

# **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

---



# DAFTAR ISI

---





# DAFTAR GAMBAR

---



# DAFTAR TABEL

---



# Listings

---



# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa





# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan flask sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat*  
*Februari, 2019*



# ACKNOWLEDGMENTS

---

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

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

R. M. A.



# ACRONYMS

---

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



# GLOSSARY

---

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





# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

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

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[?].

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



# BAB 1

---

## JUDUL BAGIAN PERTAMA

---

### 1.1 Perintah Navigasi

Perintah navigasi direktori



## BAB 2

---

## JUDUL BAGIAN KEDUA

---

### 2.1 Perintah Navigasi

Perintah navigasi direktori





## BAB 3

---

# PENGENALAN FLASK

---

### 3.1 Definisi *Micro Framework*

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

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

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

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

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

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

## 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 [?].

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 [?].

Kelebihan:

1. Dukungan bawaan

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

2. Koneksi serentak

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

3. Kinerja tinggi.

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

Tornado memiliki beberapa modul utama seperti:

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

### 3.2.4 Falcon

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

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

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

Kelebihan:

#### 1. Cepat

Salah satu persyaratan paling penting dari cloud API adalah mereka harus menanggapi 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.

Beberapa fitur yang dimiliki oleh Web2py antara lain:

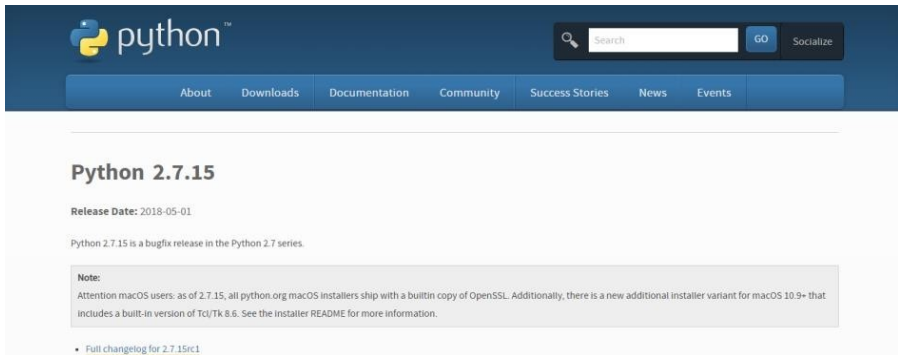
1. Dibuat oleh komunitas terpercaya.
2. Selalu backward compatible.
3. Mudah digunakan.
4. Dapat berjalan di banyak sistem operasi.
5. Dapat berjalan di banyak web server.
6. Dapat "berbicara" ke SQLite3, PostgreSQL, MySQL, MSSQL, FireBird, Oracle, IBM DB2, Informix, Ingres, dan Google App Engine.
7. Aman dari cross site scripting, injection flaws, dan eksekusi file berbahaya.
8. Mengajarkan penggunaanya 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

## 3.3 Instalasi dan Hello World di Flask

### 3.3.1 Instalasi Python 2.7.

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

1. Pertama-tama silahkan download software dari python versi 2 di laptop anda. Download python versi 2.7.15 dari situs web resminya yaitu <https://www.python.org/>.



**Gambar 3.1** Download Softfile Python 2.7.

Silahkan sesuaikan dengan kapasitas laptop anda, bisa yang win 32 atau yang win 64 (32 bit / 64 bit). Contoh downloadnya seperti pada gambar 3.1.

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

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

**Gambar 3.2** Proses Instalasi Flask

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

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

**Listing 3.1** Contoh kode program hello.py

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

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