

Aplikasi Flask Server Search Engine

Daftar Isi

Pendahuluan	3
Source Code	3
Instalasi Aplikasi	3
Petunjuk Penggunaan Sistem	4
URL Aplikasi & API Endpoint	4
API Crawler	4
API Perankingan	5
Penggunaan API	5

Pendahuluan

Aplikasi Flask Server Search Engine ini merupakan web service yang akan mengelola data pada search engine. Web service ini berbentuk REST API yang dapat diakses melalui request http pada device yang terhubung dengan internet. Pihak-pihak yang akan menerima manfaat dari penerapan aplikasi ini antara lain:

1. Pengembang Aplikasi Front-End: Mendapatkan akses yang mudah terhadap data yang dapat diakses melalui request API

Untuk dapat digunakan, Aplikasi Flask Server Search Engine perlu diinstalasi pada suatu web hosting yang memiliki dukungan untuk menjalankan sistem basis data MySQL.

Source Code

Sumber kode program aplikasi Aplikasi Flask Server Search Engine dapat diunduh melalui <https://github.com/lazuardyk/search-engine> . Kode program tidak dilampirkan di manual book ini.

Instalasi Aplikasi

Terdapat beberapa tahapan instalasi yaitu:

1. Mengunduh Source Code

Mengunduh source code dilakukan dengan cara clone repositori menggunakan git. Buat direktori baru, inisiasi git, dan clone repo.

```
$ mkdir search-engine  
$ git init  
$ git clone https://github.com/lazuardyk/search-engine
```

2. Menginstall Package yang dibutuhkan

Pada folder yang search engine yang telah diunduh, jalankan perintah di bawah :

```
$ pip install -r requirements.txt
```

3. Konfigurasi httpd.conf pada server

Konfigurasi httpd.conf diperlukan agar server dapat mengenali aplikasi yang nantinya ada dijalankan menggunakan versi python tertentu

```

WSGIDaemonProcess /search-engine python-path=/opt/rh/rh-python38/root/lib/pyt$
WSGIProcessGroup /search-engine
WSGIApplicationGroup %{GLOBAL}
WSGIScriptAlias /search-engine /var/www/html/search-engine/search-engine.wsgi
WSGIScriptReloading on
<Directory "/var/www/html/search-engine/src">
    AllowOverride All
    Options +ExecCGI
    AddHandler cgi-script .cgi .pl .py
    Order allow,deny
    allow from all
</Directory>

```

Petunjuk Penggunaan Sistem

Sistem ini sementara hanya mendukung satu user, dapat dikatakan apapun request dan perubahan data yang dilakukan dianggap dilakukan oleh satu user. Role user tersebut adalah pengelola search engine.

URL Aplikasi & API Endpoint

Sistem telah terpasang pada alamat <http://jft.web.id/search-engine>. Sistem ini telah disesuaikan untuk kebutuhan aplikasi front-end search engine.

API Endpoint merupakan sub url untuk melakukan request kepada sistem. Dalam dokumentasi ini setiap API Endpoint dibagi berdasarkan module atau fungsi seperti di bawah ini:

API Crawler

CREATE	/api/v1.0/crawling/start?duration={duration}	GET	menjalankan proses crawling
UPDATE	/api/v1.0/crawling/stop	GET	memberhentikan proses crawling
READ	/api/v1.0/crawling/pages?start={start_index}&length={length}	GET	mendapatkan data halaman yang telah dicrawl
CREATE	/api/v1.0/crawling/start_insert	POST	mendapatkan id crawl untuk memulai penambahan halaman crawler melalui API

CREATE	/api/v1.0/crawling/insert_page	POST	menambahkan data halaman yang dicrawl
READ	/api/v1.0/crawling/page_information	POST	mendapatkan detail halaman

API Perankingan

READ	/api/v1.0/document_ranking/tf_idf?keyword={keyword}&start={start_index}&length={length}	GET	mendapatkan ranking pembobotan TF IDF
READ	/api/v1.0/page_ranking/page_rank?start={start_index}&length={length}	GET	mendapatkan ranking PageRank
READ	/api/v1.0/overall_ranking/similarity?keyword={keyword}&sort={sort_by}&start={start_index}&length={length}	POST	mendapatkan ranking secara keseluruhan

Penggunaan API

API Sistem ini dibuat menggunakan arsitektur REST (REpresentational State Transfer), dimana bentuk data yang akan digunakan untuk komunikasi adalah JSON.

Berikut cara penggunaan REST API berdasarkan metodenya:

1. GET

Metode GET adalah metode yang datanya dikirim melalui URL..

Berikut merupakan contoh penggunaan request GET (contoh menggunakan python):

```
import requests

url = "http://jft.web.id/search-engine/api/v1.0/page_ranking/page_rank?start=0&length=2"
payload={}
headers = {}
response = requests.request("GET", url, headers=headers, data=payload)

print(response.text)
```

Dari request diatas akan menghasilkan response berbentuk json seperti dibawah:

```

{
  "data": [
    {
      "id_pagerank": 30,
      "page_id": 30,
      "pagerank_score": 0.0013682586525913308,
      "url": "https://adsmart.detik.com"
    },
    {
      "id_pagerank": 40,
      "page_id": 40,
      "pagerank_score": 0.001364460785806049,
      "url": "https://www.detik.com"
    }
  ],
  "message": "Sukses",
  "ok": true
}

```

2. POST

Metode POST digunakan agar data tidak ditampilkan di URL, biasanya untuk data yang sensitif. Maka dari itu penggunaan POST mengharuskan memasukan data yang akan dikirimkan dalam melakukan request.

Berikut merupakan contoh penggunaan request POST (contoh menggunakan python):

```

import requests

url = "http://jft.web.id/search-engine/api/v1.0/crawling/page_information"

payload={'id_pages': [3,4]}
files=[]
headers = {}

response = requests.request("POST", url, headers=headers, data=payload, files=files)

print(response.text)

```

Dalam melakukan request, POST memerlukan data tambahan yang biasa disebut payload. Untuk detail payload yang diperlukan pada search engine ini dapat dilihat pada dokumentasi di <https://documenter.getpostman.com/view/11687432/2s8YerLWtg>

Response yang akan diberikan pada request di atas adalah:

```
{
  "data": [
    {
      "content_text": "Klasemen Piala Dunia 2022: Maroko Gusur Belgia dari Puncak Grup  
F Sepakbola | 43 menit yang lalu,",
      "crawl_id": 1,
      "created_at": "2022-11-27 23:01:46",
      "description": "Rangkuman Berita Terpopuler dan Terbaru Hari Ini di Kanal Detikcom - detikcom",
      "duration_crawl": "0:00:00",
      "hot_url": 0,
      "html5": 1,
      "id_page": 3,
      "keywords": "Berita Terpopuler Hari Ini Detikcom, berita terbaru hari ini, berita terhangat",
      "model_crawl": "BFS crawling",
      "size_bytes": 125933,
      "title": "Berita Terpopuler dan Terbaru Hari Ini di Kanal Detikcom - de",
      "url": "https://www.detik.com/terpopuler"
    },
    {
      "content_text": "detikNetwork adalah bagian dari CT Corp. Satu akun MPC untuk be  
raktivitas di semua layanan CT Corp dan Allo Bank. Pelajari lebih lanjut. BUAT AKUN MPC LOGI  
N Klik di sini,",
      "crawl_id": 1,
      "created_at": "2022-11-27 23:01:52",
      "description": "-",
      "duration_crawl": "0:00:05",
      "hot_url": 0,
      "html5": 0,
      "id_page": 4,
      "keywords": "-",
      "model_crawl": "BFS crawling",
      "size_bytes": 3196,
      "title": "Detikconnect - Sign In",
      "url": "https://news.detik.com/kolom/kirim"
    }
  ],
  "message": "Sukses",
  "ok": true
}
```