



Web Service Framework: flask dan fastAPI

Rendi Yusuf Azhari
Fakultas Sains dan Teknologi, Universitas Putra Bangsa
email: ligerrendy@gmail.com

ABSTRAK

Interoperabilitas merupakan interaksi antar komputer yang berlangsung didalam jaringan internet. Cara untuk mencapai interoperabilitas dengan cara membuat web service. Web service akan menerima *request* dan membalas dengan *response*. Banyak framework web service yang beredar saat ini, contohnya adalah flask dan fastAPI. Keduanya mempunyai struktur program yang sama, namun berbeda tujuan. Flask dirancang untuk web secara *general purpose*, namun fastAPI dirancang hanya sebagai *web service*.

Kata Kunci: Interoperabilitas, *web service*, *web service framework*, flask, fastAPI

ABSTRACT

Interoperability is the interaction between computers that take place on the internet network. The way to achieve interoperability is by creating a web service. The web service will accept the request and reply with a response. There are many web service frameworks currently available, for example flask and fastAPI. Both have the same program structure, but different goals. Flask is designed for general web purpose, but fastAPI is designed only as a web service.

Keywords: *Interoperability*, *web service*, *web service framework*, flask, fastAPI

PENDAHULUAN

Pengguna internet di Indonesia menurut APJII pada tahun 2019 hingga 2020 sebanyak 196.71 juta, sedangkan total penduduk di Indonesia sebesar 266,91 juta (Indonesia Survey Center, 2020). Dari data APJII menghimpun pertumbuhan pengguna data internet dari tahun 2019 hingga 2020 sebesar 64,8% (Indonesia Survey Center, 2020) Jumlah pengguna yang terkoneksi dari *smartphone* dengan nilai 95,4%

(Indonesia Survey Center, 2020). Konten yang banyak diakses berupa media sosial, hiburan, berita hingga berbelanja secara online (Indonesia Survey Center, 2020).

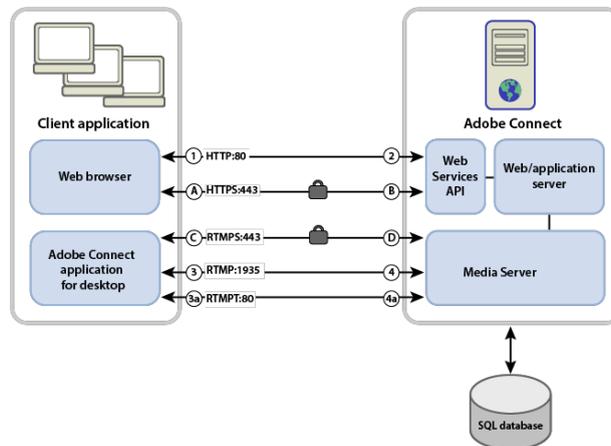
Terdapat banyak proses yang terjadi ketika kita mengakses internet seperti proses keamanan, proses pertukaran data, dan proses lainnya (Saputra, dkk 2018) (Husni,2020). Proses pertukaran data di dalam internet antara server dengan client disebut dengan API (*Application Program Interface*) (Saputra, dkk 2018) (Husni,2020). API atau sering disebut dengan *web service*, merupakan program yang berfungsi sebagai interoperabilitas (Husni,2020) . Interoperabilitas merupakan interaksi yang terjadi antar sistem (Husni,2020). Interoperabilitas memungkinkan sistem bertukar data, baik data yang tersimpan pada basis data atau proses data lainnya (Kominfo ,2020).



Gambar 1 Proses interoperabilitas sederhana

Pada gambar 1 merupakan proses interoperabilitas sederhana antara aplikasi ponsel pintar dengan API (Kominfo ,2020) (Husni,2020). Proses yang terjadi adalah ponsel pintar mengirim *request* dan diterima API. API kemudian memproses *request* dengan membalas dengan *response*. *Request* bisa berupa pencarian data, *request* permintaan data, *request login*, dan lain sebagainya. *Response* balasan dari *request* yang dikirim oleh *client*. API atau sering disebut *web service* berada di server atau sistem *cloud*. Selain membutuhkan server, API juga membutuhkan protokol dan arsitektur.

Mesin sama dengan manusia, agar dapat saling mengerti dibutuhkan tata cara berbicara. Tata cara berbicara pada sistem komputer disebut dengan protokol. Protokol berfungsi mengatur dua mesin dapat saling berbicara. Protokol yang sering digunakan dalam interoperabilitas merupakan TCP/IP. TCP/IP merupakan merupakan standar komunikasi antar komputer ketika menggunakan internet. Pada protokol TCP/IP sudah mengatur tentang *routing*, *name server*, pengalamatan, dan isi pesan. *Web service* menggunakan internet sebagai jalan utama agar dapat berkomunikasi, dapat disimpulkan *Web service* menggunakan TCP/IP.



Gamabr 2 Arsitektur *Web Service*

Ketika beberapa komputer saling terhubung di internet, maka terdapat arsitektur jaringan komputer. Arsitektur jaringan komputer merupakan desain rancangan dari beberapa komputer yang saling terhubung. Dalam proses desain arsitektur web service biasanya terdapat web sarana penerima *request* dan *response*, basis data untuk menyimpan data. Pada perancangan arsitektur *web service*, terdapat beberapa arsitektur yang dapat digunakan, seperti RESTFUL dan SOAP.

(Simple Object Access Protocol) adalah sebuah XML-based mark-up language untuk pergantian pesan diantara aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data object di dalam network. SOAP mendefinisikan empat aspek dalam komunikasi: Message envelope, Encoding, RPC call convention, dan bagaimana menyatukan sebuah message di dalam protokol transport (Gilvy dkk, 2019). Sebuah SOAP message terdiri dari SOAP Envelope dan bisa terdiri dari attachments atau tidak memiliki attachment (Gilvy dkk, 2019). SOAP envelope tersusun dari SOAP header dan SOAP body, sedangkan SOAP attachment membolehkan non-XML data untuk dimasukkan kedalam SOAP message, di-encoded, dan diletakkan ke dalam SOAP message dengan menggunakan MIMEmultipart (Gilvy dkk, 2019). HTTP berbasis API berarti API yang diekspos sebagai salah satu atau lebih HTTP URI dan respon berupa XML/JSON (Gilvy dkk, 2019). Skema respon dapat dikustomisasi untuk setiap objek pada XML/JSON (Gilvy dkk, 2019).



Gambar 3 Struktur Data JSON (kiri) Struktur XML (kanan)



Dalam interoperabilitas terdapat *request* dan *response*. Struktur data yang digunakan bisa berupa XML (Extensible Markup Language) atau JSON (JavaScript Object Notation). Perbedaan dari XML dan JSON adalah penulisan sintaks. XML penulisan seperti pada Gambar 3 (sebelah kanan), dengan setiap objek diawali dengan `<nama_obyek>` dan diakhiri dengan `</nama_obyek>`. Pada XML nilai berada diantara tanda `<objek>nilai</objek>` . Sedangkan untuk JSON dengan objek : nilai.

METODE

Penelitian ini menggunakan metode studi literatur. Bahan dan ulasan yang akan dikaji berupa perbandingan antara dua framework API, yaitu flask dan fastAPI. beberapa hal yang akan dijadikan bahan perbandingan adalah sebagai berikut :

1. Struktur pemrograman antar flask dan fastAPI
2. Kebergunaan antar flask dan fastAPI
3. Kemudahan dalam pemrograman antar flask dan fastAPI

HASIL DAN PEMBAHASAN

Flask adalah sebuah web framework yang ditulis dengan bahasa Python dan tergolong sebagai jenis microframework. 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 fungsidan komponen umum seperti validasi form, database, dan sebagainya tidak terpasang secara default di Flask. 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. Dengan begitu, fleksibilitas serta skalabilitas dari Flask dapat dikatakan cukup tinggi dibandingkan dengan framework lainnya.

```
14 lines (10 sloc) | 253 Bytes
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route('/')
5  def index():
6      return "<h1>Hello, World!</h1>"
7
8  @app.route('/user/<name>')
9  def user(name):
10     return '<h1>Hello, {0}!</h1>'.format(name)
11
12 if __name__ == '__main__':
13     app.run(debug=True)
```

Gambar 4 Struktur penulisan flask

Struktur Penulisan flask memang sederhana. Struktur dasar dari flask berupa routing dan diikuti dengan fungsi program. Routing dalam flask bertujuan untuk memberikan nama *endpoint* atau URL, contoh `@app.route(/user)` berarti domain/user. Python merupakan bahasa yang dapat mengadopsi fungsi, dan *class*, yang akhirnya beberapa fungsi dapat dipisahkan. Ketika Program berskala besar ditulis maka struktur pemrograman pada gambar 4 akan memanjang hingga kebawah, yang akhirnya tidak akan efisien untuk melakukan perbaikan program. Masalah tersebut dapat diatasi gambar 5.

```
~/LargeApp
|-- run.py
|-- config.py
|_ /env          # Virtual Environment
|_ /app         # Our Application Module
  |-- __init__.py
  |-- /module_one
        |-- __init__.py
        |-- controllers.py
        |-- models.py
  |_ /templates
        |_ /module_one
              |-- hello.html
  |_ /static
  |_ ..
  |_ .
|_ ..
```

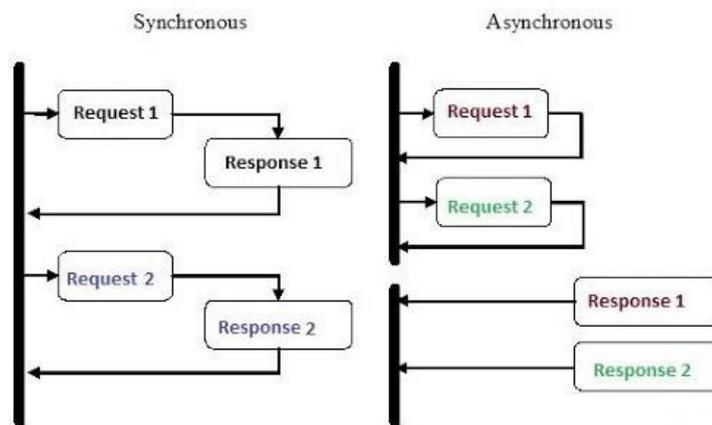
Gambar 5 Struktur flask ketika menangani proyek besar

Namun gambar 5 bukan merupakan struktur absolut, yang berarti struktur tersebut dapat diubah sesuai dengan kebutuhan developer. Pada gambar 5 menerangkan adanya struktur dengan model, controller, template, dan static. Model berisikan query atau fungsi program, controller berisikan routing, template berisikan file javascript dan html. Contoh lain struktur pemrograman flask pada gambar 6. Controller, model, dan template dapat berupa folder terpisah. Di dalam folder tersebut berisikan untuk controller berisikan tentang router, model bisa berisikan fungsi atau query, templates dapat berisikan file javascript dan html.

Flask merupakan *micro framework*, yang dapat diartikan beberapa modul dapat diunduh secara terpisah, bahkan beberapa pihak dapat membuat sendiri modul yang terkait. Flask pada dasarnya merupakan framework web, yang dapat dijadikan sebagai *web service*, atau web dengan body html.

```
~/name_project
|--app
|  |--__init__.py
|  |--model
|     |--name_model.py
|  |--controller
|     |--name_controller.py
|--template
|--static
```

Gambar 6 Alternatif penulisan struktur program flask



Gambar 7 Proses Asinkron

Ketika flask dibuat dengan keperluan secara global atau general, maka FastAPI merupakan framework yang dikhususkan untuk pembuatan API. FastAPI dibuat oleh @tiangolo (Sebastián Ramírez) dari kolombia. FastAPI mengusung konsep asinkron API. asinkron memungkinkan program tidak harus menunggu request yang datang dan diproses.

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
async def root():
    return {"message": "Hello World"}
```

Gambar 8 Struktur program fastAPI

FastAPI mempunyai struktur yang sama dengan flask, menggunakan decorator routing. Struktur flaskAPI ketika memprogram proyek besar juga dapat ditulis sebagai berikut

Tabel 1 Perbedaan Flask dan fastAPI

	flask	fastAPI
Kebergunaan	flask dapat difungsikan sebagai web, web service	hanya sebagai web servis
Struktur	menggunakan decorator router, dan bisa menggunakan konsep model view template	tidak mempunyai struktur template, decorator router mirip flask
Penyimpanan sementara	support cookies, sessions	-
Keamanan	bisa menyimpan di sessions, support token	hanya dapat menerima keamanan token
kompatibilitas	luas, karena microframework	terbatas, hanya dapat menggunakan modul eksternal
asinkronus	support	support

SIMPULAN

Dalam pemilihan arsitektur interoperabilitas sangat penting. Baik dari segi fitur, keamanan, kompatibilitas, hingga struktur program. FastAPI didesain mirip dengan flask, dengan harapan ketika programmer ingin pindah framework dapat langsung dengan mudah memahami. Dalam proyek yang besar sebaiknya program dipisahkan menjadi beberapa bagian untuk memudahkan dalam penulisan.

Harapan kedepannya akan diuji dan dibandingkan antara flask dengan fastAPI. Perbandingan akan dilakukan dari sisi keamanan data, waktu *response*, hingga penggunaan sumber daya pada komputer.

REFERENSI

Melwin Syafrizal, & others (2020). Pengantar jaringan komputer. Penerbit Andi.

Era Susanti (2020). Implementasi RESTful API dalam Pembuatan Master Data Planogram Menggunakan Framework Flask (Studi Kasus: PT Sumber Alfaria Trijaya, Tbk). *Techno. Com*, 19, 295-307.

Michael Stonebraker, & Lawrence A Rowe (1983). *Data Base Portals: A New Application Program Interface*. Computer Science Division, University of California.

Danandjaya Saputra (2018). Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan. *Jurnal Bangkit Indonesia*, 7, 17.

Robby Reza, Agung Nugroho Jati, & Umar Ali Ahmad (2016). Perancangan Dan Implementasi Penggunaan Cloud Server Paas (platform As A Service) Sebagai Penyimpan Data Pengunjung Dan Rekaman Video Pengunjung Pada Building Security System Berbasis Embedded. *eProceedings of Engineering*, 3.

Beni Adi Pranata, Astria Hijriani, & Akmal Junaidi (2018). Perancangan Application Programming Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit. *Jurnal Komputasi*, 6, 33-42.

Jon Postel, & others (1981). Internet protocol.

Husni, Ika Oktavia Suzanti, Yoga Dwitya Pramudita, Sigit Susanto Putro, & Lukman Heryawan (2020). Web Service for Search Engine Bahasa Indonesia (SEBI). *Journal of Physics: Conference Series*, 1569, 22087.