

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

DAFTAR ISI

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Judul Bagian Pertama	1
1.1 Perintah Navigasi	1
2 Judul Bagian Kedua	3
2.1 Perintah Navigasi	3
	ix

3	Pengenalan Flask	5
3.1	Definisi <i>Micro Frameworki</i>	5
3.2	Jenis-Jenis <i>Framework</i> Python serta Kelebihan dan Kekurangan	6
3.2.1	Django	6
3.3	Instalasi dan Hello World di Flask	8
3.3.1	Instalasi Python 2.7.	8
	Daftar Pustaka	11

DAFTAR GAMBAR

3.1	Download Softfile Python 2.7.	8
3.2	Froses Instalasi Flask	9

DAFTAR TABEL

Listings

3.1	Contoh kode program hello.py	8
3.2	Contoh kode program main.py	9

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan flask sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

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

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

R. M. A.

ACRONYMS

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

GLOSSARY

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

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

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

Informatics Research Center
Bandung, Jawa Barat, Indonesia

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

Rumusnya cuman sebagai contoh aja biar keren[1].

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

BAB 1

JUDUL BAGIAN PERTAMA

1.1 Perintah Navigasi

Perintah navigasi direktori

BAB 2

JUDUL BAGIAN KEDUA

2.1 Perintah Navigasi

Perintah navigasi direktori

BAB 3

PENGENALAN FLASK

3.1 Definisi *Micro Framework*

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

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

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

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

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

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

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

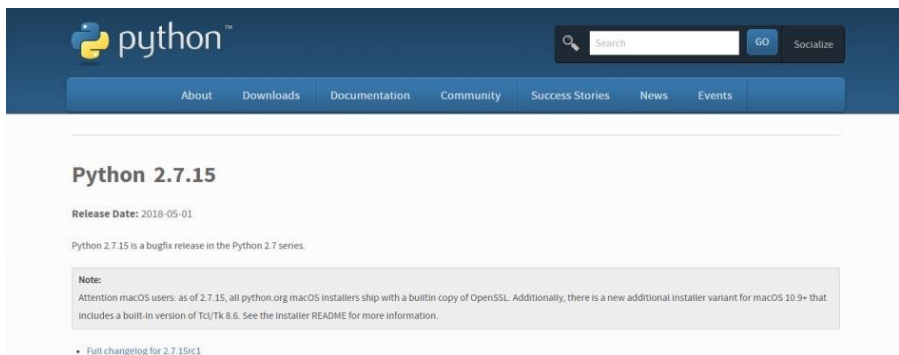
3. Semuanya didasarkan pada Django ORM
4. Komponen dikerahkan bersama
5. Pengetahuan tentang sistem lengkap diperlukan untuk bekerja.

3.3 Instalasi dan Hello World di Flask

3.3.1 Instalasi Python 2.7.

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

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



Gambar 3.1 Download Softfile Python 2.7.

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

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

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

Listing 3.1 Contoh kode program hello.py

```

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

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

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.

