

# **CERDAS MENGUASAI FLASK**

***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

---

<b>1</b>	<b>HTTP GET METHOD REQUEST</b>
----------	--------------------------------

<b>1</b>
----------



# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	

<b>1</b>	<b>HTTP GET METHOD REQUEST</b>	<b>1</b>
1.1	HTTP Method Request	1
1.1.1	Pengertian HTTP Method Request	1
1.1.2	Jenis-jenis HTTP Method Request	2
1.1.3	Penjelasan Lengkap HTTP Get Method	2



1.1.4	Pembacaan HTTP Get Method	3
1.1.5	Pembacaan Data pada URL	3
1.2	Mekanisme HTTP Method Request	3
1.2.1	Mekanisme / Alur kerja HTTP Get Method	3
1.3	Contoh URL HTTP Get Method	4
1.4	Mendapatkan Parameter GET Python Flask	10
1.5	Macam-Macam Penanganan Error Proyek	11
1.5.1	Penanganan Error pada Python dan Flask	11
Daftar Pustaka		13
Index		15

# DAFTAR GAMBAR

---

1.1	Fungsi Sederhana	11
-----	------------------	----



# DAFTAR TABEL

---

1.1	Penjelasan Schema	4
-----	-------------------	---



# Listings

---

1.1	Contoh kode untuk schema	3
1.2	Contoh kode untuk membaca file CSV	4
1.3	Contoh kode untuk menampilkan semua data dari file CSV	5
1.4	Contoh kode untuk menampilkan data dari kolom	5
1.5	Contoh kode untuk menampilkan 5 data teratas dan terbawah	5
1.6	Contoh kode untuk mengurutkan data	5
1.7	Contoh kode untuk mengganti data	5
1.8	Contoh kode untuk menghapus kolom tertentu	5
1.9	Contoh kode untuk menghapus semua data	6
1.10	Contoh kode untuk menambah data	6
1.11	Contoh kode untuk Menampilkan data csv	6
1.12	Contoh kode untuk Menampilkan seluruh jumlah baris	6
1.13	Contoh kode untuk Menampilkan data perbaris	6
1.14	Contoh kode untuk Mengubah data berdasarkan index id	7
1.15	Contoh kode untuk Menghapus data berdasarkan index id	7
1.16	Contoh kode untuk Mengubah data menjadi Json	7
1.17	Contoh kode untuk Menampilkan seluruh data	8
1.18	Contoh kode untuk Menghitung jumlah data	8

1.19	Contoh kode untuk Menghitung jumlah kolom	8
1.20	Contoh kode untuk Menghitung jumlah baris	8
1.21	Contoh kode untuk Mengembalikan nilai data	9
1.22	Contoh kode untuk URL API	9
1.23	Contoh kode untuk membaca sinyal gelombang otak	11

# 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[1].

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



# BAB 1

---

## HTTP GET METHOD REQUEST

---

### 1.1 HTTP Method Request

#### 1.1.1 Pengertian HTTP Method Request

Protokol HTTP adalah protokol permintaan atau respon. Klien mengirimkan permintaan keserver dalam bentuk metode permintaan, URL, dan versi protokol, diikuti oleh pesan seperti MIME yang berisi perubahan perminta, informasi klien, dan kemungkinan onten tubuh melalui koneksi dengan server[2]. Protokol ini sangat ringan serta generik dan tidak berstatus sehingga dapat dipergunakan oleh tipe dokumen apa saja. Method adalah sekumpulan kode yang diberi nama, untuk merujuk kesekumpulan kode yang ada kemudian digunakan sebuah nama yang disebut dengan nama method. Method sendiri mempunyai parameter sebagai input (masukan) dan nilai kembalian sebagai output (keluaran). Request adalah permintaan dimana fungsi ini digunakan sebagai istilah ataupun kinerja dalam pengembalian nilai dari masukan yang dieksekusi.

Berdasarkan beberapa penjelasann diatas, maka untuk pengertian dari HTTP Method Request sendiri merupakan seperangkat metode permintan untuk menunjukkan tindakan yang diinginkan yang akan dilakukan untuk sumber daya tertentu. Meskipun

mereka juga bisa menjadi kata benda, metode permintaan ini kadang-kadang disebut sebagai verba HTTP. Masing-masing menerapkan semantik yang berbeda, namun beberapa fitur umum digunakan bersama oleh mereka adalah misalnya Metode permintaan dapat berupa safe, idempotent, atau cacheable.

### 1.1.2 Jenis-jenis HTTP Method Request

1. GET : akan dijelaskan pada point berikutnya.
2. HEAD : Metode HEAD meminta tanggapan yang identik dengan permintaan GET, namun tanpa respon body.
3. POST : Metode POST digunakan untuk mengirimkan entitas ke sumber daya yang ditentukan, sering menyebabkan perubahan pada keadaan atau efek samping pada server.
4. PUT : Metode PUT menggantikan semua representasi terkini dari sumber target dengan muatan permintaan.
5. DELETE : Metode DELETE akan menghapus sumber daya yang ditentukan
6. CONNECT : Metode CONNECT menetapkan terowongan keserver yang diidentifikasi oleh sumber target.
7. OPTIONS : Metode OPTIONS digunakan untuk menggambarkan opsi komunikasi untuk sumber target.
8. TRACE : metode TRACE ini yaitu untuk melakukan tes pesan loop-back disepanjang jalan menuju sumber daya target.
9. PATCH : Metode PATCH digunakan untuk menerapkan modifikasi sebagian pada sumber daya.

### 1.1.3 Penjelasan Lengkap HTTP Get Method

Metode GET digunakan untuk meminta representasi sumber daya yang ditentukan. permintaan menggunakan Get seharusnya hanya mengambil data. GET adalah salah satu metode HTTP yang paling umum digunakan baik dalam pengimplementasian biasa ataupun sudah dalam bentuk pengujian. Hal yang harus diperhatikan dalam Method Get yaitu :

- Permintaan GET dapat di-cache.
- Permintaan GET tetap ada dalam riwayat browser.
- Permintaan GET dapat ditandai.
- Permintaan GET tidak boleh digunakan saat berurusan dengan data sensitif.

- Permintaan GET memiliki batasan panjang.
- Permintaan GET hanya digunakan untuk meminta data (tidak dimodifikasi).
- Permintaan GET dibatasi oleh panjang string sebanyak 2047 karakter.
- Permintaan GET memungkinkan pengunjung langsung memasukkan nilai variabel pada form proses.

### 1.1.4 Pembacaan HTTP Get Method

Data dikirimkan dalam HTTP Request dalam dua cara, tergantung dari method yang dikirimkan, yaitu :

1. Melalui URL, dengan parameter yang diberikan. Digunakan oleh GET.
2. Melalui entity body dalam HTTP Request. Digunakan untuk POST dan PUT.

Pada prakteknya terdapat satu cara lagi untuk mengirimkan data, yaitu melalui cookie, tetapi penggunaan cookie tidak akan terlalu efektif karena cookie dirancang untuk menyimpan data status pengguna.

### 1.1.5 Pembacaan Data pada URL

Pembacaan data yang dikirimkan melalui URL biasanya dilakukan untuk request dengan method GET. Untuk melihat bagaimana GET mengirimkan data, kita terlebih dahulu harus mengerti tentang sintaks penulisan URL. Secara umum, sebuah URL memiliki sintaks seperti berikut :

```
<scheme>://<user>:<password>@<host>:<port>/<path>?<query>#<fragment>
```

**Listing 1.1** Contoh kode untuk schema

Apa makna dari setiap bagian dari URL yang dijelaskan pada 1.1? Pada tabel 1.1, anda dapat melihat makna dan maksud dari contoh URL yang telah diberikan.

## 1.2 Mekanisme HTTP Method Request

### 1.2.1 Mekanisme / Alur kerja HTTP Get Method

Mekanisme adalah suatu rangkaian kerja sebuah alat yang digunakan dalam menyelesaikan sebuah masalah yang berkaitan dengan proses kerja, tujuannya untuk menghasilkan hasil yang maksimal serta mengurangi datangnya atau munculnya kegagalan. untuk mekanisme HTTP Get Method sendiri dapat diperhatikan sebagai berikut :

1. Silahkan membuat dan membangun sebuah URL API.



**Tabel 1.1** Penjelasan Schema

Nama	Deskripsi	Harus ada ?
schema	Protokol yang digunakan	Ya
user	Nama pengguna	Tidak
password	Password untuk nama pengguna	Tidak
hots	Hostname atau IP	Ya
port	Port yang akan diakses. Beberapa atau sebagian protokol memiliki port standar yaitu seperti HTTP = 80	Tergantung Protokol
paht	Lokasi data pada server	Tergantung Protokol
query	Digunakan untuk mengirimkan parameter kepada aplikasi.	Tidak
fragment	Nama dari bagian tertentu pada data (misalnya : judul pada buku)	Tidak

2. Didalam URL API (endpoint) tersebut kita akan menggunakan fungsi Method Get.
3. Kemudian didalam endpoint tersebut akan difungsikan inputan.
4. Inputan tersebut kemudian akan meghasilkan output (keluaran) dari Metgod GET tersebut.
5. Secara sederhana, garis besar mekanisme atau alur kerja Method Get nampak seperti penjelasan diatas.
6. Untuk tutorial pembangunan endpoint seperrti pada point kedua akan dijelaskan pada point selanjutnya.

### 1.3 Contoh URL HTTP Get Method

Untuk pemberian contoh ini akan dibarengin dengan tutorial pembangunannya, jadi diharapkan teman-teman dapat dengan mudah memahami dan mudah dalam mengikuti contoh yang diberikan. Berikut tutorialnya :

1. Endpoint adalah perangkat komputasi jarak jauh yang berkomunikasi bolak-balik dengan jaringan yang terhubung dengannya. Fungsi-fungsi yang dipergunakan yaitu sebagai contoh berikut :

- Fungsi 1 : LoadData.py yaitu membaca file CSV.

```

1 import pandas
2
3 # membaca file csv
4 def load_data(filename):

```

```
5 return pandas.read_csv(filename, delimiter=';')
```

**Listing 1.2** Contoh kode untuk membaca file CSV

- Fungsi 2 : Getalldata.py yaitu untuk menampilkan semua data dari file CSV.

```
1 def get_all_data(dataframe):
2     return dataframe
```

**Listing 1.3** Contoh kode untuk menampilkan semua data dari file CSV

- Fungsi 3 : Getcolumn.py yaitu untuk menampilkan data dari kolom tertentu.

```
1 def get_column(dataframe, fieldname):
2     return dataframe.loc[:, [fieldname]]
```

**Listing 1.4** Contoh kode untuk menampilkan data dari kolom

- Fungsi 4 : Gettopfive.py yaitu untuk menampilkan 5 data teratas dan terbawah.

```
1 def get_top_five(dataframe, order_by):
2     return dataframe.head() if order_by == 'first'
3     else dataframe.tail()
```

**Listing 1.5** Contoh kode untuk menampilkan 5 data teratas dan terbawah

- Fungsi 5 : Sorting.py yaitu untuk mengurutkan data dari yang terbesar ke yang terkecil dan sebaliknya.

```
1 def sorting(dataframe, order_by):
2     asc = True if order_by == 'ascending' else
3     False
4     return dataframe.sort_index(ascending=asc)
```

**Listing 1.6** Contoh kode untuk mengurutkan data

- Fungsi 6 : Convertjson.py yaitu untuk mengganti data kedalam format Json.

```
1 import json
2 def convert_json(dataframe):
3
4     return
5     json.loads(dataframe.to_json(orient='index'))
```

**Listing 1.7** Contoh kode untuk mengganti data

- Fungsi 7 : Deletecolumn.py yaitu untuk menghapus kolom tertentu.

```
1 def delete_column(dataframe, column):
2     return dataframe.drop(column, axis=1)
```

**Listing 1.8** Contoh kode untuk menghapus kolom tertentu

- Fungsi 8 : Deletealldata.py yaitu untuk Menghapus semua data pada file CSV.

```

1 def delete_all_data(dataframe):
2     dataframe.drop(['raw_value', 'data', 'sign'],
3     axis=1)

```

**Listing 1.9** Contoh kode untuk menghapus semua data

- Fungsi 9 : Insertdata.py yaitu untuk menambahkan data.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:02:27 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 from get_row_count import get_row_count
9
10 def insert_data(dataframe, newdata):
11     index = get_row_count(dataframe) + 1
12     dataframe.at[index] = newdata

```

**Listing 1.10** Contoh kode untuk menambah data

- Fungsi 10 : Getinfo.py yaitu untuk Menampilkan data csv namun dengan format json bawaan dari library pandas yang akan berupa deskripsi.

```

1 def get_info(dataframe):
2     return dataframe.describe()

```

**Listing 1.11** Contoh kode untuk Menampilkan data csv

- Fungsi 11 : Getrow.py yaitu untuk Menampilkan seluruh jumlah baris.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:20:06 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 def get_row(dataframe, id):
9     return dataframe.loc[id, ['raw_value', 'data',
10     'sign']]

```

**Listing 1.12** Contoh kode untuk Menampilkan seluruh jumlah baris

- Fungsi 12 : Getrowfield.py yaitu untuk Menampilkan data perbaris sesuai dengan kolom yang diinginkan.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:22:00 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 def get_row_field(dataframe, field):

```

```
9 return "{}:{}".format(field, dataframe[field])
```

**Listing 1.13** Contoh kode untuk Menampilkan data perbaris

- Fungsi 13 : Updatedata.py yaitu untuk Mengubah data berdasarkan index id.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:20:57 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 def update_data(dataframe, id, newdata):
9     dataframe.loc[id, :] = newdata
10    return dataframe
```

**Listing 1.14** Contoh kode untuk Mengubah data berdasarkan index id

- Fungsi 14 : Deletedata.py yaitu untuk Menghapus data berdasarkan index id.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:20:45 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 def delete_data(dataframe, id):
9     6
10    id = int(id) if type(id) != int else id
11    dataframe.drop([id], axis=0, inplace=True)
12    return dataframe
```

**Listing 1.15** Contoh kode untuk Menghapus data berdasarkan index id

- Fungsi 15 : Getrowjson.py yaitu untuk Mengubah data menjadi Json.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:25:18 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 from convert_json import convert_json
9 def get_row_json(dataframe):
10    return convert_json(dataframe)
```

**Listing 1.16** Contoh kode untuk Mengubah data menjadi Json

- Fungsi 16 : Getalldatajson.py yaitu untuk Menampilkan seluruh data dalam bentuk format Json.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:25:20 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 from convert_json import convert_json
9 def get_all_data_json(dataframe):
10     return convert_json(dataframe)

```

**Listing 1.17** Contoh kode untuk Menampilkan seluruh data

- Fungsi 17 : Amountdata.py yaitu untuk Menghitung jumlah data secara keseluruhan dari kolom dan baris.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:25:21 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 def jumlah_data(dataframe, field):
9     count = dict(row=dataframe.shape[0],
10 column=dataframe.shape[1])
11     if not field == 'all':
12         return count['row'] if field == 'row' else
13 count['column']
14     else:
15         return count

```

**Listing 1.18** Contoh kode untuk Menghitung jumlah data

- Fungsi 18 : Getcolumncount.py yaitu untuk Menghitung jumlah kolom pada file.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:25:21 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 def get_column_count(dataframe):
9     return dataframe.shape[1]

```

**Listing 1.19** Contoh kode untuk Menghitung jumlah kolom

- Fungsi 19 : Getrowcount.py yaitu untuk Menghitung jumlah baris pada file.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:26:51 2019
4
5 @author: Rahmatul Ridha
6 """

```

```

7
8 def get_row_count(dataframe):
9     return dataframe.shape[0]

```

**Listing 1.20** Contoh kode untuk Menghitung jumlah baris

- Fungsi 20 : Reloaddata.py yaitu untuk Mengembalikan nilai data.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:26:49 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 def reload_data(filename, dataframe):
9     dataframe.to_csv(filename, index=False, sep=':')

```

**Listing 1.21** Contoh kode untuk Mengembalikan nilai data

2. Endpoint 1 (URL API / Dataset) Pembatasan pertama ialah dari Endpoint 1 dimana URL API nya yaitu dataset. Silahkan tuliskan code dibawah sebagai contoh pertama :

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Mar 12 14:27:49 2019
4
5 @author: Rahmatul Ridha
6 """
7
8 @app.route('/dataset', methods=['GET', 'POST',
9 'DELETE'])
10 def dataset():
11     args = request.args
12     response = dataframe
13     if request.method == 'GET':
14         if not len(args) is 0:
15             if 'field' in args:
16                 response = get_column(response,
17 args['field'])
18             if 'topfive' in args:
19                 response = get_top_five(response,
20 args['topfive'])
21             if 'sort' in args:
22                 response = sorting(response,
23 args['sort'])
24             if 'format' in args:
25                 if args['format'] == 'json':
26                     response = jsonify(get_all_data_json(response))
27                 else:
28                     response = str(response)
29             else:
30                 response = str(response)
31     return response

```

**Listing 1.22** Contoh kode untuk URL API

Pada tabel code diatas dapat dilihat bahwa fungsi yang digunakan ialah dataset dimana data set dilakukan pemanggilan fungsi yang ada di file-file yang telah di import. Seperti yang bisa dilihat bahwa pada fungsi dataset direalisasikan dengan metode GET, POST dan DELETE jadi disesuaikan dengan fungsi pada file-file yang dihubungkan atau dipanggil kedalam fungsi dataset.

## 1.4 Mendapatkan Parameter GET Python Flask

1. Pengenalan Python Python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat interpreter, interactive, objectoriented, dan dapat beroperasi hampir di semua platform: Mac, Linux, dan Windows. Python termasuk bahasa pemrograman yang mudah dipelajari karena sintaks yang jelas, dapat dikombinasikan dengan penggunaan modulmodul siap pakai, dan struktur data tingkat tinggi yang efisien [3]. Python mendukung multi paradigma pemrograman, utamanya; namu tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis.

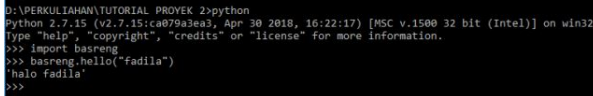
Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

2. Pengenalan Flask Flask adalah *Web Application Framework* yang ditulis dalam bahasa pemrograman Python. Flask digunakan untuk mempersingkat dan mempermudah pengembangan *Web Application*[4]. Flask disebut micro framework karena tidak membutuhkan alat-alat tertentu atau pustaka. Flask tidak memiliki database abstraction layer, validasi form, atau komponen lain dimana sudah ada database pihak ketiga yang menyediakan fungsi umum.
3. Penjelasan Parameter GET Python Flask Parameter GET pada Python Flask ini dilampirkan dan diujikan dalam bentuk file penuh dengan beberapa fungsi. File tersebut bernama Main.py. Untuk penerapan lebih dan contoh GETnya sudah ditampilkan dan dijelaskan sebelumnya pada point contoh URL GET. Namun, penggabungannya bersama Flask Python ada pada file ini. Perhatikan penjelasan dan tutorialnya agar dapat dimengerti. Namun sebelum melanjutkan tutorialnya, pertama-tama anda harus memastikan beberapa hal yaitu sebagai berikut :

## 1.5 Macam-Macam Penanganan Error Proyek

### 1.5.1 Penanganan Error pada Python dan Flask

1. Contoh Kasus 1 : Penerapan fungsi sederhana yang dieksekusi dicommand prompt. Contoh pemanggilan fungsi apabila dieksekusi di CMD, seperti gambar 1.1



```
D:\PERKULIAHAN\TUTORIAL PROYEK 2>python
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import basreng
>>> basreng.hello("fadila")
'halo fadila'
>>>
```

**Gambar 1.1** Fungsi Sederhana

ini adalah contoh untuk pengekseskuan file python yang berupa gunsi yang telah dibuat. Berikut langkah-langkahnya :

- Petama-tama masukkan kedalam directory tempat anda menyimpan file yang telah anda buat.
  - kemudian pada directory tersebut ketik python
  - Setelah masuk kedalam python silahkan masukkan file python basreng
2. Contoh kasus 2 : Kode pembawa sinyal gelombang otak (NeuroSky Mindwave EEG). Kodenya seperti contoh 1.23, silahkan tutorialnya diikuti terlebih dahulu.

```
1 import mindwave, time
2 import keyboard
3 import csv
4
5 headset = mindwave.Headset('COM4', '1425')
6 time.sleep(2)
7 a=0
8 headset.connect()
9 print "Connecting ..."
10
11 while headset.status != 'connected':
12     time.sleep(0.5)
13     if headset.status == 'standby':
14         headset.connect()
15         print "Retrying connect ..."
16 print "Connected."
17
18 while True:
19     print "raw_value: %s : %s" % (headset.raw_value, a)
20     writer.writerow({'RawValue': headset.raw_value, 'sign': a
    })
```

**Listing 1.23** Contoh kode untuk membaca sinyal gelombang otak





# 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.
2. N. R. Wyler, *Aggressive network self-defense*. Elsevier, 2005.
3. A. Kadir, "Dasar pemrograman python," *Yogyakarta: ANDI*, 2005.
4. P. Lokhande, F. Aslam, N. Hawa, J. Munir, and M. Gulamgaus, "Efficient way of web development using python and flask," 2015.



# Index

---

disruptif, xxix  
modern, xxix