()
9

```



Ketrampilan Penanganan Error

4.2.6 Soal No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Name Error** NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- **Type Error** TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkoversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

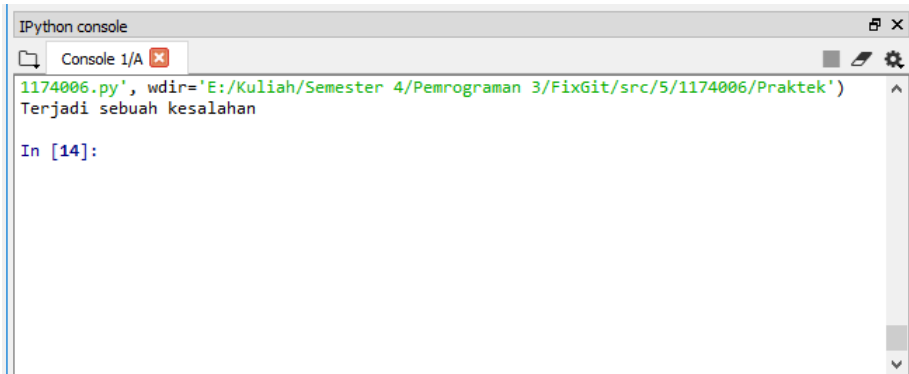
```
1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM5', 9600)
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
```

```

9     except NameError:
10         print("Variable tersebut tidak ada")
11     except TypeError:
12         print("Tipe data salah")
13     except:
14         print("Terjadi sebuah kesalahan")
15
16 tryExceptError()

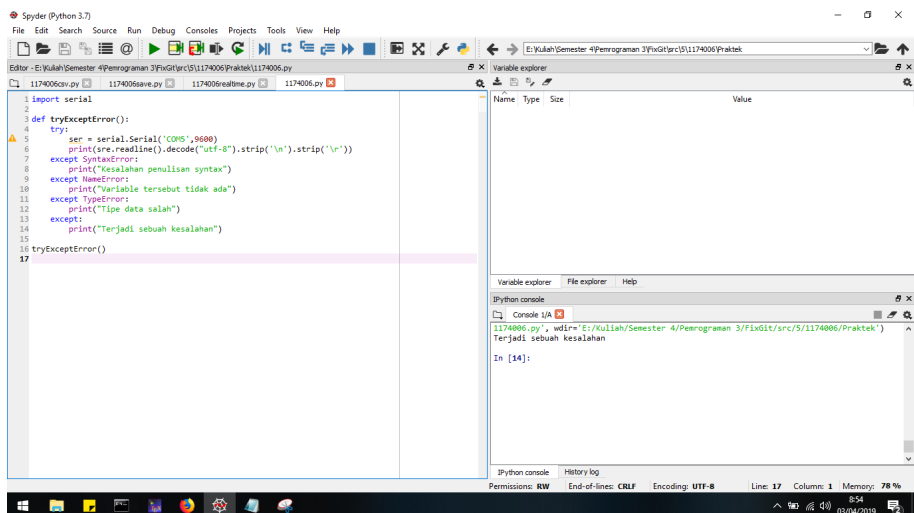
```

Listing 4.10 Fungsi untuk menanggulangi error menggunakan Try Except.

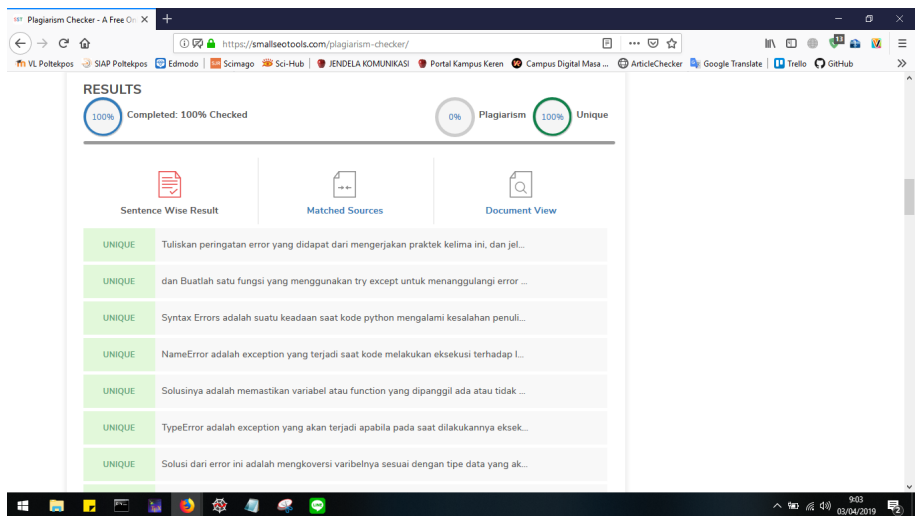


Gambar 4.10 Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.

4.2.7 Kode Program Penanganan Error



4.2.8 Cek Plagiat Penanganan Error



4.3 Damara Benedicta

Ketrampilan Pemrograman

4.3.1 Soal No. 1

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data langsung dari arduino!

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Apr  3 17:24:30 2019
4
5  @author: Damara
6  """
7
8  import serial
9
10 def getData():
11     ser = serial.Serial('COM5', 9600)
12     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
13
14 getData()
```

Listing 4.11 Fungsi untuk mendapatkan data dari Arduino.

4.3.2 Soal No. 2

Buatlah fungsi (file terpisah/library dengan nama NPMsave.py) untuk mendapatkan data langsung dari arduino dengan looping!

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getDataLoop()

```

Listing 4.12 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.

4.3.3 Soal No. 3

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv!

```

1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM5',9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = ser.readline().decode("utf-8").strip('\n').strip('\r')
13            writer.writerow({'jarak': data})
14
15 writeCsv()

```

Listing 4.13 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

4.3.4 Soal No. 4

Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi!

```

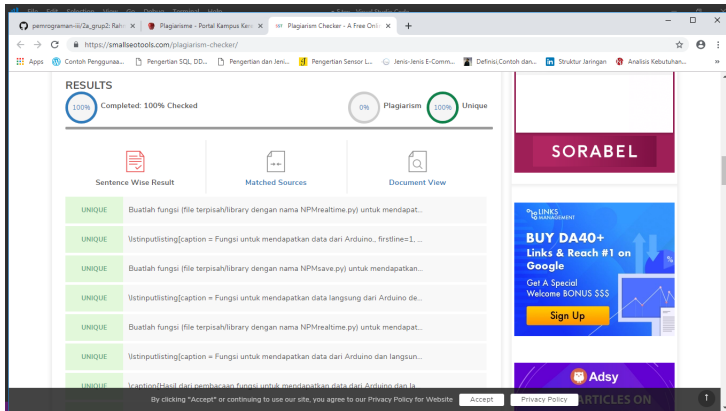
1 import csv
2
3 def readCsv():
4     with open('praktek.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['jarak'])
8

```

9 readCsv ()

Listing 4.14 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.

4.3.5 Cek Plagiat Praktek



Ketrampilan Penanganan Error

4.3.6 Soal No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Type Error** TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkoversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

```
1 import serial
2
3 def tryExceptError():
4     try:
```

```

5         ser = serial.Serial('COM5',9600)
6         print(sre.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
9     except NameError:
10        print("Variable tersebut tidak ada")
11    except TypeError:
12        print("Tipe data salah")
13    except:
14        print("Terjadi sebuah kesalahan")
15
16 tryExceptError()

```

Listing 4.15 Fungsi untuk menanggulangi error menggunakan Try Except.

4.4 Dwi Septiani Tsaniyah

4.4.1 Praktek

1. Soal 1

```

1 import serial
2
3 def ulang():
4     ser = serial.Serial('COM6',9600)
5     while(1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))

```

2. Soal 2

```

1 import serial
2
3 def ulang():
4     ser = serial.Serial('COM6',9600)
5     while(1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 ulang()

```

3. Soal 3

```

1 import serial
2
3 def ulang():
4     ser = serial.Serial('COM6',9600)
5     while(1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))

```

4. Soal 4

```

1 import csv
2
3 def bacacsv():
4     with open('uji.csv',mode='r') as csv_file:
5         baca = csv.DictReader(csv_file)
6         for row in baca:
7             print(row['jarak'])
8
9 bacacsv()

```

Ketrampilan Penanganan Error

4.4.2 Soal No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- Syntax Errors Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- Type Error TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkoversi variabelnya sesuai dengan tipe data yang akan digunakan.

cara untuk menangani eror yang dapat dilakukan adalah sebagai berikut:

```

1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM6',9600)
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7     except TypeError:
8         print("Terjadi ketidak-samaan type")
9
10 tryExceptError()

```

4.5 Muhammad Tomy Nur Maulidy

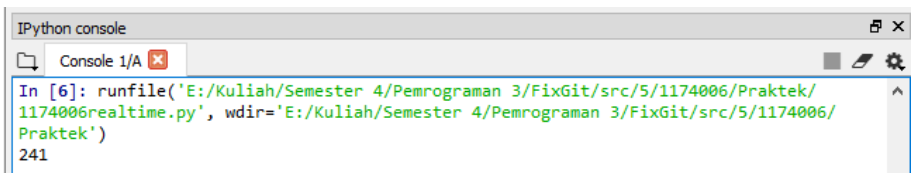
Ketrampilan Pemrograman

4.5.1 Soal No. 1

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data langsung dari arduino!

```
1 import serial
2
3 def getData():
4     ser = serial.Serial('COM5',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7 getData()
```

Listing 4.16 Fungsi untuk mendapatkan data dari Arduino.



Gambar 4.11 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino.

4.5.2 Soal No. 2

Buatlah fungsi (file terpisah/library dengan nama NPMsave.py) untuk mendapatkan data langsung dari arduino dengan looping!

```
1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8 getDataLoop()
```

Listing 4.17 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.

```
IPython console
Console 1/A
In [7]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006save.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
241
198
240
199
239
200
239
200
239
200
239
```

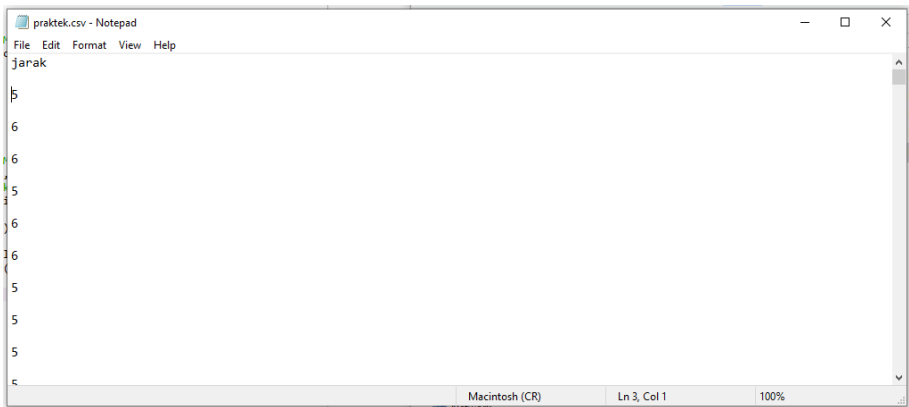
Gambar 4.12 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dengan looping.

4.5.3 Soal No. 3

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv!

```
1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM5', 9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = ser.readline().decode("utf-8").strip('\n').strip('
\r')
13            writer.writerow({'jarak': data})
14
15 writeCsv()
```

Listing 4.18 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.



Gambar 4.13 Hasil dari pembacaan fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

4.5.4 Soal No. 4

Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi!

```
1 import csv
2
3 def readCsv():
4     with open('praktek.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['jarak'])
8
9 readCsv()
```

Listing 4.19 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.


```

IPython console
Console 1/A
In [12]: runfile('E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek/1174006csv.py', wdir='E:/Kuliah/Semester 4/Pemrograman 3/FixGit/src/5/1174006/Praktek')
Out[12]:
5
6
6
5
6
6
5
5
5
5
5

```

Gambar 4.14 Hasil dari pembacaan fungsi untuk membaca file csv hasil arduino dan mengembalikan fungsi.

4.5.5 Kode Program Praktek

```

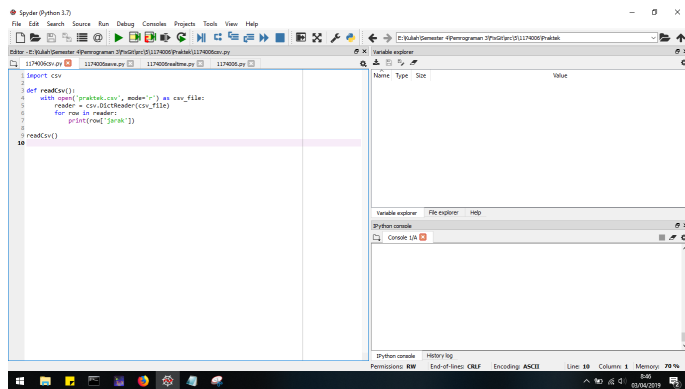
1 import serial
2
3 def getdata():
4     ser = serial.Serial('COM5', 9600)
5     print(ser.readline().decode("utf-8").strip("\n").strip("\r"))
6
7 getdata()
8
9 import serial
10 import csv
11
12 def writexml():
13     ser = serial.Serial('COM5', 9600)
14     with open('praktek.csv', mode='w') as csv_file:
15         fieldnames = ['id', 'data']
16         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
17         writer.writeheader()
18         while (1):
19             data = ser.readline().decode("utf-8").strip("\n").strip("\r")
20             writer.writerow({'id': '1', 'data': data})
21         print('')
22
23 writexml()
24

```

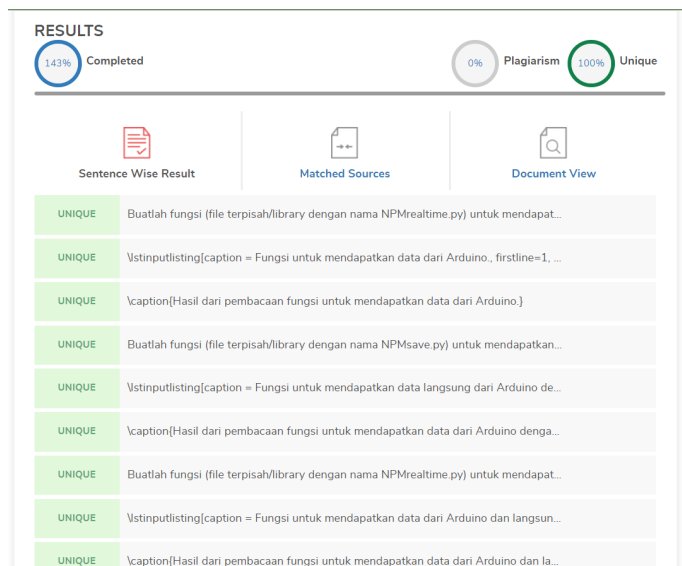
```

1 import serial
2
3 def getdata():
4     ser = serial.Serial('COM5', 9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip("\n").strip("\r"))
7
8 getdata()
9

```



4.5.6 Cek Plagiat Praktek



Ketrampilan Penanganan Error

4.5.7 Soal No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Name Error** NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- **Type Error** TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

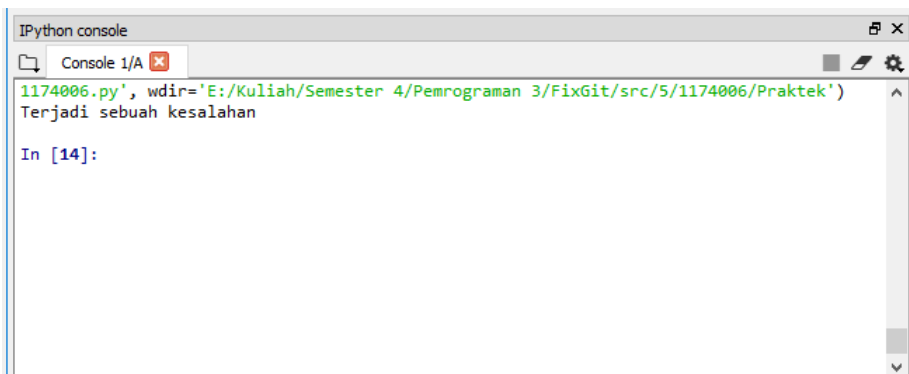
Fungsi yang menggunakan try except untuk menanggulangi error.

```

1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM5', 9600)
6         print(sre.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
9     except NameError:
10        print("Variable tersebut tidak ada")
11    except TypeError:
12        print("Tipe data salah")
13    except:
14        print("Terjadi sebuah kesalahan")
15
16 tryExceptError()

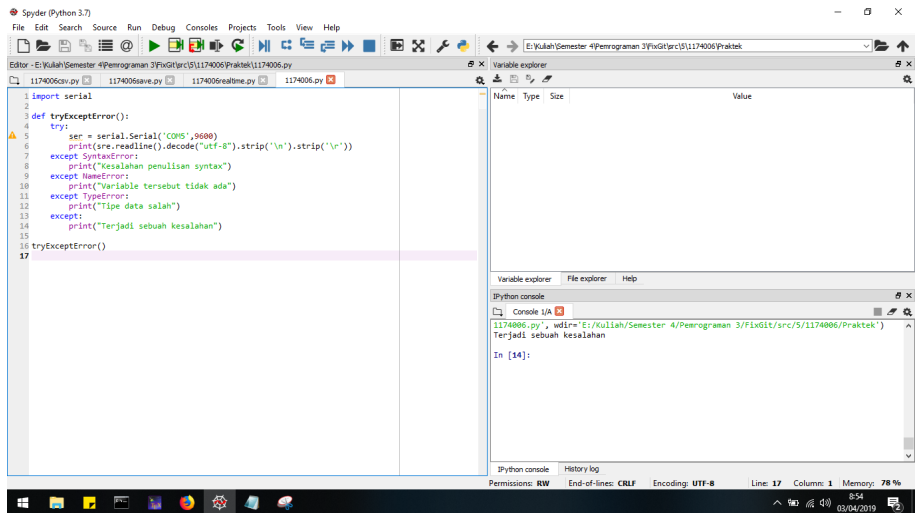
```

Listing 4.20 Fungsi untuk menanggulangi error menggunakan Try Except.

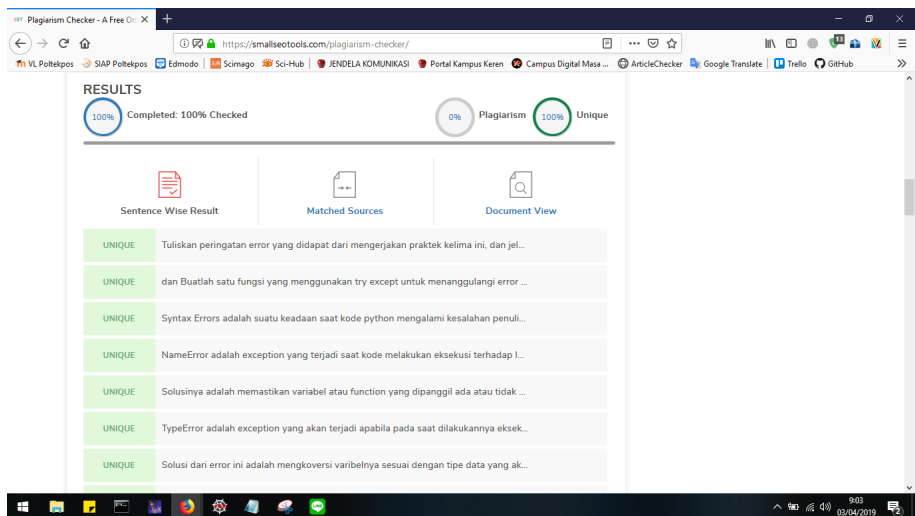


Gambar 4.15 Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.

4.5.8 Kode Program Penanganan Error



4.5.9 Cek Plagiat Penanganan Error



4.6 Felix Setiawan Lase

Ketrampilan Pemrograman

4.6.1 Soal No. 1

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data langsung dari arduino!

```

1 import serial
2
3 def getData():
4     ser = serial.Serial('COM5',9600)
5     print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
6
7     getData()
```

Listing 4.21 Fungsi untuk mendapatkan data dari Arduino.

4.6.2 Soal No. 2

Buatlah fungsi (file terpisah/library dengan nama NPMsave.py) untuk mendapatkan data langsung dari arduino dengan looping!

```

1 import serial
2
3 def getDataLoop():
4     ser = serial.Serial('COM5',9600)
5     while (1):
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7
8     getDataLoop()
```

Listing 4.22 Fungsi untuk mendapatkan data langsung dari Arduino dengan looping.

4.6.3 Soal No. 3

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv!

```

1 import serial
2 import csv
3
4 def writeCsv():
5     ser = serial.Serial('COM5',9600)
6     with open('praktek.csv', mode='w') as csv_file:
7         fieldnames = ['jarak']
8         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
9
10        writer.writeheader()
11        while (1):
12            data = ser.readline().decode("utf-8").strip('\n').strip(
13                '\r')
14            writer.writerow({'jarak': data})
15    writeCsv()
```

Listing 4.23 Fungsi untuk mendapatkan data dari Arduino dan langsung ditulis kedalam file CSV.

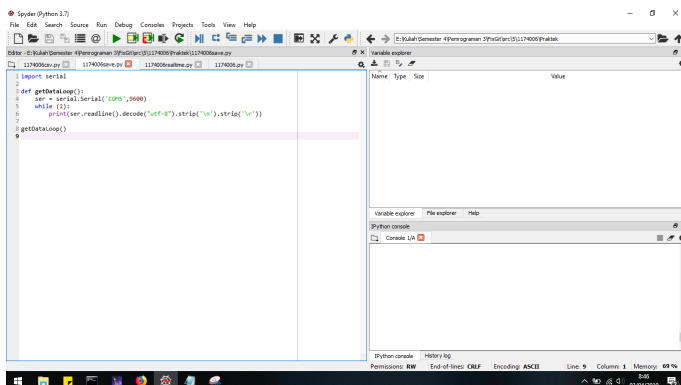
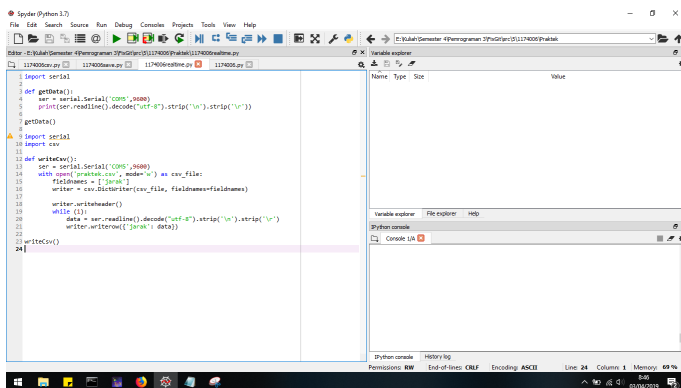
4.6.4 Soal No. 4

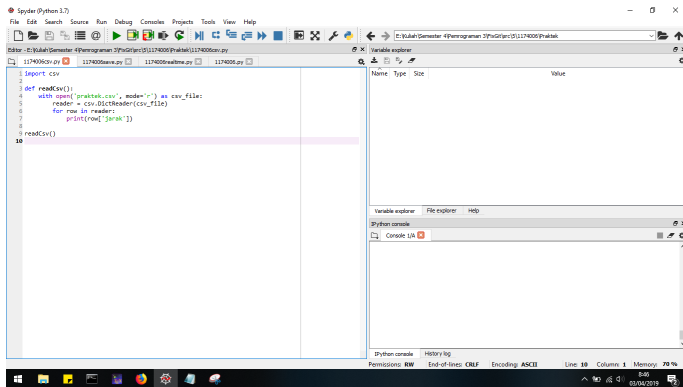
Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi!

```
1 import csv
2
3 def readCsv():
4     with open('praktek.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row['jarak'])
8
9 readCsv()
```

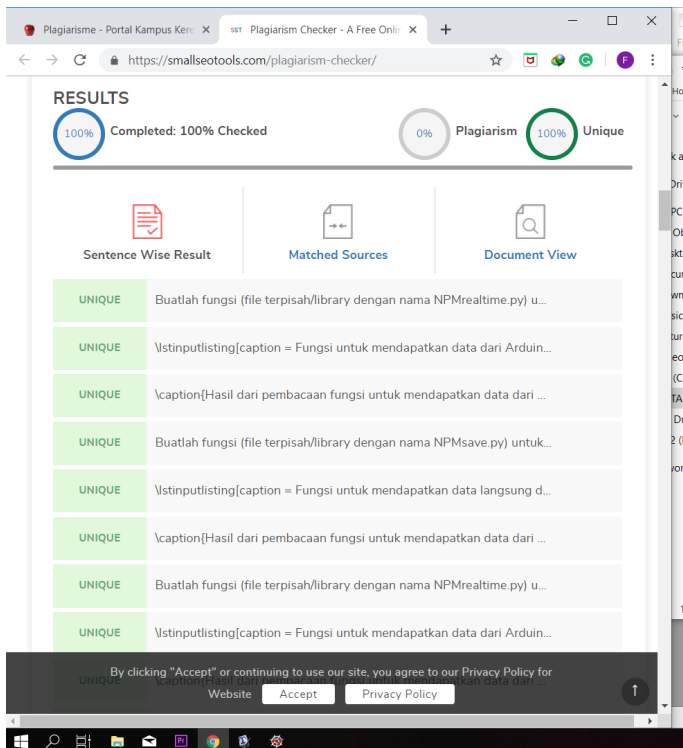
Listing 4.24 Fungsi untuk membaca file CSV hasil Arduino dan mengembalikan fungsi.

4.6.5 Kode Program Praktek





4.6.6 Cek Plagiat Praktek



Ketrampilan Penanganan Error

4.6.7 Soal No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

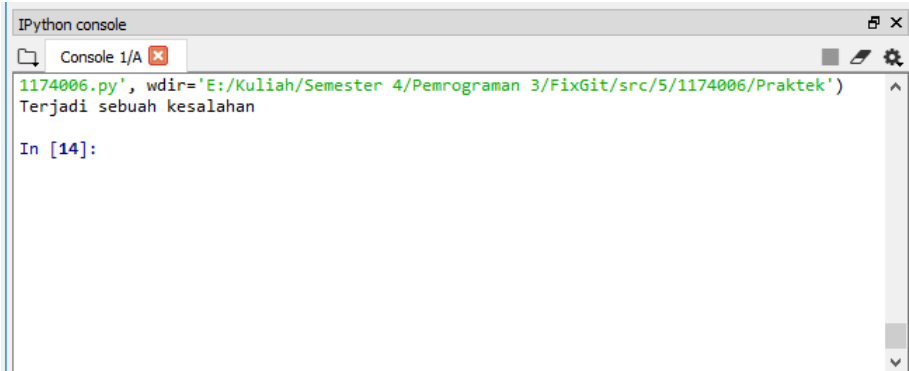
Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Name Error** NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- **Type Error** TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

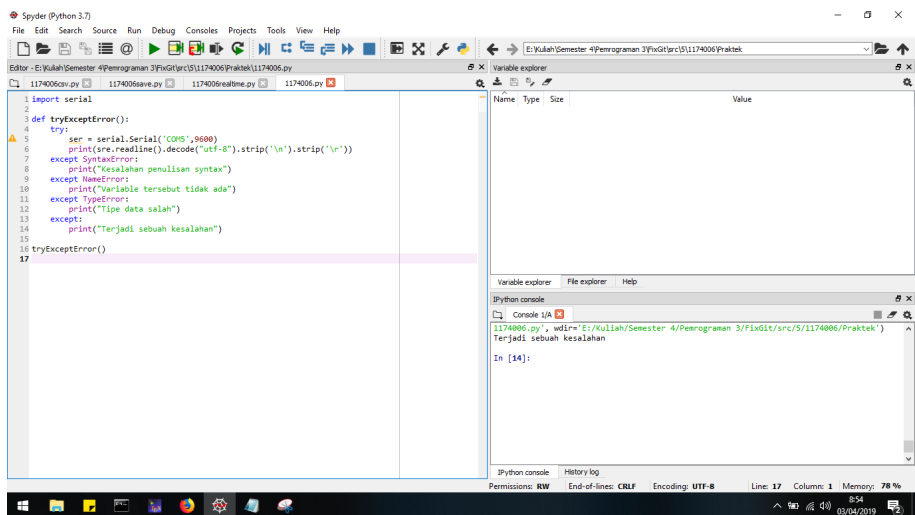
```
1 import serial
2
3 def tryExceptError():
4     try:
5         ser = serial.Serial('COM5', 9600)
6         print(sre.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
9     except NameError:
10        print("Variable tersebut tidak ada")
11    except TypeError:
12        print("Tipe data salah")
13    except:
14        print("Terjadi sebuah kesalahan")
15
16 tryExceptError()
```

Listing 4.25 Fungsi untuk menanggulangi error menggunakan Try Except.



Gambar 4.16 Hasil pembacaan fungsi untuk menanggulangi error menggunakan Try Except.

4.6.8 Kode Program Penanganan Error



4.6.9 Cek Plagiat Penanganan Error

The screenshot shows a web browser window with the URL <https://smallseotools.com/plagiarism-checker/>. The page displays the results of a plagiarism check. At the top, it says "RESULTS" and "Completed: 100% Checked". There are three circular progress indicators: "100%" (blue), "0%" (grey) for "Plagiarism", and "100%" (green) for "Unique". Below these are three tabs: "Sentence Wise Result", "Matched Sources", and "Document View". The "Sentence Wise Result" tab is active, showing a list of sentences, each marked as "UNIQUE".

Result	Sentence
UNIQUE	Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) u...
UNIQUE	\stinputlisting[caption = Fungsi untuk mendapatkan data dari Arduin...
UNIQUE	\caption{Hasil dari pembacaan fungsi untuk mendapatkan data dari ...
UNIQUE	Buatlah fungsi (file terpisah/library dengan nama NPMsave.py) untuk...
UNIQUE	\stinputlisting[caption = Fungsi untuk mendapatkan data langsung d...
UNIQUE	\caption{Hasil dari pembacaan fungsi untuk mendapatkan data dari ...
UNIQUE	Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) u...
UNIQUE	\stinputlisting[caption = Fungsi untuk mendapatkan data dari Arduin...

At the bottom of the page, there is a dark banner with the text: "By clicking 'Accept' or continuing to use our site, you agree to our Privacy Policy for Website". There are buttons for "Accept" and "Privacy Policy".

4.7 Muhammad Fahmi

Ketrampilan Pemrograman

4.7.1 Soal No. 1

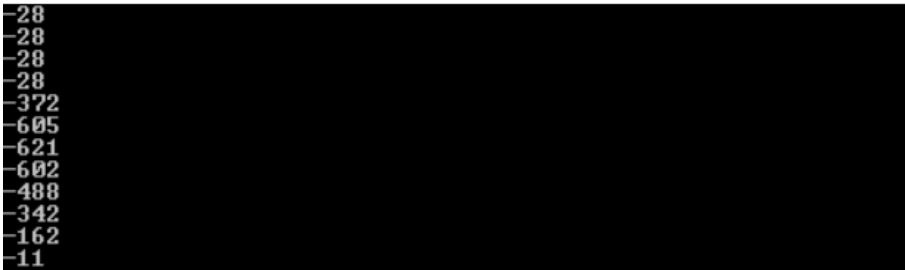
Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data langsung dari arduino. Namun disini saya memakai alat Mindwave untuk pengambilan sinyal gelombang otak.

```

1 import mindwave,time
2
3 def bacadata():
4     headset = mindwave.Headset('COM4', '1425')
5     time.sleep(1)
6
7     headset.connect()
8     print "Connecting..."
9
10    headset.status != 'connected':
11        time.sleep(1)
12        headset.status == 'standby':
13            headset.connect()
14            print "Retrying connect..."
15        print "Connected."
16
17
18    print headset.raw_value
19    time.sleep(1)
20
21 bacadata()

```

Listing 4.26 Soal no.1



```

28
28
28
28
372
605
621
602
488
342
162
11

```

Gambar 4.17 Hasil Soal no.1

4.7.2 Soal No. 2

Buatlah fungsi (file terpisah/library dengan nama NPMSave.py) untuk mendapatkan data langsung dari arduino dengan looping.

```

1 import mindwave,time
2
3 def bacadataloop():
4     headset = mindwave.Headset('COM4', '1425')
5     time.sleep(1)
6
7     headset.connect()
8     print "Connecting..."
9
10    while headset.status != 'connected':
11        time.sleep(1)
12        if headset.status == 'standby':

```

```

13     headset.connect()
14     print "Retrying connect..."
15     print "Connected."
16
17     while True:
18         print headset.raw_value
19         time.sleep(1)
20
21 bacadataloop()

```

Listing 4.27 Soal no.2

```

28
28
28
28
372
605
621
602
488
342
162
11
3
3
10
10
4
54
117
142
119
22
102
34
34

```

Gambar 4.18 Soal no.2

4.7.3 Soal No. 3

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendapatkan data dari arduino dan langsung ditulis kedalam file csv!

```

1 import mindwave,time, csv
2
3 def hasilcsv():
4     headset = mindwave.Headset('COM4', '1425')
5     time.sleep(1)
6     with open('hasil.csv', mode='w') as csv_file:
7         writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
8
9     headset.connect()
10    print "Connecting..."
11
12    while headset.status != 'connected':
13        time.sleep(1)

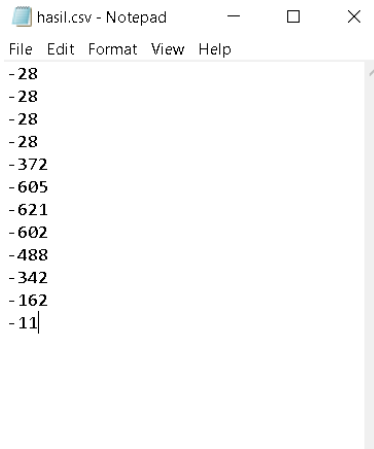
```

```

14     if headset.status == 'standby':
15         headset.connect()
16         print "Retrying connect..."
17     print "Connected."
18
19     while True:
20         print headset.raw_value
21         time.sleep(1)
22
23 hasilcsv()

```

Listing 4.28 Soal no.3



Gambar 4.19 Soal no.3

4.7.4 Soal No. 4

Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca file csv hasil arduino dan mengembalikan ke fungsi.

```

1 import csv
2
3 def bacacsv():
4     with open('hasil.csv', mode='r') as csv_file:
5         reader = csv.DictReader(csv_file)
6         for row in reader:
7             print(row[0])
8
9 bacacsv()

```

Listing 4.29 Soal no.4

```

28
28
28
28
372
605
621
602
488
342
162
11

```

Gambar 4.20 Soal no.4

4.7.5 Kode Program Mindwave.py

```

1  import select, serial, threading
2
3  # Byte codes
4  CONNECT          = '\xc0'
5  DISCONNECT       = '\xc1'
6  AUTOCONNECT      = '\xc2'
7  SYNC             = '\xaa'
8  EXCODE           = '\x55'
9  POOR_SIGNAL      = '\x02'
10 ATTENTION        = '\x04'
11 MEDITATION       = '\x05'
12 BLINK            = '\x16'
13 HEADSET_CONNECTED = '\xd0'
14 HEADSET_NOT_FOUND = '\xd1'
15 HEADSET_DISCONNECTED = '\xd2'
16 REQUEST_DENIED   = '\xd3'
17 STANDBY_SCAN     = '\xd4'
18 RAW_VALUE        = '\x80'
19
20 # Status codes
21 STATUS_CONNECTED = 'connected'
22 STATUS_SCANNING  = 'scanning'
23 STATUS_STANDBY  = 'standby'
24
25 class Headset(object):
26     """
27     A MindWave Headset
28     """
29
30     class DongleListener(threading.Thread):
31         """
32         Serial listener for dongle device.
33         """
34         def __init__(self, headset, *args, **kwargs):
35             """Set up the listener device."""
36             self.headset = headset
37             super(Headset.DongleListener, self).__init__(*args, **kwargs)
38
39         def run(self):
40             """Run the listener thread."""
41             s = self.headset.dongle
42
43             # Re-apply settings to ensure packet stream
44             s.write(DISCONNECT)

```

```

45 d = s.getSettingsDict()
46 for i in xrange(2):
47     d['rtscts'] = not d['rtscts']
48     s.applySettingsDict(d)
49
50 while True:
51     # Begin listening for packets
52     try:
53         if s.read() == SYNC and s.read() == SYNC:
54             # Packet found, determine plength
55             while True:
56                 plength = ord(s.read())
57                 if plength != 170:
58                     break
59                 if plength > 170:
60                     continue
61
62             # Read in the payload
63             payload = s.read(plength)
64
65             # Verify its checksum
66             val = sum(ord(b) for b in payload[:-1])
67             val &= 0xff
68             val = ~val & 0xff
69             chksum = ord(s.read())
70
71             #if val == chksum:
72                 if True: # ignore bad checksums
73                     self.parse_payload(payload)
74             except (select.error, OSError):
75                 break
76             except serial.SerialException:
77                 s.close()
78                 break
79
80 def parse_payload(self, payload):
81     """Parse the payload to determine an action."""
82     while payload:
83         # Parse data row
84         excode = 0
85         try:
86             code, payload = payload[0], payload[1:]
87         except IndexError:
88             pass

```

```

89 while code == EXCODE:
90     # Count excode bytes
91     excode += 1
92     try:
93         code, payload = payload[0], payload[1:]
94     except IndexError:
95         pass
96 if ord(code) < 0x80:
97     # This is a single-byte code
98     try:
99         value, payload = payload[0], payload[1:]
100     except IndexError:
101         pass
102 if code == POOR_SIGNAL:
103     # Poor signal
104     old_poor_signal = self.headset.poor_signal
105     self.headset.poor_signal = ord(value)
106     if self.headset.poor_signal > 0:
107         if old_poor_signal == 0:
108             for handler in \
109                 self.headset.poor_signal_handlers:
110                 handler(self.headset,
111                     self.headset.poor_signal)
112         else:
113             if old_poor_signal > 0:
114                 for handler in \
115                     self.headset.good_signal_handlers:
116                     handler(self.headset,
117                         self.headset.poor_signal)
118 elif code == ATTENTION:
119     # Attention level
120     self.headset.attention = ord(value)
121     for handler in self.headset.attention_handlers:
122         handler(self.headset, self.headset.attention)
123 elif code == MEDITATION:
124     # Meditation level
125     self.headset.meditation = ord(value)
126     for handler in self.headset.meditation_handlers:
127         handler(self.headset, self.headset.meditation)
128 elif code == BLINK:
129     # Blink strength
130     self.headset.blink = ord(value)
131     for handler in self.headset.blink_handlers:
132         handler(self.headset, self.headset.blink)

```



```

else:
    # This is a multi-byte code
    try:
        vlength, payload = ord(payload[0]), payload[1:]
    except IndexError:
        continue
    value, payload = payload[:vlength], payload[vlength:]
    # Multi-byte EEG and Raw Wave codes not included
    # Raw Value added due to Mindset Communications Protocol
    if code == RAW_VALUE:
        try:
            anu = value[0]
            itu = value[1]
        except IndexError:
            anu = ""
            itu = ""
        raw=ord(anu)*256+ord(itu)
        if (raw>=32768):
            raw=raw-65536
        self.headset.raw_value = raw
        for handler in self.headset.raw_value_handlers:
            handler(self.headset, self.headset.raw_value)
    if code == HEADSET_CONNECTED:
        # Headset connect success
        run_handlers = self.headset.status != STATUS_CONNECTED
        self.headset.status = STATUS_CONNECTED
        self.headset.headset_id = value.encode('hex')
        if run_handlers:
            for handler in \
                self.headset.headset_connected_handlers:
                handler(self.headset)
    elif code == HEADSET_NOT_FOUND:
        # Headset not found
        if vlength > 0:
            not_found_id = value.encode('hex')
            for handler in \
                self.headset.headset_notfound_handlers:
                handler(self.headset, not_found_id)
        else:
            for handler in \
                self.headset.headset_notfound_handlers:
                handler(self.headset, None)
    elif code == HEADSET_DISCONNECTED:
        # Headset disconnected

```

```

177         headset_id = value.encode('hex')
178         for handler in \
179             self.headset.headset_disconnected_handlers:
180             handler(self.headset, headset_id)
181     elif code == REQUEST_DENIED:
182         # Request denied
183         for handler in self.headset.request_denied_handlers:
184             handler(self.headset)
185     elif code == STANDBY_SCAN:
186         # Standby/Scan mode
187         try:
188             byte = ord(value[0])
189         except IndexError:
190             byte = None
191         if byte:
192             run_handlers = (self.headset.status !=
193                             STATUS_SCANNING)
194             self.headset.status = STATUS_SCANNING
195             if run_handlers:
196                 for handler in self.headset.scanning_handlers:
197                     handler(self.headset)
198         else:
199             run_handlers = (self.headset.status !=
200                             STATUS_STANDBY)
201             self.headset.status = STATUS_STANDBY
202             if run_handlers:
203                 for handler in self.headset.standby_handlers:
204                     handler(self.headset)
205
206
207 def __init__(self, device, headset_id=None, open_serial=True):
208     """Initialize the headset."""
209     # Initialize headset values
210     self.dongle = None
211     self.listener = None
212     self.device = device
213     self.headset_id = headset_id
214     self.poor_signal = 255
215     self.attention = 0
216     self.meditation = 0
217     self.blink = 0
218     self.raw_value = 0
219     self.status = None
220

```

```

230         self.headset_disconnected_handlers = []
231         self.request_denied_handlers = []
232         self.scanning_handlers = []
233         self.standby_handlers = []
234
235         # Open the socket
236         if open_serial:
237             self.serial_open()
238
239     def connect(self, headset_id=None):
240         """Connect to the specified headset id."""
241         if headset_id:
242             self.headset_id = headset_id
243         else:
244             headset_id = self.headset_id
245             if not headset_id:
246                 self.autoconnect()
247             return
248         self.dongle.write(''.join([CONNECT, headset_id.decode('hex')]))
249
250     def autoconnect(self):
251         """Automatically connect device to headset."""
252         self.dongle.write(AUTOCONNECT)
253
254     def disconnect(self):
255         """Disconnect the device from the headset."""
256         self.dongle.write(DISCONNECT)
257
258     def serial_open(self):
259         """Open the serial connection and begin listening for data."""
260         # Establish serial connection to the dongle
261         if not self.dongle or not self.dongle.isOpen():
262             self.dongle = serial.Serial(self.device, 115200)
263
264         # Begin listening to the serial device
265         if not self.listener or not self.listener.isAlive():
266             self.listener = self.DongleListener(self)
267             self.listener.daemon = True
268             self.listener.start()
269
270     def serial_close(self):
271         """Close the serial connection."""
272         self.dongle.close()
273

```

Ketrampilan Penanganan Error

4.7.6 Soal No. 1

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan dimana saat kode python mengalami kesalahan penulisan. Solusinya adalah dengan memperbaiki penulisan kode yang salah.
- **Name Error** NameError adalah exception yang terjadi pada saat kode melakukan eksekusi terhadap suatu local name atau global name yang tidak terdefinisi. Solusinya adalah dengan memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.

- Type Error TypeError adalah exception yang akan terjadi apabila pada saat melakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error tersebut adalah dengan mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

```
1 import serial
2
3 def error():
4     try:
5         ser = serial.Serial('COM5', 9600)
6         print(ser.readline().decode("utf-8").strip('\n').strip('\r'))
7     except SyntaxError:
8         print("Kesalahan penulisan syntax")
9     except NameError:
10        print("Variable tersebut tidak ada")
11    except TypeError:
12        print("Tipe data salah")
13    except:
14        print("Terjadi sebuah kesalahan")
15
16 error()
```

Listing 4.30 Fungsi untuk menanggulangi error menggunakan Try Except

4.7.7 Cek Plagiarisme

RESULTS

100%

Completed: 100% Checked

0%

Plagiarism

100%

Unique



Sentence Wise Result



Matched Sources



Document View

UNIQUE

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendap...

UNIQUE

Namun disini saya memakai alat Mindwave untuk pengambilan sinyal gelombang...

UNIQUE

Buatlah fungsi (file terpisah/library dengan nama NPMsave.py) untuk mendapatk...

UNIQUE

Buatlah fungsi (file terpisah/library dengan nama NPMrealtime.py) untuk mendap...

UNIQUE

Buatlah fungsi (file terpisah/library dengan nama NPMcsv.py) untuk membaca fil...

UNIQUE

Tuliskan peringatan error yang didapat dari mengerjakan praktek kelima ini, dan j...

UNIQUE

dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi err...

UNIQUE

Syntax Errors adalah suatu keadaan dimana saat kode python mengalami kesala...