

Penggunaan Python Web Framework Flask Untuk Pemula

Rahadian Irsyad

Laboratorium Telematika, Sekolah Teknik Elektro & Informatika,
Institut Teknologi Bandung, e-mail: rahadian.irsyad@gmail.com

Abstraksi— *Paper* ini memberikan gambaran umum tentang penggunaan Python *Web Framework* bernama Flask untuk pemula. Flask merupakan sebuah *web microframework* berlisensi BSD, dimana Flask memiliki fleksibilitas tinggi dalam penggunaannya karena memiliki *core* yang sederhana. Flask sangat cocok digunakan bagi pemula dalam *web development* karena tingkat kompleksitas yang lebih rendah dibanding *web framework* lain.

Keywords— *Web Framework*, Python, Flask, *Web Development*.

I. Pendahuluan

Pada saat ini, teknologi, terutama internet, telah berkembang sangat pesat. Perkembangan yang pesat ini memunculkan peningkatan yang sangat signifikan dari jumlah pengguna internet. Dari perkembangan teknologi dan peningkatan jumlah pengguna internet, timbul kebutuhan-kebutuhan dan keahlian-keahlian tentang internet, salah satunya adalah *web development*.

Web development adalah sebuah kegiatan pengembangan halaman-halaman *web* yang ada di internet. Halaman *web* adalah barang utama yang ada di internet, dimana seluruh pengguna internet pasti akan mengakses suatu *web* tertentu yang ada di internet. Oleh sebab itu, kebutuhan akan *web* di internet akan terus bertambah dan ini menimbulkan keahlian tentang *web development* pun semakin dibutuhkan.

Dengan perkembangan teknologi saat ini, terdapat banyak alat untuk membantu kegiatan *web development*, salah satunya adalah *web framework* yang dibuat menggunakan bahasa Python, yaitu Flask. Flask adalah sebuah alat yang membantu membuat kerangka untuk sebuah *web* dan dengan menggunakan Flask, pengembang pemula pun dapat menciptakan sebuah *web* yang bagus.

Pada *paper* ini akan dijelaskan gambaran umum Flask sebagai sebuah *web*

microframework serta alasan-alasan Flask cocok sebagai alat untuk *web development* bagi *developer* pemula.

II. *Web Microframework* Flask

A. *Flask*

Flask adalah sebuah *web framework* yang ditulis dengan bahasa Python dan tergolong sebagai jenis *microframework* [1]. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu *web*. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah *web* yang terstruktur dan dapat mengatur *behaviour* suatu *web* dengan lebih mudah.

Flask termasuk pada jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponen umum seperti validasi form, *database*, dan sebagainya tidak terpasang secara *default* di Flask [2]. Hal ini dikarenakan fungsi dan komponen-komponen tersebut sudah disediakan oleh pihak ketiga dan Flask dapat menggunakan ekstensi yang membuat fitur dan komponen-komponen tersebut seakan diimplementasikan oleh Flask sendiri.

Selain itu, meskipun Flask disebut sebagai *microframework*, bukan berarti Flask mempunyai kekurangan dalam hal fungsionalitas. *Microframework* disini berarti bahwa Flask bermaksud untuk membuat *core* dari aplikasi ini sesederhana mungkin tapi tetap dapat dengan mudah ditambahkan [2]. Dengan begitu, fleksibilitas serta skalabilitas dari Flask dapat dikatakan cukup tinggi dibandingkan dengan *framework* lainnya.

B. *Instalasi Flask*

Web framework Flask ditulis menggunakan bahasa Python, sehingga sebelum Flask dapat digunakan, maka *developer* harus menginstall Python pada perangkat yang akan digunakan. Oleh sebab itu, *web developer* yang akan

menggunakan Flask sebagai *web framework* untuk *web development* harus setidaknya mempelajari bahasa pemrograman Python terlebih dahulu, sebelum dapat menggunakan Flask seutuhnya.

Dalam melakukan instalasi Flask pada sebuah perangkat, dibutuhkan PIP yang biasanya sudah terinstall pada Python versi 3.4 ke atas. PIP adalah sebuah *package management system* yang biasa digunakan untuk mengatur dan menginstall *package* yang berisi modul-modul Python [3]. PIP digunakan untuk menginstall Flask karena Flask ditulis dan dikembangkan dengan bahasa dan modul-modul pemrograman Python. Dengan menggunakan PIP, semua hal yang dibutuhkan untuk instalasi Flask akan diunduh dan dipasang dalam satu perintah.

Sebelum melakukan instalasi, pengguna Flask sebaiknya menggunakan *virtual environment* dalam proses pengembangan *web*. *Virtual environment* adalah sebuah sistem untuk mengisolasi sebuah proyek Python agar tidak terjadi konflik dengan proyek Python lainnya [4]. Dengan *virtual environment*, sebuah proyek Python—atau Flask pada kasus ini, seakan ada pada satu komputer sendiri dengan dependensi sendiri dan tidak berhubungan dengan proyek Python lainnya. Untuk menginstall *virtual environment* masukkan perintah ini di *shell*.

```
$ pip install virtualenv
```

Setelah menginstall *virtual environment* di komputer, selanjutnya masuk ke direktori yang akan dijadikan tempat proyek Python berada. Setelah itu, buat dan aktifkan *virtual environment* disana dengan memasukkan perintah, sesuai dengan versi Python yang digunakan.

```
# python 2:
$ virtualenv virenv
# python 3:
$ python3 -m venv virenv
```

Perintah di atas adalah perintah untuk membuat sebuah *virtual environment* dengan nama *virenv*. Untuk mengaktifkan *virtual environment* yang telah dibuat di direktori tujuan, masukkan perintah ini.

```
$ source virenv/bin/activate
```

Setelah mengaktifkan *virtual environment*, *shell* pada komputer akan berubah menjadi seperti ini.

```
(virenv)$
```

Hal ini menandakan bahwa *virtual environment* telah aktif dan proyek Python yang akan dibuat telah terisolasi dan mempunyai sumber dependensi tersendiri.

Setelah mengaktifkan *virtual environment*, Flask dapat diinstall tanpa perlu mengkhawatirkan adanya konflik dependensi dengan proyek Python lainnya. Masukkan perintah ini setelah masuk dan mengaktifkan *virtual environment* untuk menginstall Flask.

```
$ pip install Flask
```

Untuk memeriksa keberhasilan instalasi Flask, buat sebuah file berisi kode Python berikut dan beri nama file tersebut 'hello.py'.

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```

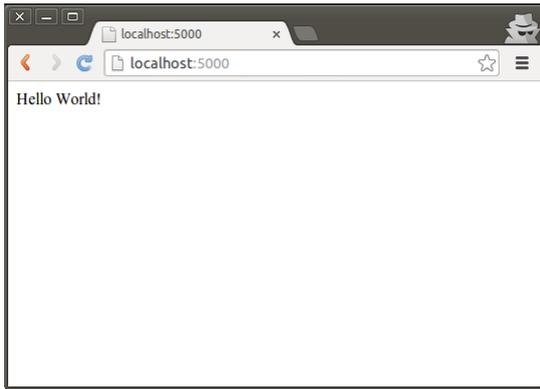
Selanjutnya, jalankan file tersebut menggunakan Flask dengan memasukkan perintah berikut.

```
$ FLASK_APP=hello.py flask
run
```

Setelah itu akan muncul pemberitahuan bahwa Flask telah berjalan di *localhost*.

```
* Running on
http://localhost:5000/
```

Selanjutnya, buka aplikasi *browser* di komputer dan akses IP 127.0.0.1 atau *localhost* di komputer tersebut. Maka akan muncul sebuah halaman seperti ini.



Gambar 1. Halaman hasil hello.py

C. Fitur-fitur Flask

Seperti yang telah dijelaskan di poin A, Flask merupakan *microframework* yang berarti Flask mempunyai *core* yang sangat sederhana dan kecil, namun tetap dapat ditambah dan tumbuh. Berdasarkan hal tersebut, fitur-fitur bawaan dari Flask sendiri termasuk sedikit jumlahnya, diantaranya adalah

- *Built-in development server.*
- *Debugger* cepat.
- *Integrated support* untuk pengetesan unit.
- Kompatibel dengan mesin aplikasi Google.
- *RESTful request dispatching.*
- *Jinja2 templating.*
- Mendukung *secure cookies.*
- Berbasis *unicode.*
- Mengikuti *WSGI 1.0*

Selain itu, Flask juga didukung dengan dokumentasi yang sangat baik dan banyak forum yang ada di internet untuk mendiskusikan masalah terkait Flask [1].

D. Cara kerja Flask

Setelah mengenal Flask dan menginstall Flask pada poin-poin sebelumnya, poin ini akan menjelaskan bagaimana cara kerja Flask secara umum. Cara kerja Flask dapat dijelaskan melalui kode file bernama 'hey.py' berikut.

```

1. from flask import Flask
2.
3. app = Flask(__name__)
4.
5. @app.route('/')
6. def home():
7.     return "Hey there!"
  
```

```

8.
9. if __name__ == '__main__':
10.     app.run(debug=True)
  
```

Pada baris ke-1, kode tersebut mengambil kode-kode dasar yang dibutuhkan dalam pembuatan proyek. `flask` disini merupakan kerangka kerja dan `Flask` adalah tipe data *class* Python. Dengan kata lain, `Flask` adalah *prototype* yang digunakan untuk membuat contoh aplikasi *web*. Selain itu, dengan menggunakan kode yang serupa dengan kode di baris-1, `Flask` dapat menggunakan berbagai fitur yang disediakan oleh pihak ketiga, seperti validasi form dengan `WTForms` dan sebagainya. Caranya cukup dengan mengunduh *package* dan *module* dari pihak ketiga tersebut dan menuliskan kode seperti ini.

```

from flask_wtf import
FlaskForm

from wtforms import
StringField

from wtforms.validators import
DataRequired
  
```

Pada baris ke-3, kode tersebut membuat sebuah *instance* dari *class* `Flask` dengan nama `app`. Variabel `__name__` adalah sebuah variabel khusus yang akan mengambil *string* dari `'__main__'`.

Pada baris ke-5 hingga 7, kode tersebut mendefinisikan sebuah fungsi dari aplikasi *web* ini. Kode `@app.route('/')` memetakan ke URL *home* atau (`localhost:5000/`) dan kode `def home():` mendefinisikan sebuah fungsi bernama `home` yang akan dipanggil ketika halaman *home* atau `'/'` diakses. Sehingga ketika halaman *home* atau `'/'` diakses akan mengembalikan sebuah *string* "Hey there!" pada halaman tersebut.

Baris ke-5 hingga 7 adalah poin utama dari `Flask`. Ketika rute masukan diganti menjadi halaman lain, contohnya `'/profile'`, maka fungsi yang dijalankan adalah fungsi yang sudah disiapkan ketika *web* diakses dengan alamat `'localhost:5000/profile'`. Seperti pada kode contoh berikut

```

from flask import Flask
  
```

```

app = Flask(__name__)

@app.route('/')
def home():
    return "Hey there!"

@app.route('/profile')
def profile():
    return profile_one

```

Selanjutnya, pada baris ke-9 dan 10, ini adalah sebuah teknik yang digunakan para *developer* yang menggunakan bahasa pemrograman Python. Pada Python, *script* yang sedang dijalankan akan diberikan nama `'__main__'` dan jika *script* diambil dari *script* lain, maka *script* tersebut tetap menggunakan nama aslinya, contohnya `'hello.py'`. Pada kasus ini, Flask menjalankan *script* ini, sehingga variabel `__name__` mempunyai nilai `'__main__'`. Dengan begitu, kode pada baris ke-9 terpenuhi dan aplikasi *web* akan dijalankan dengan kode `app.run(debug=True)`. Parameter pada *method* `app.run(debug=True)` berarti Python akan mencetak kesalahan-kesalahan yang mungkin terjadi pada halaman *web* tersebut[5].

III. Keunggulan Flask

Berdasarkan penjelasan tentang apa itu Flask, cara menginstall Flask, fitur-fitur yang ditawarkan oleh Flask, dan cara kerja Flask, Flask memiliki keunggulan serta kekurangan dibandingkan dengan *web framework* berbasis Python lainnya, seperti Django, CherryPy, dan sebagainya. Keunggulan-keunggulan dari Flask adalah

- Ringan untuk dijalankan karena mempunyai *core* yang sederhana dan desain modular.
- Dapat menangani fungsi HTTP *request* dengan mudah.
- API yang baik dan koheren.
- Dokumentasi yang banyak dan terstruktur dengan baik, penuh dengan contoh yang dapat digunakan langsung.
- Mudah untuk dipasang dan di-*deploy* untuk produksi.
- Mudah untuk diperiksa secara menyeluruh. (*Unit testability*.)

- Fleksibilitas tinggi, dengan konfigurasi yang sangat mudah diubah.

Selain memiliki keunggulan-keunggulan yang telah disebutkan, Flask juga memiliki kekurangan, diantaranya adalah.

- Tidak memiliki ORM dan *database layer* bawaan, sehingga harus menggunakan aplikasi dari pihak ketiga.
- Tidak *async-friendly*.

Karakteristik Flask yang sangat membutuhkan aplikasi pihak ketiga dalam menjalankan fitur-fiturnya menjadi pedang bermata dua yang bisa menjadi keunggulan maupun kekurangan dari Flask itu sendiri.

IV. Kesimpulan

Setelah membahas berbagai hal tentang Flask, mulai dari Flask sebagai *web microframework*, proses installasi Flask, fitur-fitur yang ditawarkan, serta cara kerja dan sedikit contoh kode Flask, dapat dikatakan bahwa Flask sangat cocok digunakan oleh *developer* pemula yang baru saja mempelajari *web development*. Hal ini didasari dengan kemudahan dalam hal mempelajari *framework* Flask dengan banyaknya dokumentasi dan contoh yang ada. Selain itu, kemudahan dalam proses installasi Flask menjadikannya pilihan untuk para *web developer* pemula. Serta, sebagai *microframework*, Flask cukup ringan untuk dijalankan tanpa perlu komputer dengan spesifikasi tinggi ataupun komponen dan *library* khusus, sehingga *developer* yang masih “coba-coba” dapat dengan mudah mencoba Flask.

Daftar Pustaka

- [1] <http://flask.pocoo.org>
(Diakses pada 16/12/2018 14:12)
- [2] <http://flask.pocoo.org:80/docs/1.0/foreword>
(Diakses pada 16/12/2018 15:33)
- [3] https://www.w3schools.com/python/python_pi_p.asp
(Diakses pada 16/12/2018 15:47)
- [4] <https://docs.python-guide.org/dev/virtualenvs/>
(Diakses pada 16/12/2018 16:18)
- [5] <https://pythonhow.com/how-a-flask-app-works/>
(Diakses pada 16/12/2018 18.20)