

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

1	Judul Bagian Pertama	1
2	Judul Bagian Kedua	3
3	Pengenalan Flask	5
4	HTTP GET METHOD REQUEST	15

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>	
1 Judul Bagian Pertama	1
1.1 Perintah Navigasi	1
2 Judul Bagian Kedua	3
2.1 Perintah Navigasi	3
	ix

3	Pengenalan Flask	5
3.1	Definisi <i>Micro Frameworki</i>	5
3.2	Jenis-Jenis <i>Framework</i> Python serta Kelebihan dan Kekurangan	6
3.2.1	Django	6
3.2.2	Flask	8
3.2.3	Tornado	9
3.2.4	Falcon	10
3.2.5	Hug	11
3.2.6	Sanic	11
3.2.7	Aiohttp	12
3.3	Instalasi dan Hello World di Flask	13
3.3.1	Instalasi Python 2.7.	13
4	HTTP GET METHOD REQUEST	15
4.1	HTTP Method Request	15
4.1.1	Pengertian HTTP Method Request	15
4.1.2	Jenis-jenis HTTP Method Request	16
4.1.3	Penjelasan Lengkap HTTP Get Method	16
4.1.4	Pembacaan HTTP Get Method	17
4.1.5	Pembacaan Data pada URL	17
4.2	Mekanisme HTTP Method Request	17
4.2.1	Mekanisme / Alur kerja HTTP Get Method	17
4.3	Contoh URL HTTP Get Method	18
4.4	Mendapatkan Parameter GET Python Flask	18
4.5	Macam-Macam Penanganan Error Proyek	19
4.5.1	Penanganan Error pada Python dan Flask	19
	Daftar Pustaka	21
	Index	23

DAFTAR GAMBAR

3.1	Download Softfile Python 2.7.	13
3.2	Proses Instalasi Flask	14
4.1	Fungsi Sederhana	19

DAFTAR TABEL

4.1	Penjelasan Schema	18
-----	-------------------	----

Listings

3.1	Contoh kode program hello.py	13
3.2	Contoh kode program main.py	13
4.1	Contoh kode untuk schema	17
4.2	Contoh kode untuk membaca sinyal gelombang otak	20

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

JUDUL BAGIAN PERTAMA

1.1 Perintah Navigasi

Perintah navigasi direktori

BAB 2

JUDUL BAGIAN KEDUA

2.1 Perintah Navigasi

Perintah navigasi direktori

BAB 3

PENGENALAN FLASK

3.1 Definisi *Micro Framework*

Microframework adalah istilah yang digunakan untuk merujuk pada kerangka kerja web minimalis. Kerangka ini sangat berbeda dengan kerangka kerja tumpukan penuh. Juga tidak memiliki sebageian besar fungsionalitas yang umum yang ada dalam kerangka kerja aplikasi web lengkap, seperti:

1. Akun, otentikasi, otorisasi, peran, dll.
2. Abstraksi basis data melalui pemetaan objek-relasional.
3. Validai *input* dan sanitasi *input*.
4. Mesin *template web*.

Biasanya, sebuah *microframework* memfasilitasi menerima permintaan HTTP, merutekan permintaan HTTP ke *controller* yang sesuai, mengirim *controller*, dan mengembalikan respons HTTP. *Microframeworks* seringkali dirancang khusus untuk membangun API untuk layanan atau aplikasi lain. Misalnya, Lumen *microframework* dirancang untuk pengembangan *Microservices* dan pengembangan API. *Mi-*

croframework, sebuah *tool* yang digunakan untuk *project* yang lebih kecil dan penggunaan untuk kasus yang spesifik. Ini sama saja dengan menyederhanakan *framework* agar lebih mudah dalam implementasi dan menyediakan *testing* dan *deployment* yang lebih cepat. *Microframework* mengeluarkan banyak sekali komponen yang ada pada pengaturan *full-stack*, termasuk:

1. *Web template engine*
2. *Input validation*
3. *Database abstraction*
4. *Roles, accounts, and authentication*

Kerugian menggunakan *microframework* adalah saat *project* mulai tumbuh besar dengan cepat. Dimana *microframework* tidak memiliki fitur yang dibutuhkan untuk mengakomodasi pertumbuhan *website*. Dengan kata lain kamu kehilangan fleksibilitas. *Micro-framework* lebih baik saat digunakan untuk *project* kecil yang membutuhkan kesederhanaan, *overhead* yang rendah dan *deployment* yang cepat. *Developer* yang sudah berpengalaman bisa saja menggunakan *microframework* pada awal *project* dan menambahkan tambahan *microframework* jika diperlukan. Hal ini merupakan pilihan yang menarik, tetapi untuk pemula dan *developer* menengah harus menghindari ini [2].

3.2 Jenis-Jenis *Framework* Python serta Kelebihan dan Kekurangan

3.2.1 Django

Django adalah kerangka kerja web Python yang memungkinkan individu dalam pengembangan yang bersih dan cepat. Kerangka kerja web secara umum dikatakan sebagai campuran komponen yang membantu pengembang mengembangkan situs web lebih cepat dan mudah. Karena itu, ini adalah kerangka kerja sumber bebas dan terbuka. Ini dapat disebut sebagai kerangka kerja yang memungkinkan pengembang untuk mengambil konsep penyelesaian secepat mungkin. Django sebagai kerangka kerja membantu mengurangi beberapa kesalahan keamanan umum yang dapat diawasi dengan mudah saat mengembangkan aplikasi. Skalabilitas adalah fitur lain yang disediakan oleh kerangka ini.

Django memiliki *tagline* "*The web framework for perfectionists with deadlines*", bagaimana tidak, karena secara *default* Django sudah memiliki berbagai modul umum yang biasa digunakan ketika mengembangkan aplikasi web.

Kelebihan:

1. Cepat.

Ini telah dirancang sedemikian rupa untuk membantu pengembang membuat aplikasi secepat mungkin. Dari ide, produksi hingga rilis, Django membantu menjadikannya efektif dan efisien. Dengan demikian itu menjadi solusi ideal bagi pengembang yang memiliki fokus utama pada tenggat waktu.

2. Penuh dimuat.

Ini bekerja dengan cara yang mencakup puluhan tambahan untuk membantu dengan otentikasi pengguna, peta situs, administrasi konten, umpan RSS, dan banyak lagi hal-hal seperti itu. Aspek-aspek ini membantu dalam melaksanakan proses pengembangan web sepenuhnya.

3. Aman.

Ketika melakukannya di Django, dipastikan bahwa pengembang tidak melakukan kesalahan yang terkait dengan keamanan. Beberapa kesalahan umum termasuk injeksi SQL, pemalsuan permintaan lintas situs, clickjacking dan skrip lintas situs. Untuk mengelola nama pengguna dan kata sandi secara efektif, sistem otentikasi pengguna adalah kuncinya.

4. Dapat diukur.

Untuk memenuhi permintaan lalu lintas terberat, manfaat kerangka Django dapat dilihat. Oleh karena itu, situs tersibuk menggunakan media ini untuk dengan cepat memenuhi permintaan lalu lintas.

5. Serbaguna.

Manajemen konten, *platform* komputasi ilmiah, dan bahkan organisasi besar, semua aspek ini dikelola secara efisien dengan penggunaan Django.

6. Dokumentasi yang sangat lengkap dan kamu tidak perlu banyak - banyak *googling* karena sudah disediakan contoh.

7. Modul administrasi yang *auto generate* sesuai dengan model yang didefinisikan di dalam aplikasi. Lebih dari sekedar *CRUD generator*.

8. Sistem migrasi *database* otomatis yang tidak perlu kamu tulis *script*-nya. Cukup mengubah class dan struktur *database* pun berubah sesuai perubahan terakhir.

9. Memiliki sistem form yang kokoh.

10. Sudah *built-in* untuk sistem autentikasi dan roles bila Anda menggunakan *relational database* yang didukung Django seperti MySQL dan PostgreSQL.

11. Memiliki ekstensi - ekstensi yang bisa membuat kamu lebih produktif seperti *Django Rest Framework*, *Django Rest Auth*, *Django Celery*, *Django Mongo-engine*, *GeoDjango*, dan lainnya.

12. Memiliki *template engine* sendiri yang lebih *powerful*.

13. Kompatibilitas dengan berbagai modul dan *library* lain.

Kekurangan:

1. Menggunakan pola perutean, tentukan URL-nya

2. Django terlalu monolitik

3. Semuanya didasarkan pada Django ORM
4. Komponen dikerahkan bersama
5. Pengetahuan tentang sistem lengkap diperlukan untuk bekerja.

3.2.2 Flask

Python adalah Flask, yang merupakan kerangka kerja mikro untuk Python berdasarkan teknologi seperti Werkzeug, Jinja 2. Flask pada dasarnya adalah kerangka kerja web Python yang dibangun dengan inti kecil dan selanjutnya mudah untuk memperpanjang ekstensi Flask lebih berorientasi Python daripada Django karena beberapa alasan yang jelas. Karena ada sedikit kode *boilerplate* yang harus ditangani oleh pengembang, Flask adalah kerangka kerja web yang mungkin tidak perlu dikerjakan pengembang lebih lama untuk memahami mereka. Banyak aplikasi terkenal di luar sana ditulis dalam kerangka kerja Flask seperti Pinterest, LinkedIn dan halaman web komunitas untuk Flask itu sendiri [3].

Flask sendiri dapat dikatakan sebagai *web framework* yang fleksibel terhadap library apapun untuk Python. Selain itu dokumentasinya yang jelas membuat Flask sangat diminati oleh kawula muda.

Kelebihan:

1. *Framework* yang mudah untuk digunakan dan dipahami.
2. Berisi pengembangan server dan *debugger*.
3. *RESTfull request dispatching*.
4. Menggunakan Jinja2 *template engine*.
5. Dukungan untuk *secure cookies* pada sisi *client*.
6. 100% *Web Server Gateway Interface* (WSGI).
7. Berbasis *Unicode* yaitu suatu standar yang dirancang untuk mengizinkan *text* dan *symbol* dari semua tulisan untuk menampilkan dan dimanipulasi secara konsisten oleh komputer.
8. Dokumentasi yang ekstensif.
9. Kompatibilitas *Google App Engine*.

Kekurangan:

1. Fungsionalitas

Beberapa fitur Flask yang perlu kamu ketahui antara lain:

1. *Built-in development server* dan *debugger*.
2. Terintegrasi dengan unit *testing*.

3. RESTful.
4. Menggunakan *template engine* Jinja2.
5. Mendukung *secure cookie*.
6. 100% mendukung WSGI 1.0.
7. *Unicode based*.
8. Dokumentasi yang baik.
9. Komunitas yang kuat.

3.2.3 Tornado

Tornado adalah salah satu kerangka kerja web terbaik dari bahasa pemrograman Python. Kerangka kerja ini memungkinkan pendekatan yang lebih bersih untuk pemrograman server Web dan memiliki fokus yang tajam pada operasi non-pemblokiran, dapat meningkatkan skala ke sejumlah besar koneksi terbuka [4].

Kelebihan:

1. Dukungan bawaan

Tornado hadir dengan dukungan bawaan dan menemukan solusi untuk sebagian besar aspek pengembangan Web yang membosankan seperti templat, pelokalan, cookie yang ditandatangani, dll. Tornado juga memungkinkan pengguna untuk mencampurnya dengan kerangka kerja lain, dengan cuplikan yang sesuai, sesuai untuk kebutuhan mereka.

2. Koneksi serentak

Tornado menawarkan layanan waktu nyata dan mendukung sejumlah besar koneksi konkuren, streaming HTTP (protokol komunikasi yang diterapkan oleh Apple Inc) dan polling panjang (ini adalah teknologi 'sembur', yang memungkinkan mekanisme pendorong emulatif dalam keadaan di yang dorongan nyata tidak mungkin). Dengan Tornado, sangat mudah untuk menulis layanan waktu nyata. FriendFeed memelihara koneksi terbuka, terutama untuk penggunaanya yang sering terlibat.

3. Kinerja tinggi.

Ini adalah fitur paling menarik dari Tornado. Ini sangat cepat dibandingkan dengan semua kerangka kerja Python Web lainnya. Mempertimbangkan keluaran dasarnya, kecepatannya sekitar empat kali lebih tinggi dan juga cukup efisien.

Tornado memiliki beberapa modul utama seperti:

1. *Web framework*.
2. *HTTP Server and Client*.
3. *Asynchronous Networking, Coroutines and Concurrency*.
4. *Utilities*.

3.2.4 Falcon

Falcon merupakan pustaka WSGI yang membantu membangun API web dengan kecepatan lebih cepat. Saat Anda membuat kerangka HTTP API selain Falcon dapat memuat banyak dependensi dan abstraksi yang tidak diperlukan. Falcon, di sisi lain, mengurangi semua ketergantungan dan menyediakan pengembang untuk mengembangkan desain yang lebih bersih yang memungkinkan gaya arsitektur HTTP dan REST [5].

Falcon mengklaim bahwa ia dapat menangani lebih banyak permintaan dengan perangkat keras yang sama jika sedang ditangani oleh kerangka kerja lain. Kerangka kerja ini bertujuan untuk memiliki cakupan kode 100%, sehingga membuatnya lebih andal. Sebagian besar fitur di atas dimungkinkan karena Falcon hanya mempertahankan 2 dependensi pihak ketiga seperti `enamlib`, `mimemagic`. Sesuai dengan halaman `Falith` Github perusahaan seperti RackSpace, OpenStack dan LinkedIn menggunakan Falcon.

Dengan tagline *The Minimalist Python WSGI Framework*, Falcon siap menyuguhkan berbagai fitur yang dapat mempermudah kamu membangun sebuah RESTful API. Falcon merupakan *high performance web framework* yang dapat digunakan untuk membangun HTTP API dan backend apps.

Kelebihan:

1. Cepat

Salah satu persyaratan paling penting dari cloud API adalah mereka harus menangani permintaan secepat mungkin. Ini menjadi fitur penting dalam skenario real-time ketika jumlah permintaan bersamaan tinggi. Falcon adalah salah satu kerangka kerja tercepat yang tersedia.

2. Cahaya

Kerangka kerja dengan jejak ketergantungan yang lama menjadi sangat sulit untuk disatukan dalam berbagai lingkungan karena pembatasan yang diberlakukan oleh ketergantungan tersebut. Falcon hanya memiliki dua dependensi: `enamlib` (pustaka kompatibilitas Python 2 dan 3, yang memfasilitasi basis kode untuk bekerja pada Python 2 dan 3 tanpa memerlukan perubahan apa pun) dan `mimemagic` (yang menyediakan fungsi seperti parsing nama tipe-mime). Ini membuat Falcon lebih mudah untuk diuji dan digunakan.

3. Fleksibel

Falcon tidak membatasi pengembang ketika memilih perpustakaan sehubungan dengan database, otorisasi, dll. Pengembang dapat memilih perpustakaan yang mereka sukai, yang cocok dengan persyaratan skenario proyek saat ini.

Kekurangan:

1. Tidak cocok untuk melayani halaman HTML.

2. Belum tentu lebih cepat daripada Flask.

3.2.5 Hug

Hug adalah kerangka kerja berbasis web Python yang lain memberi para pengembang fleksibilitas mengembangkan API dan memungkinkan dapat mengkonsumsinya sesuka mereka. Pengembangan API telah disederhanakan secara drastis melalui beberapa antarmuka. Biarlah itu pengembangan lokal atau melalui HTTP atau bahkan melalui antarmuka baris perintah (CLI), sejauh ini merupakan cara modern tercepat untuk mengembangkan API. Kerangka kerja Hug telah dibangun dengan fokus tunggal pada kinerja dalam pikiran. Dikatakan mengkonsumsi sumber daya hanya bila diperlukan dan selanjutnya dikompilasi menggunakan Cython untuk mencapai angka-angka luar biasa pada kinerja. Dengan semua alasan yang jelas ini, Hug mencuri mahkota sebagai kerangka kerja web tercepat untuk Python 3.

Kelebihan:

1. Jadikan mengembangkan API yang digerakkan oleh Python sesingkat definisi tertulis.
2. Kerangka kerja harus mendorong kode yang mendokumentasikan sendiri.
3. Itu harus cepat. Pengembang seharusnya tidak pernah merasa perlu mencari di tempat lain karena alasan kinerja.
4. Tes penulisan untuk API yang ditulis di atas pelukan harus mudah dan intuitif.
5. Sihir yang dilakukan sekali, dalam kerangka kerja API, lebih baik daripada mendorong masalah yang disetel ke pengguna kerangka API.
6. Menjadi dasar untuk API Python generasi berikutnya, merangkul teknologi terbaru.

Kekurangan:

1. Hanya dapat memuat kode sedikit.

3.2.6 Sanic

Sanic adalah kerangka kerja web Python *cocok untuk Python 3.5* yang dibangun di atas *uvloop* dan dirancang untuk respons HTTP yang lebih cepat melalui penanganan permintaan yang tidak sinkron. Karena struktur internal dan ketergantungannya yang kuat pada *uvloop*, itu tidak dapat dikembangkan atau digunakan pada lingkungan Windows. Sampai saat ini, Sanic masih dalam tahap pengembangan dan dianggap sebagai bayi di antara kerangka web lain yang tersedia untuk Python. Dengan ini, ada sejumlah kode yang telah ditulis di sekitar Sanic agar Anda dapat menggunakannya untuk keperluan bisnis yang kompleks. Mengingat masih dalam pengembangan, tidak ada banyak aplikasi atau ekstensi untuk Sanic dibandingkan dengan Flask atau Django. Mengingat semua itu, kerangka kerja ini memungkinkan Anda untuk mengambil keuntungan dari sintaks *async* atau menunggu untuk mendefinisikan

fungsi asinkron Anda sendiri. Ini memberikan kekuatan menulis aplikasi asinkron seperti apa yang dapat dicapai dengan menggunakan Node.js.

Kelebihan:

1. Server yang dikonfigurasi untuk harus dilampirkan ke aplikasi yang ada.
2. Aplikasi Sanic dapat menentukan rute reguler yang akan hidup berdampingan dengan server Engine.IO. Pola khas adalah menambahkan rute yang melayani aplikasi klien dan file statis terkait dengan aplikasi ini.

Kekurangan:

- Tidak memiliki banyak aplikasi atau ekstensi.

3.2.7 Aiohttp

Kerangka kerja asinkron yang sangat bergantung pada dan menggunakan fitur Python 3.5+ seperti `async` dan `menunggu`. Kerangka kerja ini tidak hanya kerangka kerja server web tetapi juga bertindak sebagai kerangka kerja klien juga, karena mendukung baik WebSocket Server dan Klien. Ini adalah kerangka kerja terkenal yang memanfaatkan perpustakaan asinkron populer - `asyncio` yang ada di sana sejak awal perpustakaan. `aiohttp` seperti Flask menyediakan objek permintaan dan router untuk memungkinkan pengalihan permintaan ke fungsi yang dikembangkan untuk menanganinya. Sebagai pengembang layanan-mikro, Anda bisa fokus membangun pandangan seperti yang akan Anda lakukan dengan Flask.

Kelebihan:

1. Efisiensi.

Menangani jumlah permintaan yang setara dengan server yang lebih sedikit atau lebih kecil dibandingkan dengan sinkronisasi. Skalabilitas dibatasi oleh jumlah koneksi soket terbuka dalam proses tunggal vs. jumlah utas/proses bersamaan untuk sinkronisasi kerangka kerja web (ribuan hingga puluhan ribu untuk `async` vs. puluhan hingga ratusan untuk sinkronisasi). Server kecil (dalam hal CPU dan memori) menjalankan web `async` layanan dalam satu proses akan cocok dan seringkali mengungguli yang lebih besar server menjalankan layanan web sinkronisasi menggunakan puluhan hingga ratusan utas/proses.

2. Mampu menangani sejumlah besar permintaan bersamaan.

Mengizinkan fungsionalitas *push* yang efisien melalui soket web, `EventSource`, atau koneksi berumur panjang lainnya

Kekurangan:

- Tunggal.

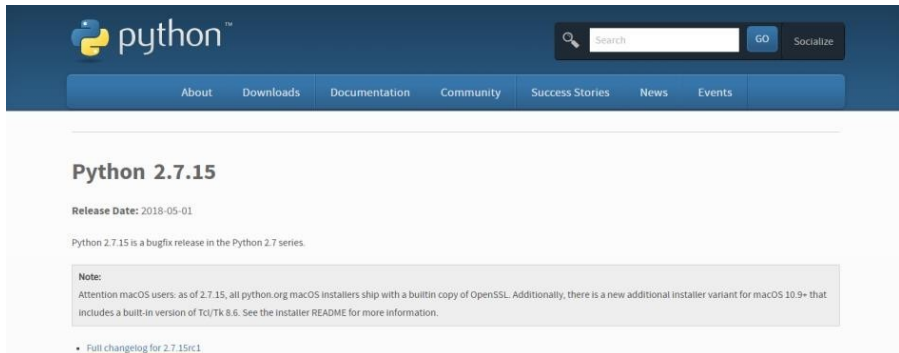
Memiliki model yang lebih kompleks untuk dipikirkan status bersama dan bagaimana hal itu dapat berubah dari satu proses sinkronisasi, harus diingat bahwa status bersama dapat berubah di antara saat-saat menghasilkan kontrol ke loop acara dan mengembalikan kontrol ke kode Anda.

3.3 Instalasi dan Hello World di Flask

3.3.1 Instalasi Python 2.7.

Mulai dengan tutorial dalam menginstall Python 2.7. Python ini digunakan untuk *code* pembaca data dari sinyal gelombang otak yang telah dihasilkan oleh alat EEG yaitu NeuroSky Mindwave. Baiklah langsung kita mulai saja:

1. Pertama-tama silahkan download software dari python versi 2 di laptop anda. Download python versi 2.7.15 dari situs web resminya yaitu <https://www.python.org/>. Silahkan sesuaikan dengan kapasitas laptop anda, bisa yang win 32 atau yang win 64 (32 bit / 64 bit). Contoh downloadnya seperti pada gambar 3.1.



Gambar 3.1 Download Softfile Python 2.7.

2. Setelah di *download* silahkan buka Command Prompt di komputer/PC anda.
3. Kemudian silahkan ketikkan perintah `pip install flask`.
4. Setelah mengetikkan perintah tersebut, silahkan tekan *enter* maka prosesnya akan berjalan. Silahkan tunggu hasilnya. Hasilnya akan nampak seperti gambar 3.2.

Perintah python flask. Contoh *source code* seperti pada listing 3.1.

```
1 def hello():
2     return "Hello World!"
```

Listing 3.1 Contoh kode program hello.py

Contoh *source code* main.py seperti pada listing 3.2.

```
1 from flask import Flask
2 from hello import hello
3
4 app = Flask(__name__)
5
6 @app.route('/oke', methods=['GET'])
```

```
C:\Users\admin\AppData\Local\Programs\Python\Python36\Scripts>pip install flask
Collecting flask
  Using cached Flask-0.12.2-py2.py3-none-any.whl
Collecting itsdangerous>=0.21 (from flask)
  Using cached itsdangerous-0.24.tar.gz
Collecting Werkzeug>=0.7 (from flask)
  Using cached Werkzeug-0.12.2-py2.py3-none-any.whl
Collecting Jinja2>=2.4 (from flask)
  Using cached Jinja2-2.9.6-py2.py3-none-any.whl
Collecting click>=2.0 (from flask)
  Using cached click-6.7-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->flask)
  Using cached MarkupSafe-1.0.tar.gz
Installing collected packages: itsdangerous, Werkzeug, MarkupSafe, Jinja2, click, flask
  Running setup.py install for itsdangerous ... done
  Running setup.py install for MarkupSafe ... done
Successfully installed Jinja2-2.9.6 MarkupSafe-1.0 Werkzeug-0.12.2 click-6.7 flask-0.12.2 itsdangerous-0.24
```

Gambar 3.2 Proses Instalasi Flask

```
7 def annisa():
8     args = request.args
9     if request.method == 'GET':
10         if not len(args) is 0:
11             if 'word' in args:
12                 response = hello(response, args['word'])
13             return response
14
15 if __name__ == "__main__":
16     app.run(debug=True)
```

Listing 3.2 Contoh kode program main.py

BAB 4

HTTP GET METHOD REQUEST

4.1 HTTP Method Request

4.1.1 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[5]. 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.

4.1.2 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.

4.1.3 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.

4.1.4 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.

4.1.5 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 :

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

Listing 4.1 Contoh kode untuk schema

Apa makna dari setiap bagian dari URL yang dijelaskan pada 4.1? Pada tabel 4.1, anda dapat melihat makna dan maksud dari contoh URL yang telah diberikan.

4.2 Mekanisme HTTP Method Request

4.2.1 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 4.1 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 menghasilkan 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.

4.3 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.
2. Fungsi-fungsi yang dipergunakan yaitu sebagai contoh berikut :

4.4 Mendapatkan Parameter GET Python Flask

1. Pengenalan Python Python adalah salah satu bahasa pemograman tingkat tinggi yang bersifat interpreter, interactive, objectoriented, dan dapat beroperasi ham-

pir 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 [6]. 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*[7]. 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 dilampurkan 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 :

4.5 Macam-Macam Penanganan Error Proyek

4.5.1 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 4.1

```
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 4.1 Fungsi Sederhana

ini adalah contoh untuk pengeksekusian 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 4.2, 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 4.2 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. M. I. Fadhil and G. M. Plexindo, ".net microframework untuk pemula."
3. A. L. Ramdani and H. B. Firmansyah, "Clustering application for ukt determination using pillar k-means clustering algorithm and flask web framework," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 1, no. 2, pp. 53–59, 2018.
4. P. M. Panjaitan, "Sistem monitoring cuaca dan identifikasi keadaan cuaca menggunakan teknik web scraping," 2018.
5. N. R. Wyler, *Aggressive network self-defense*. Elsevier, 2005.
6. A. Kadir, "Dasar pemrograman python," *Yogyakarta: ANDI*, 2005.
7. 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