Penerapan Breadth First Search untuk Web Scraper dan Search Engine pada Situs IMDB

Bambang Haryo Pramdio Bagus Anggito 13518080
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
bambanganggit@gmail.com 13518080@std.stei.itb.ac.id

***Abstract*—Web Scraper adalah alat untuk mengekstrak data dari sebuah halaman page. Penggunaan Breadth First Search pada web scraper memungkinkan web scraper dapat menjelajahi halaman web yang berhubungan dengan halaman awal web scraper dijalankan.**

Keywords—Breadth First Search, Web Scraper, Data

I. PENDAHULUAN

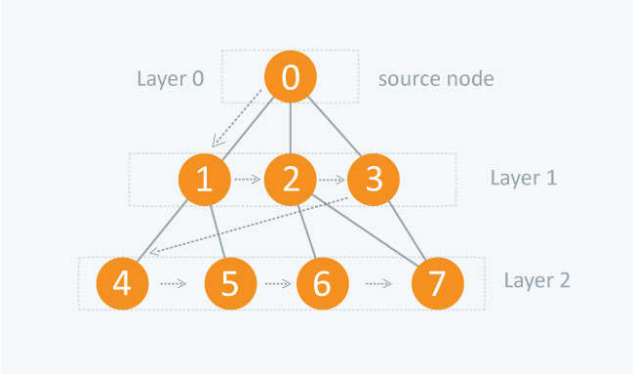
Keterhubungan komputer sekarang ini sangatlah luas. Sebuah komputer dapat mengakses berbagai macam informasi dengan hanya terhubung ke internet. Informasi-informasi yang tersebar diinternet biasanya terdapat pada website yang diakses pada browser. Untuk mengekstrak informasi yang banyak, mengakses setiap halaman website secara manual akan memerlukan waktu yang lama. Agar mempersingkat waktu ekstrak, diciptakan alat untuk mengekstrak website yaitu web scraper.

Web scraper akan mengekstrak hal-hal yang dianggap penting dari sebuah halaman. Penggunaan algoritma pencarian breadth first search akan memungkinkan web scraper menjelajahi halaman website yang terhubung dengan halaman lain dan mengekstrak data dari setiap halaman.

II. DASAR TEORI

A. *Breadth First Search*

Breadth First Search (BFS) adalah teknik untuk menjelajahi graf yang dimulai dari simpul pertama dan dilanjutkan penjelajahan kepada simpul tetangga secara berlapis hingga ditemukan simpul yang dicari. Jarak dari lapisan 2 ke simpul mulai lebih besar daripada lapisan 1 ke simpul mulai sehingga, lapisan 1 akan dijelajahi terlebih dahulu.



Gambar 1. *Breadth First Search* (Sumber: hackerearth.com)

Pada gambar 1, simpul 0 adalah simpul mulai yang akan diproses pertama kali dan jika simpul tersebut bukan simpul yang dicari maka tetangga dari simpul tersebut akan dibangkitkan dan dimasukkan ke dalam antrian simpul yang akan dibangkitkan.

Pseudocode

```
BFS (G, s):
   Q.enqueue(s)

   mark s as visited
   while (Q is not empty):
      v  =  Q.dequeue( )
      // process node v

      for all neighbours w of v in Graph G
         if w is not visited:
            Q.enqueue(w)
            mark w as visited.
```

Sebuah graf dapat memiliki sirkuit seperti pada simpul 0, 2 ,3 dan 7 pada gambar 1. Graf yang memiliki sirkuit dapat menimbulkan masalah pada saat pembangkitan simpul dan terjadi *infinite loop*. Untuk menghindari permasalahan *infinite loop*, setiap simpul diberi penanda bahwa simpul telah dikunjungi sebelumnya.

B. *Web Scraper*

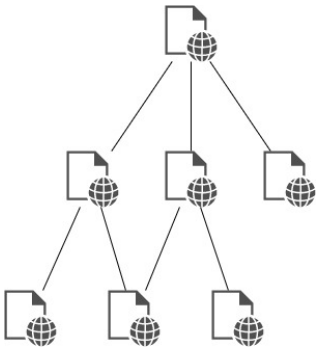
Web Scraping mengacu pada mencari dan mengekstraksi data pada website, sedangkan Web Scraper adalah agen atau alat yang digunakan untuk mencari dan mengekstraksi data pada website. Data diambil dengan memanfaatkan response dari website ketika agen melakukan request terhadap suatu website. Data dapat berupa sebuah html yang dapat diekstrak. Web Sraper biasa digunakan untuk mengumpulkan data dari sebuah website yang memiliki banyak data.

III. PEMBAHASAN

Halaman pada sebuah website biasanya terhubung dengan halaman lain menggunakan sebuah url yang disematkan di dalam html. Dengan mengekstrak url pada html, agen dapat melakukan request pada url tersebut dan mengekstrak data pada url tersebut hingga hal yang dicari ditemukan atau semua data yang diinginkan berhasil diekstrak.

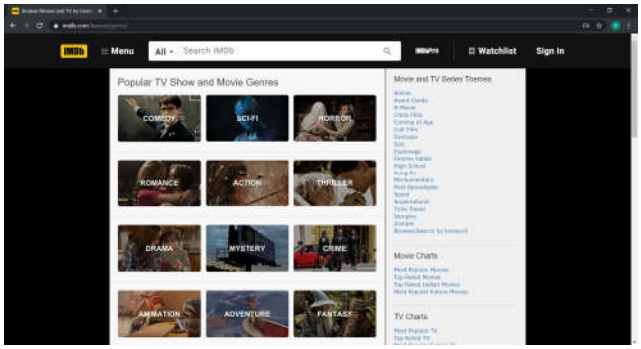
A. Model

Halaman website dapat dianggap sebagai sebuah simpul yang terhubung dengan tentangganya melalui url yang disematkan dalam html. Halaman – halaman yang saling terhubung tersebut dapat dianggap sebagai sebuah graf sehingga dapat dijelajahi menggunakan BFS.

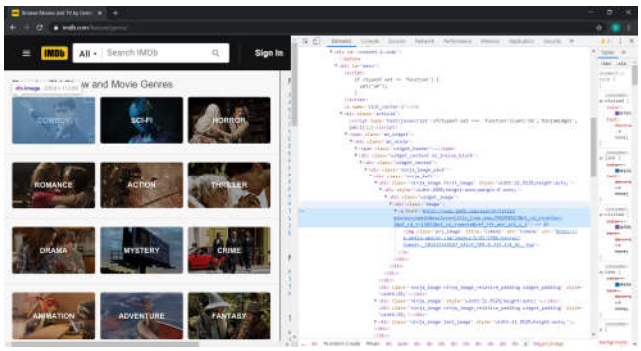


Gambar 2. Ilustrasi website sebagai graf (Sumber: koleksi pribadi)

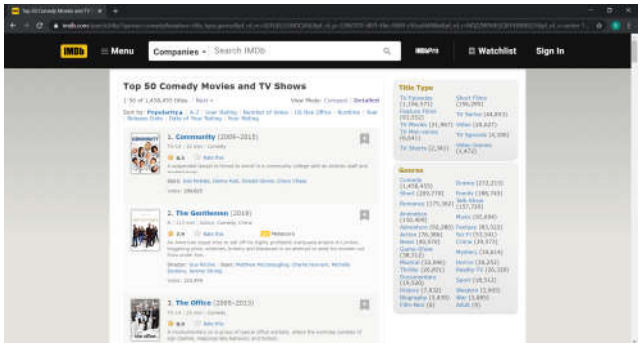
Pada situs IMDB terdapat halaman yang mengelompokkan film film berdasarkan genrenya pada https://www.imdb.com/feature/genre/



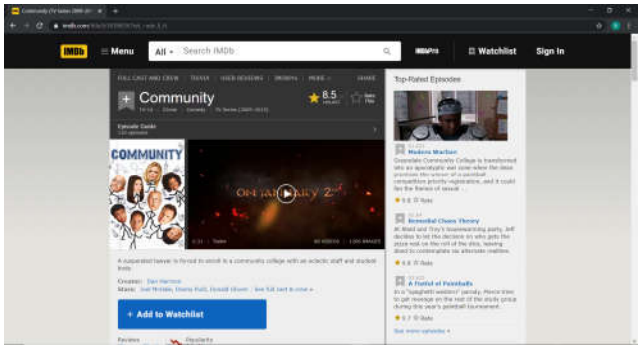
Setiap gambar pada genre disisipkan sebuah url yang dapat diekstrak untuk dan digunakan untuk menelusuri halaman web IMDB berdasarkan kategori.



Url yang disisipkan pada gambar ternyata merujuk pada halaman berisi list film yang sudah dikategorikan. Pada halaman tersebut juga terdapat url yang disematkan pada hyperlink next yang dapat diekstrak untuk menjelajahi halaman selanjutnya.



Setiap film memiliki halaman masing masing yang dapat diakses dari halaman berisi list film. Di halaman setiap film terdapat judul, rating, waktu, genre, dan tahun rilis yang akan diekstrak.



B. Implementasi

Menggunakan Python, website dapat diakses menggunakan library requests.

```
import requests

url = ‘https://www.imdb.com/feature/genre/’
response = requests.get(url)
```

Parse hasil request menggunakan library BeautifulSoup.

```
from bs4 import BeautifulSoup

html_soup = BeautifulSoup(response.text,
‘html.parser')
```

Mengambil url halaman berisi genre dan memasukan ke dalam queue.

```
genres_container = html_soup.find_all('di
v', class_ = 'image')

for container in genres_container:
    urlQ.put(container.a['href'])
```

Melakukan loop hingga queue habis dan mencari halaman selanjutnya dan dimasukkan ke dalam queue

```
while not urlQ.empty():
    url = urlQ.get()
    response_genre = requests.get(url)
    page = BeautifulSoup(response_genre.t
ext, 'html.parser')

    # proses halaman

    # pengambilan halaman selanjutnya

    next_page = page.find('a', class_ = '
lister-page-next next-page')
    if next_page != None:
        url = 'https://www.imdb.com' + ne
xt_page['href']
        urlQ.put(url)
```

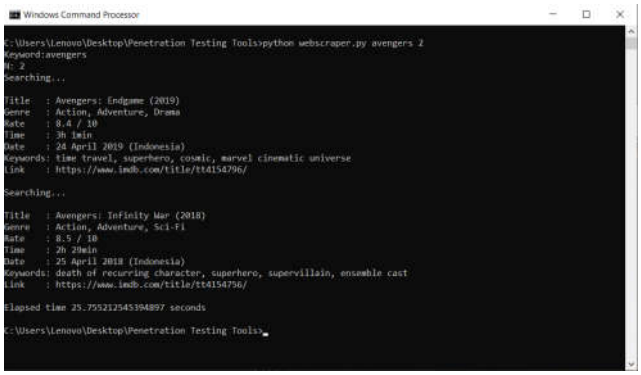
Halaman berisi list film diproses dan setiap judul film dilakukan pencocokan dengan keyword. Jika judul film dan keyword cocok, judul film dan url film akan disimpan dalam list result.

```
for item in items:
    title = item.h3.a.text
    link = 'https://www.imdb.com' + item.
h3.a['href']
    if re.search(keyword, title.lower())
and (title, link) not in result:
        result.append((title, link))
        film = getFilm(link)
        printFilm(film)
        films.append(film)
        break
    if len(result) >= N:
        break
```

Melakukan request kepada halaman film dan mengekstrak data yang penting.

```
response_film = requests.get(link)
page = BeautifulSoup(response_film.text,
'html.parser')
titlebar = page.find('div', class_ = 'tit
le_block')
if titlebar != None:
    title = titlebar.find('div', class_ =
 'title_wrapper').h1
    title = title.text.strip()
    rate = titlebar.find('span', itemprop
 = 'ratingValue')
    rate = rate.text
    bestrate = titlebar.find('span', item
prop = 'bestRating')
    bestrate = bestrate.text
    ratingCount = titlebar.find('span', i
temprop = 'ratingCount')
    ratingCount = ratingCount.text
    reviewCount = titlebar.find_all('span
', itemprop = 'reviewCount')
    subtext = subtext.text.split('|')
    subtext = [text.replace('\n','').stri
p() for text in subtext]
    time = subtext[1]
    genre = subtext[2]
    releasedate = subtext[3]
```

Hasil dari program.



IV. KESIMPULAN

Penggunaan algoritma Breadth First Search pada Web Scraper memungkinkan web scraper untuk menjelajahi banyak halaman dan mengekstrak halaman. Proses ekstrak data menjadi lebih cepat jika dibandingkan dengan mengekstrak secara manual.

VIDEO LINK AT YOUTUBE

Demo program dapat dilihat pada link berikut https://youtu.be/CLSsRHYZGEY

LINK GITHUB

Source *code* dapat dilihat pada link berikut https://github.com/bambangharyopba/IMDBFilmScraper

UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Tuhan Yang Maha Esa karena berkat-Nya penulis dapat menyelesaikan makalah ini. Penulis juga berterima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc selaku dosen K2 karena telah membimbing dan memberi penulis ilmu untuk menyelesaikan makalah ini..

REFERENSI

1. Munir, Rinaldi. "Breadth First Search (BFS) dan Depth First Search (DFS) (pdf, revisi 2020)". http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-(2020).pdf, diakses pada tanggal 2 Mei 2020
2. Munir, Rinaldi. "Pencocokan string dengan Regular Expression (Regex)".

   http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf, diakses pada tanggal 2 Mei 2020
3. Perez, Martin. "What is Web Scraping and What is it Used For?". https://www.parsehub.com/blog/what-is-web-scraping/, dikases pada tanggal 2 Mei 2020
4. Tim hackerearth. "Breadth First Search".

   https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/tutorial/, diakses pada tanggal 2 Mei 2020

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 4 Mei 2020

Bambang Haryo Pramudio Bagus Anggito / 13518080