

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	47

DAFTAR ISI

Daftar Gambar	xiii
Daftar Tabel	xv
Foreword	xix
Kata Pengantar	xxi
Acknowledgments	xxiii
Acronyms	xxv
Glossary	xxvii
List of Symbols	xxix
Introduction	xxx
<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 Framework</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
3.2.8	Piramid	13
3.2.9	Web2py	14
3.2.10	TurboGears2	14
3.2.11	Cherrypy	15
3.3	Instalasi dan Hello World di Flask	16
3.3.1	Instalasi Python 2.7.	16
3.3.2	Instalasi Python 3.6	17
3.3.3	Instalasi Framework Flask	19
3.3.4	Instalasi Library Pandas	20
3.3.5	Contoh Penerapan Fungsi Pada Flask Python	21
3.4	Penanganan Error	24
3.4.1	Penanganan Error pada Python	24
3.4.2	Penanganan Error dalam Flask	36
4	HTTP GET METHOD REQUEST	47
4.1	HTTP Method Request	47
4.1.1	Pengertian HTTP Method Request	47
4.1.2	Jenis-jenis HTTP Method Request	48
4.1.3	Penjelasan Lengkap HTTP Get Method	48
4.1.4	Pembacaan HTTP Get Method	49
4.1.5	Pembacaan Data pada URL	49
4.2	Mekanisme HTTP Method Request	49
4.2.1	Mekanisme / Alur kerja HTTP Get Method	49
4.3	Contoh URL HTTP Get Method	50
4.4	Mendapatkan Parameter GET Python Flask	56
4.5	Macam-Macam Penanganan Error Proyek	57
4.5.1	Penanganan Error pada Python dan Flask	57

DAFTAR GAMBAR

3.1	Download Softfile Python 2.7.	16
3.2	Pengecekan Python 2.7.	17
3.3	Pengubahan Environment Python 2.7.	17
3.4	Download Softfile Python 3.6.	18
3.5	Pengecekan Python 3.6.	18
3.6	Pengubahan Environment Python 3.6.	19
3.7	Download Flask	20
3.8	Proses Instalasi Flask	20
3.9	Download Pandas	21
3.10	Proses Instalasi Pandas	22
3.11	Instalasi Pip Flask	22
3.12	Pemanggilan Fungsi Flask Opsi 1	23
3.13	Pemanggilan Fungsi Flask Opsi 2	24

3.14	Output Hello World	24
3.15	Eksekusi Fungsi Percobaan Pertama	25
3.16	Eksekusi Fungsi Percobaan Kedua	25
3.17	Eksekusi Fungsi Percobaan Ketiga	26
3.18	Eksekusi Fungsi Percobaan Keempat	27
3.19	Eksekusi Fungsi Percobaan Kelima	28
3.20	Eksekusi Fungsi Percobaan Keenam	29
3.21	Eksekusi Fungsi Percobaan Ketujuh	29
3.22	Eksekusi Fungsi Percobaan Kedelapan	30
3.23	Eksekusi Fungsi Percobaan Kesembilan	31
3.24	Eksekusi Fungsi Percobaan Kesepuluh	31
3.25	Eksekusi Fungsi Percobaan Kesebelas	31
3.26	Deskripsi Pengukuran	33
3.27	Error pada program coba.py	34
3.28	Keterangan error pada program coba.py	34
3.29	Eksekusi Program import.py	37
3.30	Eksekusi Program import.py	38
3.31	Eksekusi Program import.py	38
3.32	Hasil Eksekusi Program import.py	39
3.33	Eksekusi Program main.py	40
3.34	Hasil Eksekusi Program main.py	41
3.35	Hasil Eksekusi Program main.py	42
3.36	Hasil Eksekusi Program main.py	42
3.37	Eksekusi Program main.py	44
3.38	Hasil Eksekusi Program main.py	45
4.1	Fungsi Sederhana	57

DAFTAR TABEL

4.1	Penjelasan Schema	50
-----	-------------------	----

Listings

3.1	Contoh kode program hello.py	23
3.2	Contoh kode program main.py	23
3.3	File satu.py	24
3.4	File baru satu.py	25
3.5	File baru satu.py	26
3.6	File baru satu.py	27
3.7	File baru satu.py	27
3.8	File baru dua.py	28
3.9	File baru dua.py	29
3.10	File baru dua.py	30
3.11	File coba.py	32
3.12	File coba.py	32
3.13	File coba.py	33
3.14	File try_except.py	34
3.15	File coba_lagi.py	35
3.16	File coba_lagi.py	36
3.17	File import.py	37
3.18	File import.py	38

3.19	File <code>get_top_five.py</code>	39
3.20	File <code>main.py</code>	39
3.21	File <code>get_top_five.py</code>	40
3.22	File <code>delete_column.py</code>	43
3.23	File <code>main.py</code>	43
3.24	File <code>delete_all_data.py</code>	44
4.1	Contoh kode untuk schema	49
4.2	Contoh kode untuk membaca file CSV	50
4.3	Contoh kode untuk menampilkan semua data dari file CSV	51
4.4	Contoh kode untuk menampilkan data dari kolom	51
4.5	Contoh kode untuk menampilkan 5 data teratas dan terbawah	51
4.6	Contoh kode untuk mengurutkan data	51
4.7	Contoh kode untuk mengganti data	51
4.8	Contoh kode untuk menghapus kolom tertentu	51
4.9	Contoh kode untuk menghapus semua data	52
4.10	Contoh kode untuk menambah data	52
4.11	Contoh kode untuk Menampilkan data csv	52
4.12	Contoh kode untuk Menampilkan seluruh jumlah baris	52
4.13	Contoh kode untuk Menampilkan data perbaris	52
4.14	Contoh kode untuk Mengubah data berdasarkan index id	53
4.15	Contoh kode untuk Menghapus data berdasarkan index id	53
4.16	Contoh kode untuk Mengubah data menjadi Json	53
4.17	Contoh kode untuk Menampilkan seluruh data	54
4.18	Contoh kode untuk Menghitung jumlah data	54
4.19	Contoh kode untuk Menghitung jumlah kolom	54
4.20	Contoh kode untuk Menghitung jumlah baris	54
4.21	Contoh kode untuk Mengembalikan nilai data	55
4.22	Contoh kode untuk URL API	55
4.23	Contoh kode untuk membaca sinyal gelombang otak	58
4.24	Contoh1 kasus 2	58

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 sebagian besar fungsionalitas secara 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. Validasi *input* dan sanitasi *input*.
4. Mesin *template web*.

Biasanya, sebuah *microframework* memfasilitasi untuk 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* yang dirancang untuk pengembangan *Microservices* dan pengembangan API.

Microframework merupakan 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 diantaranya:

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 template, 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.

3.2.8 Piramid

Kerangka kerja yang telah dikembangkan atau dibangun untuk aplikasi yang lebih besar. Piramida, nama itu sendiri menunjukkan bahwa itu fleksibel, tidak seperti Django yang menawarkan pendekatan "semuanya di dalam kotak". Aplikasi web dibangun menggunakan Pyramid, mulai dari modul file tunggal dan kemudian proyek-proyek ini berkembang menjadi proyek yang lebih besar dan ambisius dalam waktu singkat. Kerugian dari kerangka kerja web ini adalah dokumentasi mereka sendiri, yang tidak terlalu jelas dan terkadang membingungkan. Chameleon Pyramid dipasang untuk menggunakan templat Chameleon alih-alih templat Jinja. Dibutuhkan beberapa waktu dalam mengembangkan aplikasi file tunggal dengan Pyramid, tetapi kemudian, ini dapat ditingkatkan lebih cepat karena pengaturan awal lebih keras dan rawan kesalahan.

Kelebihan:

1. Fleksibilitas.

Sistem rendering template, cara menyambungkan ke database, cara memetakan url ke tampilan, semacam sistem otentikasi, dan banyak hal lainnya. Piramida sangat bagus karena semua komponen ini dapat ditukar. Anda dapat memilih mesin rendering template, memiliki dua cara berbeda untuk memetakan url ke tampilan dan dapat menggunakannya keduanya di aplikasi yang sama, dapat menggunakan metode apa pun yang ingin disambungkan ke database (meskipun SQLAlchemy umumnya digunakan), dan bahkan dapat terhubung ke beberapa database dari tipe yang sangat berbeda.

2. Kemudahan AJAX.

Penggunaan dekorator dan tampilan XHR membuatnya sangat mudah untuk mendapatkan permintaan AJAX untuk pergi ke tempat yang Anda inginkan. Menjelaskan XHR, dekorator, atau AJAX jauh di luar cakupan tutorial ini, tetapi saya jamin penjelasannya ada di luar sana.

3. Dukungan SQLAlchemy

SQLAlchemy adalah hal yang sangat kuat, banyak orang yang sangat pintar berpikir itu adalah ORM terbaik di sekitar. Jika Anda memilih Django Anda memilih untuk tidak menggunakan SQLAlchemy. Itu hanya masalah besar jika aplikasi Anda sangat (SQL) database intensif - jika Anda ingin melakukan pertanyaan rumit dengan cara yang waras. Di sisi lain, jika aplikasi Anda sederhana maka SQLAlchemy tidak akan banyak membantu, tetapi tidak ada salahnya untuk memilikinya.

Kekurangan:

1. Fungsionalitas

Beberapa fitur Pyramid:

1. Kompatibel dengan berbagai template engine seperti Jinja2, Chameleon, dan Mako.

2. Mempunyai sistem form yang handal.
3. Menggunakan SQL Alchemy untuk teknologi database.
4. Memiliki bootstraper.
5. Function decorator.
6. Asset management.
7. Event dan subscriber

3.2.9 Web2py

Merupakan salah satu *full-stack enterprise framework* yang *free* dan *open source* untuk membangun aplikasi web berbasis *database* yang aman. Web2Py merupakan salah satu *web framework* yang masih ada hingga hari ini. Tidak hanya soal *url routing*, Web2Py pun memiliki *template engine* yang cukup *powerful* untuk membuat halaman web.

Beberapa fitur yang dimiliki oleh Web2py antara lain:

1. Dibuat oleh komunitas terpercaya.
2. Selalu backward compatible.
3. Mudah digunakan.
4. Dapat berjalan di banyak sistem operasi.
5. Dapat berjalan di banyak web server.
6. Dapat "berbicara" ke SQLite3, PostgreSQL, MySQL, MSSQL, FireBird, Oracle, IBM DB2, Informix, Ingres, dan Google App Engine.
7. Aman dari cross site scripting, injection flaws, dan eksekusi file berbahaya.
8. Mengajarkan penggunanya apa arti MVC yang sesungguhnya.
9. Kompatibel dengan berbagai protokol seperti REST, RSS, HTML, REST, XML-RPC, dan lainnya.
10. Dukungan berbagai modul dan library yang sudah disediakan oleh Web2py

3.2.10 TurboGears2

TurboGears2 adalah kerangka web berbasis Python yang didasarkan pada paradigma ObjectDispatch. Ini secara khusus dimaksudkan untuk memungkinkan untuk menulis aplikasi kecil dan ringkas dalam mode Minimal dan aplikasi yang jauh lebih kompleks dalam Mode Stack Penuh. Fitur TurboGears2 adalah ORM dengan dukungan

multi-database nyata, dan juga mendukung partisi data horizontal, sistem widget untuk menyederhanakan pengembangan aplikasi AJAX.

Kelebihan:

1. Dukungan Web Server Gateway Interface (WSGI)
2. Sistem widget yang mempermudah pembuatan aplikasi AJAX
3. Mendukung multi data-exchange format
4. Dapat membuat pluggable application
5. Template engine yang sangat designer friendly

Kekurangan:

1. Fungsionalitas

3.2.11 Cherrypy

CherryPy yang merupakan kerangka kerja berikutnya dalam daftar, yang digunakan untuk menjadi jalan antara masalah dan programmer. Aplikasi web yang dibangun menggunakan kerangka CherryPy terlihat seperti aplikasi Python lainnya dan berjalan tanpa memberikan pengaturan rumit dan penyesuaian terbaik. Bersamaan dengan itu, ia juga memperluas dukungannya ke berbagai jenis server web seperti Apache, IIS dan banyak lagi. Karena itu, CherryPy mengemasnya bersama server web, sehingga aplikasi dapat digunakan di mana pun Python diinstal. Ini juga memungkinkan Anda untuk memulai beberapa server HTTP sekaligus. Tidak ada paksaan yang diberlakukan oleh CherryPy untuk menggunakan mesin templat tertentu, ORM atau pustaka JavaScript dan karenanya kami sebagai pengembang memiliki pilihan untuk memilih mana yang sesuai dengan kebutuhan kami dengan lebih baik.

Kelebihan:

1. CherryPy berjalan dengan mudah di lingkungan yang kompatibel dengan WSGI seperti server web Apache dan bahkan dapat dijalankan di server mandiri tanpa gangguan langkah otorisasi dan akses backend.
2. Pengembangan aplikasi web adalah tersedianya pilihan penyesuaian, dengan demikian menarik lebih banyak peminat untuk itu. Dan bit penyesuaian hanya dimungkinkan karena CherryPy menawarkan berbagai macam fitur dan alat untuk memungkinkannya bagi pengembang.

Kekurangan:

1. Tidak memiliki sistem templating sendiri sehingga harus memilih template yang cocok.

CherryPy memiliki dukungan seperti berikut:

1. URL routing.

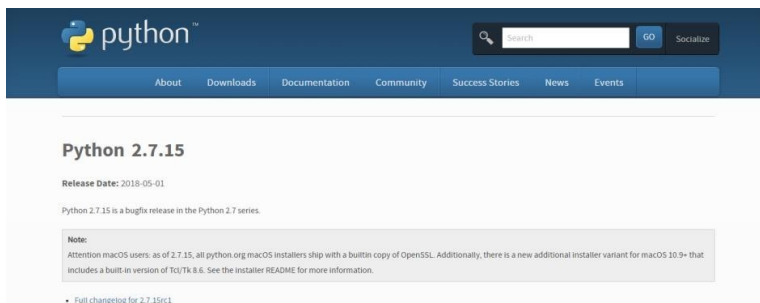
2. Static file management.
3. Managable configuration.
4. Kompatibel dengan WSGI dan HTTP/1.1.
5. Built-in profiling, coverage, dan dukungan testing.
6. Dukungan terhadap session, authentication, static content, dan banyak lagi.
7. Dukungan bawaan untuk caching dan encoding.
8. Sistem konfigurasi yang enak.

3.3 Instalasi dan Hello World di Flask

3.3.1 Instalasi Python 2.7.

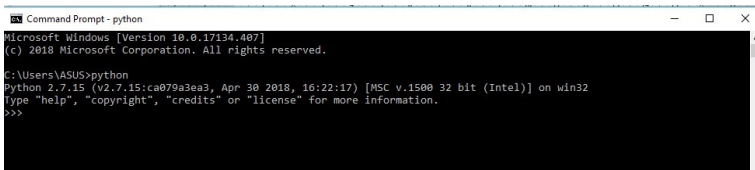
Mulai dengan tutorial dalam menginstall Python 2.7. Python ini digunakan untuk code pembaca data dari sinyal gelombang otak yang telah dihasilkan oleh alat EEG yaitu NeuroSky Mindwave. Baiklah langsung kita mulai saja:

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



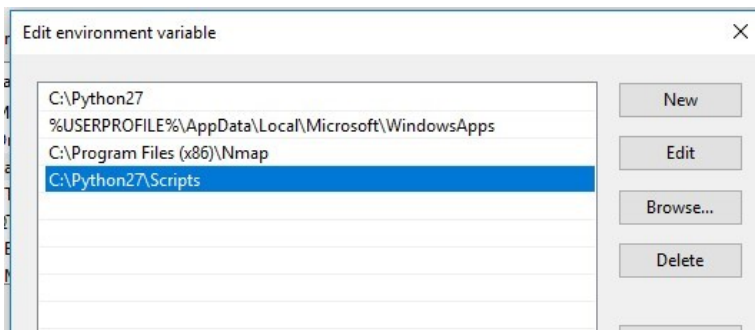
Gambar 3.1 Download Softfile Python 2.7.

2. Setelah berhasil mendownload mentahan pythonnya silahkan lakukan instalasi seperti biasa anda lakukan. Setelah selesai instalasi pythonnya silahkan cek di *Command Prompt*, apakah python telah terbaca/*running* disesuaikan dengan laptop anda atau belum. Contoh pengecekan di Command Prompt seperti pada gambar 3.2
3. Apabila tampilannya telah sesuai dengan gambar 3.2, maka python anda siap digunakan.



Gambar 3.2 Pengecekan Python 2.7.

4. Pastikan python yang terbaca versi 2.7.15 atau bahkan belum ada sama sekali silahkan lakukan konfigurasi ini:
 - Silahkan buka *Control Panel* anda.
 - Pilih System and Security.
 - Kemudian pilih lagi system.
 - Lalu di bagian kiri tampilan ada sub menu Advanced system setting.
 - Pada sub menu tersebut silahkan pilih button Environment Variabel.
 - Silahkan ganti path dengan C:\Python27 dan C:\Python27\Scripts (Lokasi anda menyimpan mentahan python yang telah anda install tadi).
 - Maka tampilannya akan seperti pada gambar 3.3.



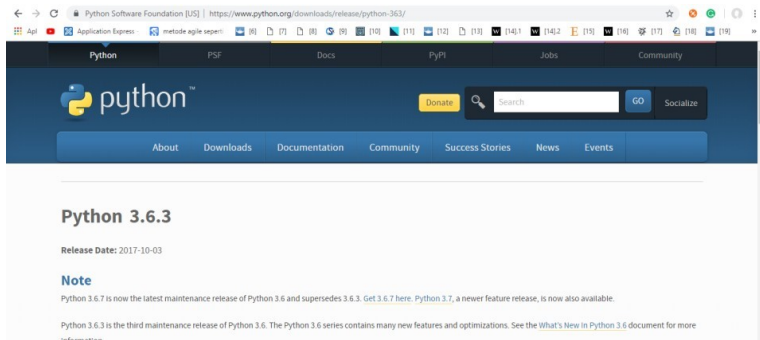
Gambar 3.3 Pengubahan Environment Python 2.7.

- Jangan lupa untuk memasukkan script dari pythonnya sehingga benar-benar bisa terbaca untuk pipnya. Silahkan klik button ok sampai selesai.
- Setelah itu, lakukan pengecekan ulang di Command Prompt maka hasilnya akan berubah menjadi versi 2.7.15.

3.3.2 Instalasi Python 3.6

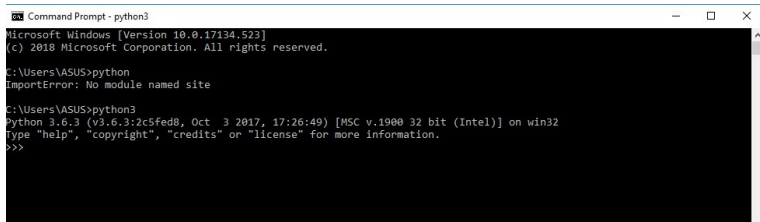
Selanjutnya kita mulai dengan tutorial dalam menginstall Python 3.6. Python ini digunakan untuk code pengolahan data csv ke dalam flask python.

1. Pertama-tama silahkan download software dari python versi 3 di laptop anda. Download python versi 3.6.3 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 seperti pada gambar 3.4.



Gambar 3.4 Download Softfile Python 3.6.

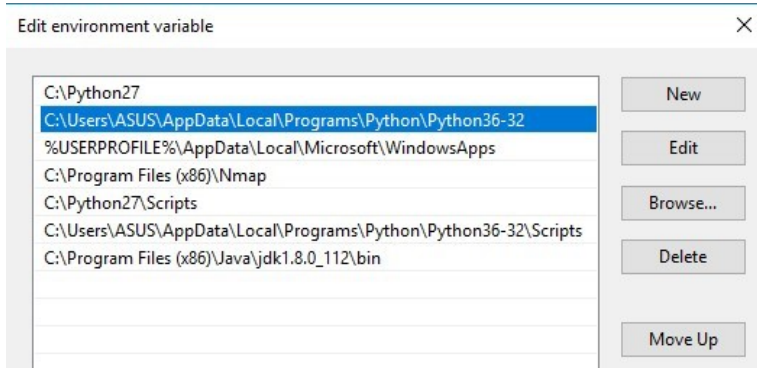
2. Setelah berhasil mendownload mentahan pythonnya silahkan lakukan instalasi seperti biasa anda lakukan. Setelah selesai instalasi pythonnya silahkan check di Command Prompt, apakah Pythonnya telah terbaca / running disesuaikan dengan laptop anda atau belum. Contoh pengecekan di Command Prompt seperti pada gambar 3.5



Gambar 3.5 Pengecekan Python 3.6.

3. Apabila tampilannya telah sesuai dengan contoh diatas, maka python anda siap digunakan.
4. Pastikan python yang terbaca versi 3.6.3 atau bahkan belum ada sama sekali silahkan lakukan konfigurasi ini:
 - Silahkan buka Control Panel anda.
 - Pilih System and Security.
 - Kemudian pilih lagi system.

- Lalu di bagian kiri tampilan ada sub menu Advanced system setting.
- Pada sub menu tersebut silahkan pilih button Environment Variabel.
- Silahkan ganti path dengan `C:\Python36` dan `C:\Python36\Scripts` (Lokasi anda menyimpan mentahan python yang telah anda install tadi).
- Maka tampilannya akan seperti pada gambar 3.6.



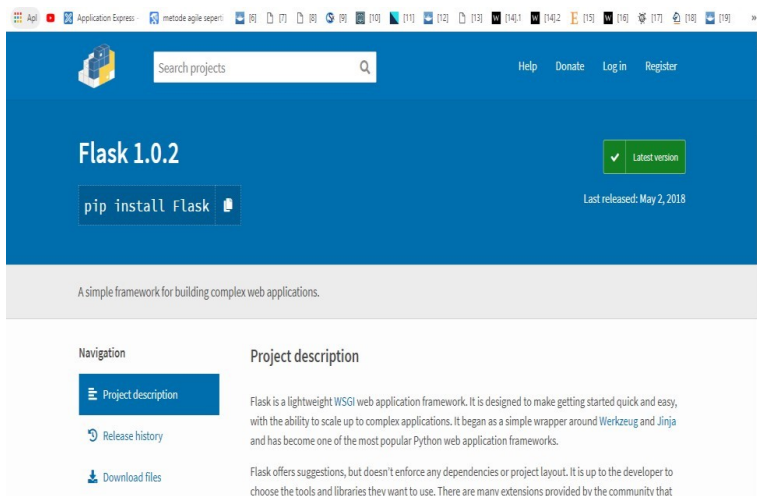
Gambar 3.6 Pengubahan Environment Python 3.6.

- Jangan lupa untuk memasukkan script dari pythonnya sehingga benar-benar bisa terbaca untuk pipnya. Silahkan klik button ok sampai selesai.
- Setelah itu, lakukan pengecekan ulang di Command Prompt maka hasilnya akan berubah menjadi versi 3.6.3
- Setelah semua tahap di atas selesai, maka silahkan lanjutkan ke tahap berikutnya.

3.3.3 Instalasi Framework Flask

Tutorial selanjutnya ialah kita akan menginstall Framework Flask di komputer/laptop kita sehingga bisa digunakan untuk tutorial selanjutnya. Perlu kita ketahui bahwa flask merupakan *microframework* dari python, dengan penggunaannya, aktivitas apapun yang kita lakukan baik pengolahan data dan lain sebagainya akan terasa lebih mudah dan rapih. Ayo kita mulai instalasinya!

1. Pertama, silahkan nyalakan PC/laptop Anda.
2. Kemudian apabila PC/laptop Anda sudah siap digunakan, silahkan buka web browser seperti Chrome, Mozilla Firefox, dsb.
3. Selanjutnya ketikkan pada web browser Anda alamat laman resmi <https://pypi.org/project/> untuk mengunduh Flask. Contohnya akan nampak seperti pada gambar 3.7.
4. Setelah proses mengunduh selesai, silahkan buka Command Prompt di PC/laptop Anda.



Gambar 3.7 Download Flask

5. Kemudian silahkan ketikkan perintah `pip install flask`.
6. Setelah mengetikkan perintah tersebut, silahkan tekan enter maka prosesnya akan berjalan. Hasilnya akan nampak seperti pada gambar 3.8.

```
C:\Users\admin\AppData\Local\Programs\Python\Python36\Scripts>pip install flask
Collecting flask
  Using cached Flask-0.12.2-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 3.8 Proses Instalasi Flask

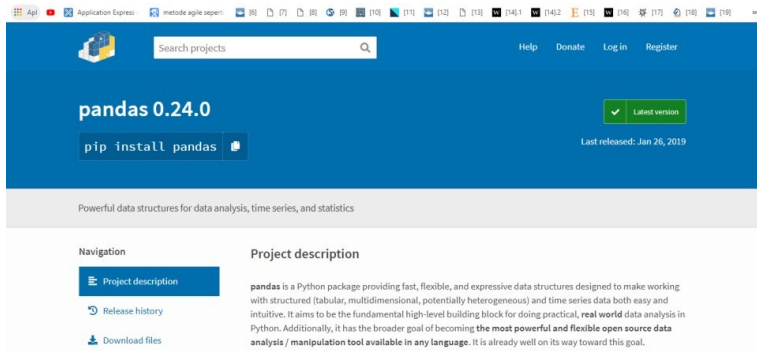
7. Proses instalasi sudah berhasil apabila tampilan pada PC/laptop Anda sudah sama seperti pada gambar 3.8. Namun jika masih terjadi *error*, coba lakukan kembali langkah-langkah instalasi flask, dan pastikan Anda telah melakukan semua proses sesuai dengan tutorial di buku ini. Setelah flask sudah terpasang pada PC/laptop Anda, silahkan untuk melanjutkan ke tutorial selanjutnya.

3.3.4 Instalasi Library Pandas

Perlu kita ketahui bahwa pandas merupakan *library* dari bahasa pemrograman python. Library ini digunakan untuk pemrosesan data analitik. Mengapa kita gunakan? Karena

memang pandas ini akan mengolah data analitik dari CSV file yang berisikan data sinyal gelombang otak yang bisa kita baca dan tangkap dari aktifitas tertentu (lampu sein saat bermotor). Ayo kita mulai instalasinya!

1. Pertama silahkan nyalakan PC/laptop Anda.
2. Kemudian apabila PC/laptop Anda sudah siap digunakan, silahkan buka web browser seperti Chrome, Mozilla Firefox, dsb.
3. Selanjutnya ketikkan pada web browser Anda alamat laman resmi <https://pypi.org/project/pandas/> mengunduh pandas. Contohnya nampak seperti pada gambar 3.9.



Gambar 3.9 Download Pandas

4. Setelah proses mengunduh selesai, silahkan buka Command Prompt di PC/laptop Anda.
5. Kemudian silahkan ketikkan perintah `pip install pandas`.
6. Setelah mengetikkan perintah tersebut, silahkan tekan enter maka prosesnya akan berjalan. Silahkan tunggu untuk melihat hasilnya yang akan nampak seperti pada gambar 3.10.
7. Proses intalasi sudah berhasil apabila tampilan pada PC/laptop Anda sudah sama seperti pada gambar ???. Namun jika masih terjadi *error*, coba lakukan kembali langkah-langkah instalasi pandas, dan pastikan Anda telah melakukan semua proses sesuai dengan tutorial di buku ini.
8. Setelah pandasnya terpasang di komputer/laptop anda maka silahkan lanjutkan ke tutorial selanjutnya.

3.3.5 Contoh Penerapan Fungsi Pada Flask Python

Flask adalah kerangka kerja aplikasi web mikro yang ditulis dalam bahasa pemrograman Python dan berdasarkan Werkzeug toolkit dan template engine Jinja2. Berlisensi BSD.

```
C:\Windows\system32>pip install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/7b/5c/51ad4b4f431b5735b239
6da44d44bd6f62dbad209e36e0be46311710162b/pandas-0.24.0-cp36-cp36m-win_amd64.whl
(8.7MB)
    100% |#####| 8.8MB 867kB/s
  Downloading https://files.pythonhosted.org/packages/31/7e/8905636f7e4f9b9d7078
aa0e701500634f832f145855a11be098d3b0fb1/numpy-1.16.0-cp36-cp36m-win_amd64.whl
(11.9MB)
    100% |#####| 11.9MB 506kB/s
Collecting pytz>=2011k (from pandas)
  Using cached https://files.pythonhosted.org/packages/61/28/1d3920e4dd50b19bc5
d24398a7cd85cc7b9a75a490570d5a30c57622d34/pytz-2018.9-py2.py3-none-any.whl
Collecting python-dateutil>=2.5.0 (from pandas)
  Using cached https://files.pythonhosted.org/packages/74/68/d87d9b36af36f44254a
8d512cbfc48369103a3b9e474be9bdf5e536abfc45/python_dateutil-2.7.5-py2.py3-none-any
.whl
Collecting six>=1.5 (from python-dateutil>=2.5.0->pandas)
  Using cached https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe
898238ce23f502a721c0ac0ecfedb80e0d88c64e9/six-1.12.0-py2.py3-none-any.whl
Installing collected packages: numpy, pytz, six, python-dateutil, pandas
Successfully installed numpy-1.16.0 pandas-0.24.0 python-dateutil-2.7.5 pytz-201
8.9 six-1.12.0
```

Gambar 3.10 Proses Instalasi Pandas

Langkah-langkah membangun Flask:

1. Pertama, pastikan bahwa python telah terinstal pada komputer.
2. Kemudian install framework python yaitu Flask pada CMD.
3. Buka CMD.
4. Kemudian ketikkan `pip install flask`. Contohnya seperti pada gambar 3.11.

```
D:\>pip install flask
Requirement already satisfied: flask in c:\python27\lib\site-packages (1.0.2)
Requirement already satisfied: Jinja2>=2.10 in c:\python27\lib\site-packages (from flask) (2.10)
Requirement already satisfied: itsdangerous>=0.24 in c:\python27\lib\site-packages (from flask) (1.1.0)
Requirement already satisfied: click>=5.1 in c:\python27\lib\site-packages (from flask) (7.0)
Requirement already satisfied: Werkzeug>=0.14 in c:\python27\lib\site-packages (from flask) (0.14.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\python27\lib\site-packages (from Jinja2>=2.10->flask) (1.1.0)
```

Gambar 3.11 Instalasi Pip Flask

5. Gambar 3.11 menunjukkan bahwa laptop anda telah memiliki flask, jadi sudah bisa digunakan.
6. Kemudian masuklah pada *text editor* seperti sublime, *notepad++*, dan lain sebagainya.
7. Kemudian silahkan untuk mulai memasukkan perintah-perintah di Flask.
8. Sebelum mencobanya, alangkah baiknya jika kita mempelajari flask terlebih dahulu.
9. Selanjutnya kita masuk ke bagian perintah python flask. Pada bagian ini kita akan mencoba membuat *hello world*. Berikut ini *source code* untuk membuat hello world 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 bisa dilihat seperti pada listing 3.2.

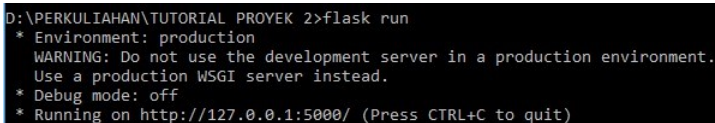
```

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

10. Masukkan perintah `import hello` untuk memanggil dan menghubungkan fungsi ke dalam file flask ini.
11. Pada file flask juga didefinisikan pemanggilan untuk fungsi hello world nya.
12. Pemanggilannya berupa, apabila parameter word difungsikan maka akan langsung terhubung dengan fungsi hello world dan tulisan hello worldnya akan tampil sesuai dengan request GetSilahkan jalankan file pada CMD, maka hasilnya akan nampak seperti gambar 3.12.



```

D:\PERKULIAHAN\TUTORIAL PROYEK 2>flask run
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

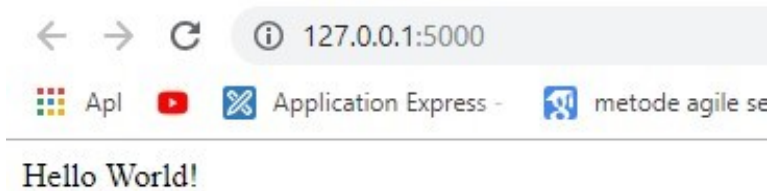
```

Gambar 3.12 Pemanggilan Fungsi Flask Opsi 1

13. Atau bisa juga seperti pada gambar 3.13.
14. Copy URL yang ada pada CMD, lalu cek URL di Web Browser. Maka hasilnya akan nampak seperti pada gambar 3.14.

```
D:\PERKULIAHAN\TUTORIAL PROYEK 2>python check.py
* Serving Flask app "check" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 179-326-677
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Gambar 3.13 Pemanggilan Fungsi Flask Opsi 2



Gambar 3.14 Output Hello World

3.4 Penanganan Error

3.4.1 Penanganan Error pada Python

3.4.1.1 Contoh kasus 1 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python. Ini contoh yang pertama. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut:

1. Pertama, pastikan bahwa python telah terinstal di PC/laptop Anda. Anda dapat menggunakan python versi 2 maupun python versi 3. Untuk tutorial pertama ini, kita akan menggunakan Python 3.
2. Selanjutnya silahkan buka text editor. Untuk text editornya bisa apa saja. Namun, untuk tutorial ini dicontohkan dengan menggunakan Sublime 3.
3. Selanjutnya silahkan buat script Pythonnya, dengan nama file satu.py. Scriptnya dicontohkan seperti pada listing 3.3.

```
1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
```

Listing 3.3 File satu.py

4. Apabila telah membuat script seperti pada listing 3.3, maka silahkan buka Command Prompt (CMD).

5. Kemudian pada CMD silahkan masuk ke tempat file anda tersimpan
6. Setelah masuk, silahkan check terlebih dahulu dengan perintah “Python2 1.py
“
7. Apabila jalan maka berhasil. Silahkan keluar dari tempat file tersebut. Pada CMD silahkan ketikkan Python3
8. Kemudian silahkan eksekusi fungsi yang ada pada file tersebut untuk melihat hasilnya. Untuk pengeksesusiannya seperti pada gambar 3.15.

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import satu
>>> satu.hello("Annisa")
>>>
```

Gambar 3.15 Eksekusi Fungsi Percobaan Pertama

9. Ketika dieksekusi seperti ini. Tapi kenapa tidak ada hasilnya? Padahal untuk fungsi dan parameternya telah disetting. Coba perhatikan lagi pada script Pythonnya. Ternyata ada yang kurang, selain inputan seharusnya ada keluaran
10. Nah untuk perintah keluarannya belum ada. Untuk perintahnya bisa Print. Silahkan ganti script satu.py seperti listing 3.4.

```
1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
4     print penggabungan
```

Listing 3.4 File baru satu.py

11. Setelah mengganti script diatas, maka silahkan buka CMD. Masuk ke tempat penyimpanan file.
12. Kemudian ketikan perintah Python3 (untuk masuk ke Pythonnya)
13. Silahkan eksekusi ulang dengan cara yang sama
14. Hasilnya seperti pada gambar 3.16.

```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import satu
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "D:\Buat Praktek\satu.py", line 5
        print penggabungan
        ^
    
```

Gambar 3.16 Eksekusi Fungsi Percobaan Kedua

15. Ternyata malah menghasilkan error yang berbeda. Nah untuk error berikut dinyatakan bahwa ada yang salah dengan perintah keluaran diatas.
16. Setelah ditelusuri ternyata salahnya pada penulisan perintah. Mengapa? Dikarenakan kita menggunakan script Python yang sangat sensitif jadi kita harus menyesuaikan perintah dengan versi Python yang digunakan. Ternyata untuk pemanggilannya tadi kita menggunakan jenis tulisan perintah untuk Python 2. Seharusnya untuk Python3 kita harus menambahkan beberapa tanda yaitu () tanda kurung pada parameternya sehingga bisa dijalankan
17. Silahkan ganti code seperti pada listing 3.5 berikut :

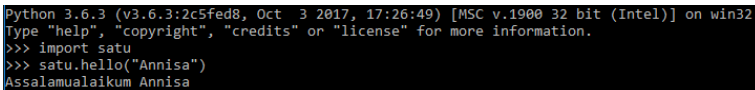
```

1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
4     print (penggabungan)

```

Listing 3.5 File baru satu.py

18. Setelah mengganti code seperti contoh diatas maka silahkan buka kembali CMD. Command Prompt lalu masuk ke file tempat penyimpanan file satu.py
19. Silahkan lakukan kembali perintah pemanggilannya seperti pada gambar 3.17 berikut :



```

Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import satu
>>> satu.hello("Annisa")
Assalamualaikum Annisa

```

Gambar 3.17 Eksekusi Fungsi Percobaan Ketiga

20. Hasil yang benar akan nampak seperti gambar tersebut. Dimana ketika telah berhasil melakukan penyettingan pada parameter yang dieksekusi maka fungsi dari file tersebut akan mengeluarkan keluaran yang sesuai
21. Keluarannya tentunya akan berupa string. Semuanya sesuai dengan settingan yang telah dibuat pada file satu.py

3.4.1.2 Contoh kasus 2 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python. Ini contoh yang kedua. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut: (Contoh kedua ini akan muncul error yang sama namun dengan penyelesaian yang berbeda dari contoh sebelumnya)

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 3.

2. Selanjutnya silahkan buka text editor. Untuk text editornya bisa apa saja. Namun, untuk tutorial ini dicontohkan dengan menggunakan Sublime 3.
3. Selanjutnya silahkan buat script Pythonnya, dengan nama file satu.py. Scriptnya dicontohkan seperti pada listing 3.6.

```
1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
```

Listing 3.6 File baru satu.py

4. Apabila telah membuat script seperti diatas maka silahkan buka CMD. CMD adalah Command Prompt, software bawaan laptop anda. Kemudian pada CMD silahkan masuk ke tempat file anda tersimpan. Setelah masuk, silahkan check terlebih dahulu dengan perintah “ Python2 1.py “
5. Apabila jalan maka berhasil. Silahkan keluar dari tempat file tersebut. Pada CMD silahkan ketikkan Python3
6. Kemudian silahkan eksekusi fungsi yang ada pada file tersebut untuk melihat hasilnya. Untuk pengeksesksiannya seperti pada gambar 3.18.

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import satu
>>> satu.hello("Annisa")
>>>
```

Gambar 3.18 Eksekusi Fungsi Percobaan Keempat

7. Ketika dieksekusi seperti ini. Tapi kenapa tidak ada hasilnya? Padahal untuk fungsi dan parameternya telah disetting
8. Coba perhatikan lagi pada script Pythonnya. Ternyata ada yang kurang, selain inputan seharusnya ada keluaran. Nah untuk perintah keluarannya belum ada. Untuk perintahnya bisa Return. Contoh yang pertama menggunakan print dan ternyata seharusnya kita menggunakan perintah Return karena lebih sering dan memang seharusnya dipakai untuk pengembalian keluaran pada fungsi Python
9. Silahkan ganti script satu.py seperti pada listing 3.7.

```
1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
4     return penggabungan
```

Listing 3.7 File baru satu.py

10. Setelah mengganti script diatas, maka silahkan buka CMD. Masuk ke tempat penyimpanan file. Kemudian ketikkan perintah Python3 (untuk masuk ke Pythonnya). Silahkan eksekusi ulang dengan cara yang sama

```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import satu
>>> satu.hello("Annisa")
Assalamualaikum Annisa
```

Gambar 3.19 Eksekusi Fungsi Percobaan Kelima

11. Hasilnya seperti pada gambar 3.19.
12. Hasil yang benar akan nampak seperti gambar 3.19.
13. Dimana ketika telah berhasil melakukan penyettingan pada parameter yang dieksekusi maka fungsi dari file tersebut akan mengeluarkan keluaran yang sesuai. Keluarannya tentunya akan berupa string
14. Semuanya sesuai dengan settingan yang telah dibuat pada file satu.py

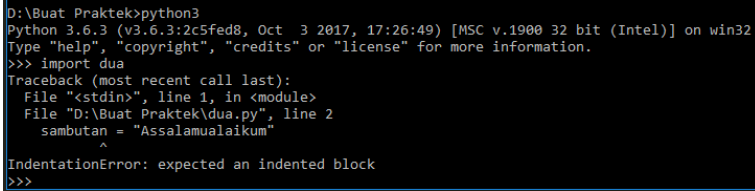
3.4.1.3 Contoh kasus 3 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python. Ini contoh yang ketiga. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut:

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 3.
2. Selanjutnya silahkan buka text editor. Untuk text editornya bisa apa saja. Namun, untuk tutorial ini dicontohkan dengan menggunakan Sublime 3
3. Selanjutnya silahkan buat script Pythonnya, dengan nama file dua.py
4. Scriptnya dicontohkan seperti pada listing3.8

```
1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
4     return (penggabungan)
```

Listing 3.8 File baru dua.py

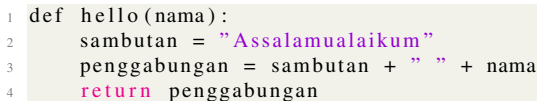
5. Apabila telah membuat script seperti diatas maka silahkan buka CMD. CMD adalah Command Prompt, software bawaan laptop anda. Kemudian pada CMD silahkan masuk ke tempat file anda tersimpan. Setelah masuk, silahkan check terlebih dahulu dengan perintah “ Python2 dua.py “
6. Apabila jalan maka berhasil, silahkan keluar dari tempat file tersebut. Pada CMD silahkan ketikkan Python3, kemudian silahkan eksekusi fungsi yang ada pada file tersebut untuk melihat hasilnya.
7. Untuk pengekseseksiannya seperti pada gambar 3.20



```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dua
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Buat Praktek\dua.py", line 2
    sambutan = "Assalamualaikum"
    ^
IndentationError: expected an indented block
>>>
```

Gambar 3.20 Eksekusi Fungsi Percobaan Keenam

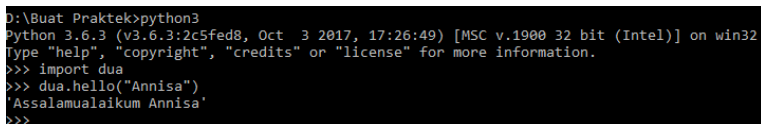
8. Ketika dieksekusi, hasilnya nampak seperti gambar 3.20.
9. Tapi kenapa tidak ada hasilnya? Padahal untuk fungsi dan parameternya telah disetting. Coba perhatikan lagi pada script Pythonnya. Ternyata ada yang kurang, selain inputan seharusnya ada keluaran. Nah untuk errornya sendiri ternyata ada pada blokingannya. Dimana Python sangatlah sensitif maka scriptnya juga harus benar. Kita harus menyamakan perintah dengan benar. Penanganan errornya yaitu memberikan SPASI ataupun TAB pada perintah untuk memasukkan inputan dengan benar. Harus sesuai dengan perintah yang lainnya agar dapat dieksekusi dengan benar
10. Silahkan ganti script dua.py seperti pada listing 3.9.



```
1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
4     return penggabungan
```

Listing 3.9 File baru dua.py

11. Setelah mengganti script diatas, maka silahkan buka CMD. Masuk ke tempat penyimpanan file. Kemudian ketikkan perintah Python3 (untuk masuk ke Pythonnya). Silahkan eksekusi ulang dengan cara yang sama
12. Hasilnya seperti pada gambar 3.21.



```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dua
>>> dua.hello("Annisa")
Assalamualaikum Annisa
>>>
```

Gambar 3.21 Eksekusi Fungsi Percobaan Ketujuh

13. Hasil yang benar akan nampak seperti gambar 3.21.
14. Dimana ketika telah berhasil melakukan penyettingan pada parameter yang dieksekusi maka fungsi dari file tersebut akan mengeluarkan keluaran yang sesuai. Keluarannya tentunya akan berupa string. Semuanya sesuai dengan settingan yang telah dibuat pada file dua.py

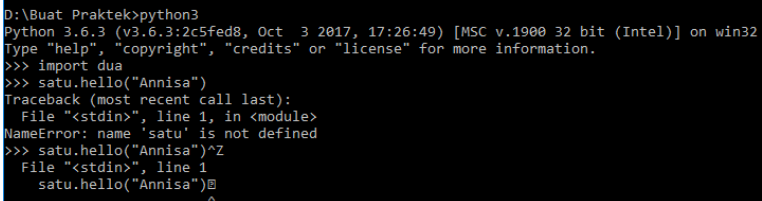
3.4.1.4 Contoh kasus 4 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python. Ini contoh yang empat. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut:

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 3.
2. Selanjutnya silahkan buka text editor. Untuk text editornya bisa apa saja. Namun, untuk tutorial ini dicontohkan dengan menggunakan Sublime 3. Selanjutnya silahkan buat script Pythonnya, dengan nama file dua.py
3. Scriptnya dicontohkan seperti pada listing 3.10

```
1 def hello(nama):
2     sambutan = "Assalamualaikum"
3     penggabungan = sambutan + " " + nama
4     return penggabungan
```

Listing 3.10 File baru dua.py

4. Setelah membuat script diatas, maka silahkan buka CMD. Masuk ke tempat penyimpanan file. Kemudian ketikkan perintah Python3 (untuk masuk ke Pythonnya)
5. Silahkan eksekusi mengikuti perintah berikut seperti pada gambar 3.22



```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dua
>>> satu.hello("Annisa")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'satu' is not defined
>>> satu.hello("Annisa")^Z
File "<stdin>", line 1
    satu.hello("Annisa")
                    ^
SyntaxError: EOL while scanning string literal
```

Gambar 3.22 Eksekusi Fungsi Percobaan Kedelapan

6. Hasil yang benar akan nampak seperti gambar tersebut.
7. Terjadi error pada eksekusi tersebut. Ternyata untuk error ini terjadi karena salah perintah dalam pengeksekusian. Pada perintah tersebut, kita memasukkan “ import dua “. Maka akan masuk ke file dua.py
8. Namun, setelah itu, kita malah memasukkan perintah “ satu.hello(“annisa”)”. Yang kalau diartikan, maka perintah tersebut ditujukan untuk file satu.py. Maka dari itu, kita harus mengganti perintah yang kita masukkan dalam pengeksekusian. Kalau mengeksekusi file dua.py maka import dan arahan perintahnya harus ke file dua.py

```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import dua
>>> dua.hello("Annisa")
'Assalamualaikum Annisa'
>>>
```

Gambar 3.23 Eksekusi Fungsi Percobaan Kesembilan

9. Silahkan eksekusi lagi dengan perintah berikut seperti pada gambar 3.23.
10. Apabila perintah telah sesuai, maka akan menampilkan hasil yang seperti contoh diatas. Keluarannya tentunya akan berupa string. Semuanya sesuai dengan settingan yang telah dibuat pada file dua.py. Dengan permasalahan yang sama, apabila kita secara tidak sengaja menemukan error juga ketika mengeksekusi file satu.py maka penyelesaiannya mirip seperti contoh diatas
11. Dimisalkan untuk file satu.py errornya seperti pada gambar 3.24.

```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import satu
>>> dua.hello("Annisa")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'dua' is not defined
>>>
```

Gambar 3.24 Eksekusi Fungsi Percobaan Kesepuluh

12. Nah, tentunya kita sudah tau bahwa salahnya pada pemanggilan yang harusnya mengarah kepada file satu.py. Silahkan dibenarkan lagi.
13. Ikuti contoh eksekusi seperti pada gambar 3.25 untuk mendapatkan hasil yang benar.

```
D:\Buat Praktek>python3
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import satu
>>> satu.hello("Annisa")
'Assalamualaikum Annisa'
```

Gambar 3.25 Eksekusi Fungsi Percobaan Kesebelas

14. Nah, penyelesaiannya seperti diatas. Kita harus memperhatikan perintah yang kita gunakan untuk eksekusi.

3.4.1.5 Contoh kasus 5 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python. Ini contoh yang kelima. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk

kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut: (Contoh Python ini digunakan dalam pembuatan script python untuk pembacaan gelombang otak)

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 2.
2. Selanjutnya silahkan buka text editor. Untuk text editornya bisa apa saja. Namun, untuk tutorial ini dicontohkan dengan menggunakan Sublime 3.
3. Selanjutnya silahkan buat script Pythonnya, dengan nama file coba.py
4. Scriptnya dicontohkan seperti pada listing 4.23.

```

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 3.11 File coba.py

5. Pada code diatas mendefinisikan perintah dimana fungsinya akan dipanggil dan berhubungan dengan mindwave.py. Seperti contoh perintah headset.connect dimana fungsinya akan diproses di mindwave.py sehingga apabila hasilnya benar dan sesuai maka akan muncul kata “connecting” seperti perintah print diatas. Apabila alat dan code sesuai maka fungsinya akan jalan, baik itu connected, standby dll.
6. Nah untuk pembacaan sinyalnya, kita menggunakan variabel dari mindwave.py. Untuk variabelnya bisa dicoba yang attention / blink / meditation dll secara bersamaan juga bisa asal perintahnya benar.
7. Perhatikan code ini:

```

1 headset = mindwave.Headset('COM4', '1425')
2 time.sleep(2)

```

Listing 3.12 File coba.py

8. Untuk mengkoneksikan code dengan alat sehingga terhubung maka serial dari alat tersebut harus benar. Portnya yaitu COM4 dapat dilihat dengan cara :
 - Buka pengaturan laptop anda
 - Kemudian pilih device manager
 - Lalu pilih alat yang anda gunakan misalnya : Mindwave Neurosky
 - Apabila telah diklik maka akan muncul portnya yaitu COM4
 - Untuk serialnya dapat dilihat dari alatnya sendiri
 - Pengecekannya yaitu di dalam tempat baterai. Serialnya 1425
 - Port dan serial itu beda-beda tergantung dari alatnya masing-masing
9. Time.sleep itu gunanya untuk memberikan jeda pembacaan sinyal sehingga nilai gelombang yang dibaca tidak terlalu cepat dan dapat lebih terpantau. Lebih jelasnya lagi, mengapa time.sleepnya bernilai (0.001953125) dikarenakan :
10. Silahkan perhatikan ini:

```

1 print "raw_value: %s : %s" % (headset.raw_value , a)
2 writer.writerow({'RawValue': headset.raw_value , 'sign' : a})

```

Listing 3.13 File coba.py

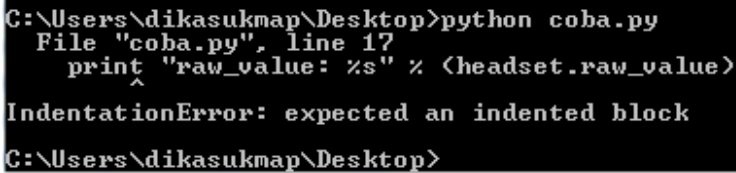
Measures

Outputs 12 bit Raw-Brainwaves (3 - 100Hz) with Sampling rate at 512Hz
 Outputs EEG power spectrums (Alpha, Beta, etc.)
 Outputs NeuroSky proprietary eSense meter such as Attention,
 Meditation, and other future meters
 EEG/ECG signal quality analysis (can be used to detect poor contact and
 whether the device is off the head)

Gambar 3.26 Deskripsi Pengukuran

11. Sampling rate dari alat yang digunakan ialah 512Hz jadi tentunya kita harus mengikuti dan menyesuaikannya dengan baik, untuk code dan alatnya. Nah kemudian 512Hz tersebut kita bagi dengan 1 (detik) jadi hasilnya akan menjadi jeda antara sinyal yang telah ditangkap/ dibaca dari alat. $1/512 = 0.001953125$. nah nilai itu yang akan menjadi jeda pada code. Selain itu. Kita juga harus memastikan apakah benar bahwa dalam 1 detik itu ada 512 data yang dibaca dan ditangkah oleh alat Dan sesuai hasil dari penguijannya sudah dibuktikan dalam setiap detiknya ada sekitar 480-520an data perdetik.
12. Setelah semua langkah di atas telah dilakukan, kita akan mencoba running coba.py melalui cmd, dengan cara :

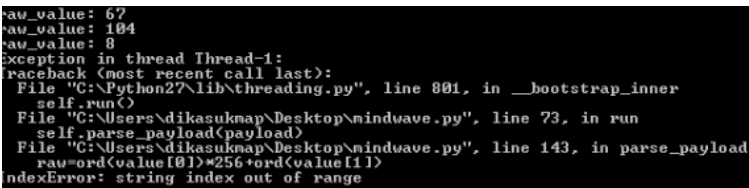
13. Buka Command Prompt di komputer/laptop anda masing-masing. Silahkan pada CMD arahkan ke folder / directory tempat file coba.py anda simpan
14. Namun sebelum itu pastikan alat sudah terpasang baterai dan lampu indikator berwarna merah. Masukkan USB ke port yang ada di laptop anda. USBnya tentu milik dari alat mindwavenya
15. Kemudian silahkan masukkan perintah python coba.py
16. Apabila pas anda jika terjadi ERROR seperti gambar 3.27.



```
C:\Users\dikasukmap\Desktop>python coba.py
File "coba.py", line 17
    print "raw_value: %s" % <headset.raw_value>
    ^
IndentationError: expected an indented block
C:\Users\dikasukmap\Desktop>
```

Gambar 3.27 Error pada program coba.py

17. ERROR diatas sebenarnya menunjukkan bahwa indentasi codingan anda tak sesuai maka yang perlu dilakukan ialah rapikan code anda sesuai dengan perintah-perintah terkait. Biasanya ERROR ini terjadi ketika code anda tak berada pada tempatnya maka silahkan sesuaikan dengan SPACI ataupun TAB.
18. Silahkan Run kembali
19. Ketika di Run, dan terjadi ERROR ditengah jalan seperti pada gambar 3.28



```
raw_value: 67
raw_value: 104
raw_value: 8
Exception in thread Thread-1:
Traceback (most recent call last):
  File "C:\Python27\lib\threading.py", line 801, in __bootstrap_inner
    self.run()
  File "C:\Users\dikasukmap\Desktop\mindwave.py", line 73, in run
    self.parse_payload(payload)
  File "C:\Users\dikasukmap\Desktop\mindwave.py", line 143, in parse_payload
    raw=ord(value[0])*256+ord(value[1])
IndexError: string index out of range
```

Gambar 3.28 Keterangan error pada program coba.py

20. Maka anda harus menerapkan Try and Except pada codingan mindwavenya.
21. Contohnya seperti pada listing 3.14

```
1 if code == RAW_VALUE:
2     try:
3         anu = value[0]
4         itu = value[1]
5     except IndexError:
6         anu = "0"
```

```
7 itu = "0"
```

Listing 3.14 File try_except.py

22. Penggunaan try dan except yaitu untuk menangani error pada code yang dihindari untuk terjadi ketika sedang membaca sinyal. Dimisalkan penanganan errornya untuk penanganan index. Selain index sebenarnya bisa juga untuk menangani error pada IO, database, dictionary dan lain-lain.
23. Setelah penerapan tersebut silahkan re-run file coba.py. Maka tidak akan terjadi error lagi.

3.4.1.6 Contoh kasus 6 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python. Ini contoh yang keenam. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut: (Contoh Python ini digunakan dalam pembuatan script python untuk pembacaan gelombang otak)

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 2.
2. Selanjutnya silahkan buka text editor. Untuk text editornya bisa apa saja. Namun, untuk tutorial ini dicontohkan dengan menggunakan Sublime 3.
3. Selanjutnya silahkan buat script Pythonnya, dengan nama file coba.py
4. Scriptnya dicontohkan seperti pada listing 3.15

```
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
9 headset.connect()
10 print "Connecting ..."
11
12 while headset.status != 'connected':
13     time.sleep(0.5)
14     if headset.status == 'standby':
15         headset.connect()
16         print "Retrying connect..."
17 print "Connected."
18
19 while True:
20     print "raw_value: %s : %s" % (headset.raw_value, a)
```

Listing 3.15 File coba_lagi.py

5. Pada code diatas mendefinisikan perintah dimana fungsinya akan dipanggil dan berhubungan dengan mindwave.py
6. Seperti contoh perintah headset.connect dimana fungsinya akan diproses di mindwave.py sehingga apabila hasilnya benar dan sesuai maka akan muncul kata “connecting” seperti perintah print diatas. Apabila alat dan code sesuai maka fungsinya akan jalan, baik itu connected, standby dll.
7. Nah untuk pembacaan sinyalnya, kita menggunakan variabel dari mindwave.py. Sebenarnya penjelasan lengkap untuk tutorial pembangunan script python ini sama dengan contoh kasus kelima
8. Namun, akan muncul error yang berbeda.
9. Setelah semua langkah di atas telah dilakukan, kita akan mencoba running coba.py melalui cmd, dengan cara :
10. Buka Command Prompt di komputer/laptop anda masing-masing. Silahkan pada CMD arahkan ke folder / directory tempat file coba.py anda simpan.
11. Namun sebelum itu pastikan alat sudah terpasang baterai dan lampu indikator berwarna merah. Masukkan USB ke port yang ada di laptop anda. USBnya tentu milik dari alat mindwavenya
12. Kemudian silahkan masukkan perintah python coba.py. Maka akan muncul error dimana hasilnya tidak akan muncul
13. Hal tersebut dikarenakan ada perintah yang kurang dalam script. Seharusnya kita juga memasukkan perintah seperti pada listing 3.16 berikut:

```
1 writer.writerow({ 'RawValue': headset.raw_value , 'sign': a })
```

Listing 3.16 File coba_lagi.py

14. Silahkan Run kembali, maka file akan mengeluarkan sinyal datanya.

3.4.2 Penanganan Error dalam Flask

3.4.2.1 Contoh kasus 1 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python Flask. Ini contoh yang pertama. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut:

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 3.
2. Selanjutnya silahkan buka text editor. Untuk text editornya bisa apa saja. Namun, untuk tutorial ini dicontohkan dengan menggunakan Sublime 3.

3. Selanjutnya silahkan buat script Python Flasknya. Script lengkap Python Flasknya dapat ditemukan di Tutorial Proyek yang telah dibuat sebelumnya
4. Contoh Potongan scriptnya seperti pada listing 3.17.

```

1 from flask import Flask, request, jsonify
2
3 # import fungsi
4 from load_data import load_data
5 from get_all_data import get_all_data
6 from get_column import get_column
7 from get_top_five import get_top_five
8 from sorting import sorting
9 from convert_json import convert_json
10 from delete_column import delete_column
11 from delete_all_data import delete_all_data
12 from insert_data import insert_data
13 from get_info import get_info
14 from get_row import get_row
15 from get_row_field import get_row_field
16 from update_data import update_data
17 from delete_data import delete_data
18 from get_row_json import get_row_json
19 from get_all_data_json import

```

Listing 3.17 File import.py

5. Apabila telah membuat script seperti diatas maka silahkan buka CMD. CMD adalah Command Prompt, software bawaan laptop anda. Kemudian pada CMD silahkan masuk ke folder tempat penyimpanan file. Kemudian aktifkan dulu enviroment dari file tersebut sehingga Flasknya dapat berjalan.
6. Untuk pengeksekusiannya seharusnya nampak seperti pada gambar 3.29.

```

(env) D:\Buat Praktek\done>python3 main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 201-749-500
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Gambar 3.29 Eksekusi Program import.py

7. Namun, ketika dieksekusi malah hasilnya nampak seperti gambar berikut:
8. Error tersebut menyatakan bahwa kita tidak mendefinisikan module pandas pada file codingan yang telah dibuat
9. Maka kita harus memasukkan perintah seperti pada listing 3.18.

```
1 import pandas as pd
```

Listing 3.18 File import.py

10. Perintah tersebut ditaruh setelah perintah pengimporan module yang lain seperti jsonify, request maupun flask. Setelah itu silahkan buka CMD kembali. Pada CMD silahkan lakukan perintah yang sama yaitu menjalankan file Main.py dengan mengaktifkan environmentnya terlebih dahulu.
11. Contohnya nampak seperti pada gambar 3.30.

```
Traceback (most recent call last):
  File "main.py", line 6, in <module>
    from load_data import load_data
  File "D:\Buat Praktek\done\load_data.py", line 1, in <module>
    import pandas
ModuleNotFoundError: No module named 'pandas'
```

Gambar 3.30 Eksekusi Program import.py

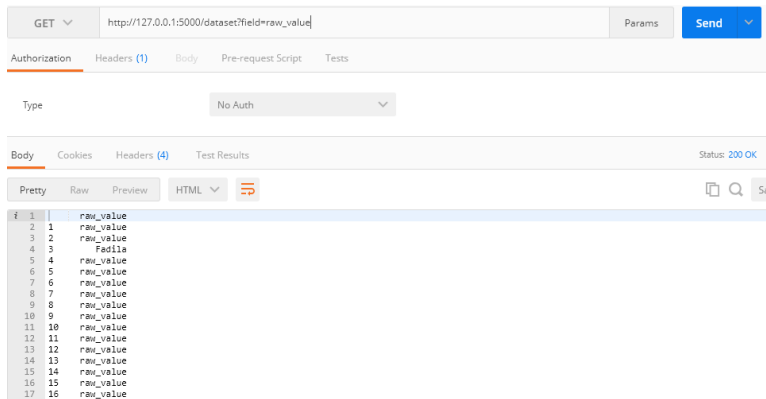
12. Apabila perintah yang anda lakukan nampak seperti pada contoh diatas, berarti flasknya telah berhasil berjalan. Silahkan untuk pengujiannya bisa dilihat dengan masuk ke dalam URL yang diperlihatkan pada CMD. Pengujian bisa dilakukan dengan memasukkan URL ke dalam web browser biasa maupun Postman
13. Contoh pengujian ini akan menggunakan Postman biar lebih mudah dan rapih.
14. Hasilnya akan nampak seperti gambar 3.31, dimana perintahnya telah disesuaikan dengan request yang ada dalam file Main.py

```
D:\Buat Praktek\done>env\scripts\activate
(env) D:\Buat Praktek\done>python3 main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 201-749-500
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Gambar 3.31 Eksekusi Program import.py

15. Apabila mendapatkan hasil seperti gambar 3.32, maka tutorial ini berhasil.

3.4.2.2 Contoh kasus 2 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python Flask. Ini contoh yang kedua. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut:



Gambar 3.32 Hasil Eksekusi Program import.py

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 3.
2. Kita akan mengeksekusi main.py, sehingga kita bisa menemukan beberapa error. Karena kita akan mengeksekusi main.py, kita juga akan membangun terlebih dahulu fungsi yang akan dimasukkan dalam main.py (flask).
3. Silahkan buka text editor. Gunakan sublime 3.
4. Kemudian silahkan buat script seperti pada listing 3.21.

```
1 def get_top_five ( dataframe , order_by ) :
2     return dataframe . head ()
```

Listing 3.19 File get_top_five.py

5. Fungsi tersebut disimpan dengan nama file get_top_five.py. Fungsi tersebut telah dimasukkan kedalam Flask yaitu main.py sehingga ketika dijalankan fungsinya akan dapat dieksekusi di file flask yang sama dengan fungsi yang lain. Fungsinya yaitu untuk menampilkan 5 data teratas ataupun terbawah pada file CSV.
6. Adapun code flask yang mengeksekusi fungsi ini ialah nampak seperti pada listing 3.20.

```
1 @app.route ( '/dataset' , methods=[ 'GET' , 'POST' , 'DELETE' ])
2 def dataset () :
3     args = request . args
4     response = dataframe
5     if request . method == 'GET' :
6         if 'topfive' in args :
```

Listing 3.20 File main.py

7. Silahkan masuk ke CMD. Jalankan file main.py dengan masuk ke folder penyimpanannya. Kemudian aktifkan dulu environment dari file tersebut sehingga Flasknya dapat berjalan.
8. Untuk pengekseseksiannya seharusnya nampak seperti pada gambar 3.33.

```
(env) D:\Buat Praktek\done>python3 main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 201-749-500
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

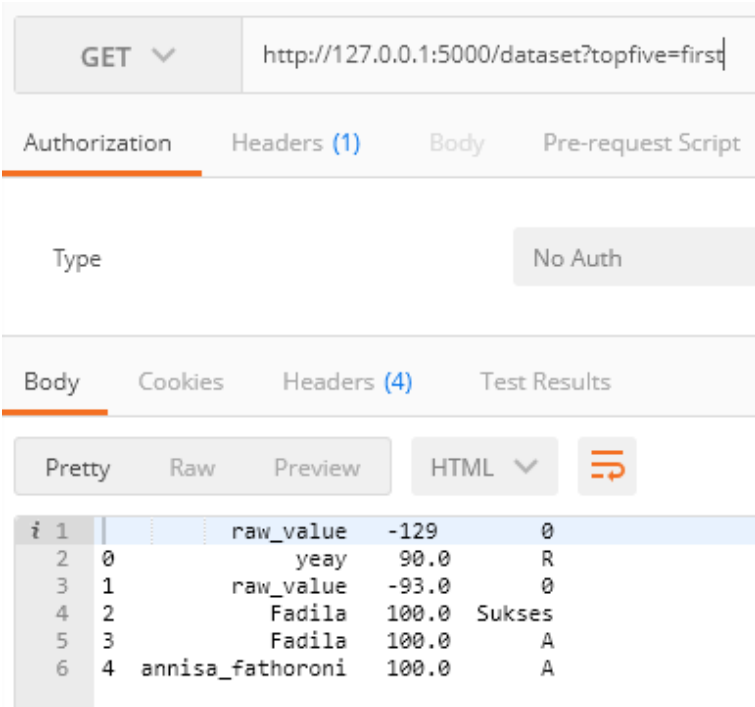
Gambar 3.33 Eksekusi Program main.py

9. Silahkan untuk pengujiannya bisa dilihat dengan masuk ke dalam URL yang diperlihatkan pada CMD. Pengujian bisa dilakukan dengan memasukkan URL ke dalam web browser biasa maupun Postman. Contoh pengujian ini akan menggunakan Postman biar lebih mudah dan rapih.
10. Hasilnya akan nampak seperti pada gambar 3.34, dimana perintahnya telah disesuaikan dengan request yang ada dalam file main.py.
11. Pengujian diatas menunjukkan hasil untuk 5 angka teratas, dengan perintah `/dataset?topfive=first`
12. Kemudian untuk pengujian hasil 5 angka terbawah, dengan perintah `"/dataset?topfi` seperti pada gambar 3.35.
13. Hasil pengujiannya tidak berbeda dengan perintah `?topfive=first`. Mengapa? Terjadi error atau kesalahan disini. Ternyata setelah dicek, pada fungsi `get_top_five.py` hanya mendefinisikan fungsi untuk keluaran 5 data teratas (headframe). Apabila kita ingin tetap menampilkan 5 data terbawah maka kita harus menambahkan perintah berikut pada file `get_top_five.py` seperti pada listing 3.21.

```
1
2 if order_by == 'first' else dataframe.tail()
```

Listing 3.21 File `get_top_five.py`

14. Setelah menambahkan perintah tersebut silahkan disimpan. Kemudian buka kembali CMD. Kemudian aktifkan ulang environment dari main.py
15. Lalu silahkan masuk ke Postman untuk pengujiannya
16. Dengan perintah `/dataset?topfive=last`, maka hasilnya nampak seperti pada gambar 3.36.



GET ▼ http://127.0.0.1:5000/dataset?topfive=first

Authorization Headers (1) Body Pre-request Script

Type No Auth

Body Cookies Headers (4) Test Results

Pretty Raw Preview HTML ▼ ⇒

i	1	raw_value	-129	0
	2	yeay	90.0	R
	3	raw_value	-93.0	0
	4	Fadila	100.0	Sukses
	5	Fadila	100.0	A
	6	annisa_fathoroni	100.0	A

Gambar 3.34 Hasil Eksekusi Program main.py

GET ▼ http://127.0.0.1:5000/dataset?topfive=last

Authorization Headers (1) Body Pre-request Script

Type No Auth

Body Cookies Headers (4) Test Results

Pretty Raw Preview HTML ▼

i 1		raw_value	-129	0
2	0	yeay	90.0	R
3	1	raw_value	-93.0	0
4	2	Fadila	100.0	Sukses
5	3	Fadila	100.0	A
6	4	annisa_fathoroni	100.0	A

Gambar 3.35 Hasil Eksekusi Program main.py

DELETE ▼ http://127.0.0.1:5000/dataset?field=sign

Authorization Headers (1) Body ● Pre-request Script Tests

Type No Auth ▼

Body Cookies Headers (5) Test Results

Pretty Raw Preview HTML ▼

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <title>KeyError: '"[sign]' not found in axis' // Werkzeug Debugger</title>
6 <link rel="stylesheet" href="?__debugger__=yes&cmd=resource&f=style.css"
7 type="text/css">
8 <!-- We need to make sure this has a favicon so that the debugger does
9 not by accident trigger a request to /favicon.ico which might
10 change the application state. -->
11 <link rel="shortcut icon"
12 href="?__debugger__=yes&cmd=resource&f=console.png">
13 <script src="?__debugger__=yes&cmd=resource&f=jquery.js"></script>
14 <script src="?__debugger__=yes&cmd=resource&f=debugger.js"></script>
15 <script type="text/javascript">
16 var TRACEBACK = 160088912,
17     CONSOLE_MODE = false,
18     EVAI FX = true.

```

Gambar 3.36 Hasil Eksekusi Program main.py

3.4.2.3 Contoh kasus 3 Untuk tutorial ini, akan dicontohkan penanganan error untuk contoh Python Flask. Ini contoh yang ketiga. Sebelum menangani error. Akan dijelaskan terlebih dahulu tutorial untuk pembangunan Pythonnya kemudian akan masuk kepada penanganan error yang ada. Silahkan simak dan ikuti tutorial berikut:

1. Pertama-tama silahkan pastikan bahwa di laptop anda telah terinstall Python. Python yang digunakan ada Python 2 dan Python 3. Untuk tutorial pertama ini, kita akan menggunakan Python 3.
2. Kita akan mengeksekusi main.py, sehingga kita bisa menemukan beberapa error. Karena kita akan mengeksekusi main.py, kita juga akan membangun terlebih dahulu fungsi yang akan dimasukkan dalam main.py (flask)
3. Silahkan buka text editor. Gunakan sublime 3
4. Kemudian silahkan buat script seperti pada listing 3.22.

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

Listing 3.22 File delete_column.py

5. Fungsi tersebut disimpan dengan nama file delete_column.py
6. Fungsi tersebut telah dimasukkan kedalam Flask yaitu main.py sehingga ketika dijalankan fungsinya akan dapat dieksekusi di file flask yang sama dengan fungsi yang lain. Fungsinya yaitu untuk menghapus data sesuai dengan kolom yang telah dijadikan parameter/argumen pada file fungsi tersebut
7. Adapun code flask yang mengeksekusi fungsi ini ialah nampak seperti pada listing 3.23.

```
1 elif request.method == 'DELETE':
2     if not len(args) is 0:
3         if 'field' in args:
4             response = delete_column(response, args['field'])
```

Listing 3.23 File main.py

8. Silahkan masuk ke CMD. Jalankan file main.p dengan masuk ke folder penyimpanannya. Kemudian aktifkan dulu environment dari file tersebut sehingga Flasknya dapat berjalan.
9. Untuk pengekseseksiannya seharusnya nampak seperti pada gambar dibawah:
10. Silahkan untuk pengujiannya bisa dilihat dengan masuk ke dalam URL yang diperlihatkan pada CMD. Pengujian bisa dilakukan dengan memasukkan URL ke dalam web browser biasa maupun Postman. Contoh pengujian ini akan menggunakan Postman biar lebih mudah dan rapih.
11. Hasilnya akan nampak seperti gambar 3.37, dimana perintahnya telah disesuaikan dengan request yang ada dalam file main.py.

```
(env) D:\Buat Praktek\done>python3 main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 201-749-500
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

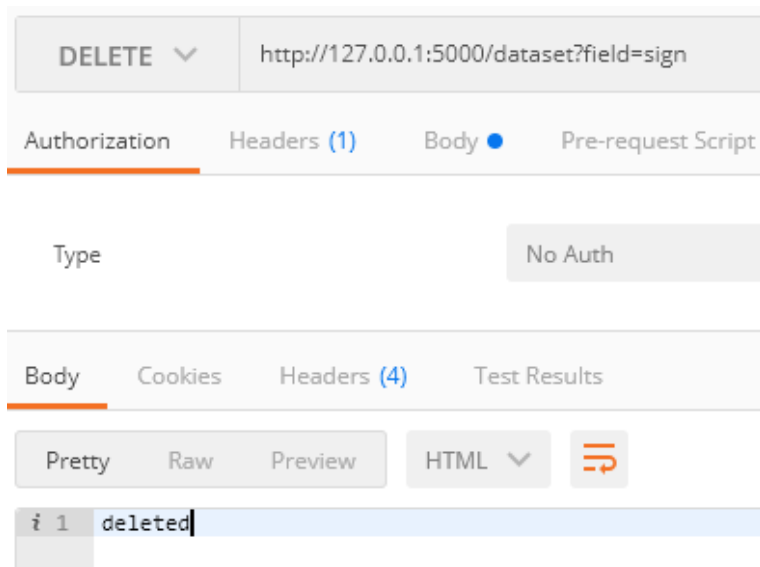
Gambar 3.37 Eksekusi Program main.py

12. Terjadi error.
13. Error terjadi karena pada file delete_all_data.py yang terhubung dengan fungsi delete_column.py tidak sesuai. Perintah yang dimasukkan tidak di definisikan pada file delete_all_data.py yaitu untuk menghapus field “ sign “. Maka harus ditambahkan parameter/argument tersebut
14. Dan tampilan delete_all_data.py nampak seperti pada listing 3.24.

```
1 # menghapus semua data dari csv
2 def delete_all_data(dataframe):
3     dataframe.drop(['raw_value', 'data', 'sign'], axis=1)
```

Listing 3.24 File delete_all_data.py

15. Nah bisa dilihat bahwa pada fungsi didefinisikan field apa saja yang bisa dihapus. Tentunya field ini akan dapat dieksekusi melalui file delete_column.py.
16. Ada 3 kolom yang menjadi field yang dapat dieksekusi yaitu Raw_value, data dan sign. Nah parameter signnya sudah ada, maka apabila dijalankan kembali tidak akan terjadi error.
17. Silahkan kita check ulang fungsinya. Masuk lagi ke CMD untuk merefresh atau membuka kembali file Flasknya yaitu Main.py. Jangan lupa untuk mengaktifkan environmentnya seperti pada contoh-contoh yang telah dijelaskan sebelumnya
18. Hasil pengujiannya nampak seperti pada gambar 3.38.
19. Apabila hasilnya nampak seperti pada gambar 3.38, maka fungsinya telah berhasil dieksekusi sesuai request yang ada di Python Flask
20. Maka telah selesai pula tutorial penanganan error ini.



Gambar 3.38 Hasil Eksekusi 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[6]. 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 :

```
<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. 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 4.2 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 4.3 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 4.4 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 4.5 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 4.6 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 4.7 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 4.8 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 4.9 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 4.10 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 4.11 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 4.12 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 4.13 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 4.14 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 4.15 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 4.16 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 4.17 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 4.18 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 4.19 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 4.20 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 4.21 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 4.22 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. Yang akan dibahas disini adalah MMethod GETnya saja. Tutorialnya sebagai berikut :

- Setelah melakukan perintah 4.22
- Kemudian masukkan perintah yang sesuai
- Pendefinisian endpoint dataset
- Definisikan request atau permintaan metode yang digunakan. Ada Get
- Pada Method GET : Silahkan masukkan beberapa argument yang akan diolah.
- Argumentnya ialah parameter field, topfive, sort dan format
- Penjelasan Parameter :
 - All : Mengambil semua data dari file csv
 - Field : Mengambil data berdasarkan parameter field yaitu ada Raw-Value, sign dan daya
 - Topfive : Mengambil 5 data berdasarkan data paling atas dan 5 data paling bawah
 - Sort : Mengambil data dan diurutkan sesuai dengan urutan ascending (kecil ke besar) dan descending (besar ke kecil)
 - Format : Menampilkan data dengan format tulisan JSON dan Raw
- Selanjutnya pastikan setiap argument memberikan respon sesuai dengan fungsi masing-masing
- Kemudian reloaddata akan menampilkan data terbaru setelah dilakukan perintah delete
- Silahkan uji endpointnya sesuai dengan parameter tersebut menggunakan POSTMAN maka hasilnya akan seperti berikut :

4.4 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 [7]. 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*[8]. 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 :

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 pengekseskuan file python yang berupa gunsi yang telah dibuat. Berikut langkah-langkahnya :

- Pertama-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 4.23, 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 4.23 Contoh kode untuk membaca sinyal gelombang otak

- Script Code pada listing 4.23 mendefinisikan perintah dimana fungsinya akan dipanggil dan berhubungan dengan mindwave.py
- Seperti contoh perintah headset.connect dimana fungsinya akan diproses di mindwave.py sehingga apabila hasilnya benar dan sesuai maka akan muncul kata “connecting” seperti perintah print diatas
- Apabila alat dan code sesuai maka fungsinya akan jalan, baik itu connected, standby dll.
- Untuk pembacaan sinyalnya, kita menggunakan variabel dari mindwave.py
- Untuk variabelnya bisa dicoba yang attention / blink / meditation dll secara bersamaan juga bisa asal perintahnya benar.
- Perhatikan script code pada listing 4.24 ini :

```

1
2 headset = mindwave.Headset('COM4', '1425')

```

Listing 4.24 Contoh1 kasus 2

- Untuk mengkoneksikan code dengan alat sehingga terhubung maka serial dari alat tersebut harus benar. Portnya yaitu COM4 dapat dilihat dengan cara :

1. Buka pengaturan laptop anda
 2. Kemudian pilih device manager
 3. Lalu pilih alat yang anda gunakan misalnya : Mindwave Neurosky
 4. Apabila telah diklik maka akan muncul portnya yaitu COM4
 5. Untuk serialnya dapat dilihat dari alatnya sendiri
 6. Pengecekannya yaitu di dalam tempat baterai. Serialnya 1425 g. Port dan serial itu beda-beda tergantung dari alatnya masing-masing
- Time.sleep itu gunanya untuk memberikan jeda pembacaan sinyal sehingga nilai gelombang yang dibaca tidak terlalu cepat dan dapat lebih terpantau. Lebih jelasnya lagi, mengapa time.sleepnya bernilai (0.001953125) dikarenakan :

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. S. S. Azzahra, "Karya tulis," 2017.
6. N. R. Wyler, *Aggressive network self-defense*. Elsevier, 2005.
7. A. Kadir, "Dasar pemrograman python," *Yogyakarta: ANDI*, 2005.
8. P. Lokhande, F. Aslam, N. Hawa, J. Munir, and M. Gulamgaus, "Efficient way of web development using python and flask," 2015.

Index

disruptif, xxxi
modern, xxxi