TEKNIK SCRAPING DAN CRAWLING UNTUK MENGEKSTRAKSI REVIEW HOTEL ONLINE PADA WEBSITE TRAVELOKA (BERBASIS AJAX)

(TEKNIK SCRAPING DAN CRAWLING UNTUK MENGEKSTRAKSI REVIEW HOTEL ONLINE PADA WEBSITE TRAVELOKA (BERBASIS AJAX))

SELVI OKTARIA

ABSTRACT

The internet can be a source of public data available on various websites. The process of retrieving data from a website requires certain techniques because the data found on the website is unstructured data. Data retrieval or extraction techniques are known as scraping processes. A website also has many web pages that are interconnected so that techniques are also needed to be able to check all web pages where data will be taken. The technique for accessing linked web pages is called crawling. In the process of processing data from extraction, structured data is needed, therefore we need a scraping and crawling system that can produce structured data from a website. The use of javascript and ajax on a website makes accessing data on a website does not require an overall refresh of the web page. To do crawling on websites that use javascript and ajax, certain techniques are needed so that the crawling system can interact with ajax and the scraping process can retrieve all the data on a web page. Scraping and crawling techniques are developed using and integrating various existing technologies. The development of scraping and scrawling techniques is carried out through several stages. The stage starts from evaluating the website that will be the source of the data to get the elements where the data is. The element selection is done by using the xpath selector. All of these techniques were developed using the Python programming language. The result of developing this technique is a scraping and crawling system to extract hotel review data from the Traveloka web. The system can run steadily taking millions of hotel reviews. Review data can also be stored and displayed properly on elasticsearch.

Keyword*: scraping, crawling, scrapy, selenium, ajax, xpath, traveloka*

PENDAHULUAN

Pemerintah membutuhkan informasi tentang pariwisata untuk menentukan kebijakan-kebijakan terkait pengembangan sektor pariwisata. Salah satu parameter yang digunakan untuk membuat kebijakan adalah tingkat kunjungan wisatawan (Kementerian Pariwisata, 2016). Tingkat kunjungan wisatawan dipengaruhi oleh kepuasan wisatawan terhadap fasilitas yang ada di daerah tujuan wisata (Ni Nyoman Ayu Wiratini M, 2018).Kepuasan konsumen dapat didefinikan sebagai: "Respon konsumen terhadap evaluasi diskrepansi/ketidaksesuaian yang dirasakan antara ekspetasi sebelumnya (atau beberapa norma kinerja lain) dan kinerja aktual dari produk sebagaimana yang dirasakan setelah pengkonsumsiannya". (Wilton, 1998)

Hotel merupakan salah satu sarana akomodasi utama untuk menjadi prioritas bagi wisatawan untuk menentukan tujuan wisatanya. Tingkat kepuasan wisatawan terkait akomodasi dapat dilihat dari *review* penggunaan hotel. *Review* penggunaan hotel dapat dilihat pada *feedback* dan *review* yang diberikan oleh wisatawan setelah menggunakan fasilitas hotel terkait. Sehingga, melalui informasi *feedback* dari wisatawan bisa didapatkan parameter tingkat kepuasan wisatawan selama penggunaan fasilitas hotel. Penelitian membuktikan secara signifikan bahwa loyalitas konsumen terhadap hotel dipengaruhi oleh kepuasan konsumen (Dyah Sugandini, 2018).

Biasanya wisatawan tidak memberikan *review* secara langsung terkait *feedback* ke

sebuah hotel, melainkan menyalurkannya ke forum-forum diskusi, sosial media dan situs-situs terkait termasuk *booking online hotel*. Forum diskusi, sosial media dan *booking online hotel* tersebut banyak tersedia dalam bentuk *website*. *Feedback* atau *review* yang disampaikan melalui *website* berupa teks. Halaman sosial media sebuah *brand* telah menjadi instrumen penting yang memungkinkan konsumen berpartisipasi secara sukarela memberikan *feedback* untuk memberikan masukan ide peningkatan dan berkolaborasi dengan konsumen yang lain (Carlson, 2018).

Dari uraian diatas, *feedback* yang ada pada *website* dapat menjadi sumber data yang dibutuhkan dalam evaluasi tingkat kepuasan wisatawan. Menurut beberapa penulis, *feedback* dan *review* yang ada di *website* merupakan data tidak terstruktur (Plamen Milev, 2017). Untuk mengolah data *review* para wisatawan, dibutuhkan data yang terstruktur. Oleh karena itu dibutuhkan sebuah sistem yang dapat mengambil data dari *website* dan menghasilkan data *review* dari wisatawan yang terstruktur.

Sistem *scraping* dan *crawling* digunakan untuk mengekstrasi data dari *website* dan menghasilkan data yang terstruktur dalam model data. Penggunaan *javascript* dan *ajax* pada sebuah *website* membuat akses data pada sebuah website tidak memerlukan *refresh* keseluruhan halaman web. Data pada *website* dapat ditampilkan dengan lebih interaktif. Untuk melakukan *crawling* pada *website* yang menggukan *javascript* dan *ajax* diperlukan teknik tertentu sehingga sistem scrawling dapat berinteraksi dengan *ajax* dan proses scraping dapat mengambil semua data yang ada pada sebuah halamam web. Pengembangan teknik web *scraping* merupakan proses yang cukup kompleks (Plamen Milev, 2017). Sehingga dibutuhkan pendekatan teknik *scraping* yang disesuaikan dengan sember *website*. Setelah mendapatkan data yang terstruktur, data tersebut disimpan dan dapat digunakan untuk proses lebih lanjut, misalnya proses statistik, proses sentimen analisis, dan lain-lain.

Mengekstraksi data *review* hotel dari sebuah *website* sangat dibutuhkan untuk menggali informasi yang ada pada data dalam proses teks *mining*. *Website* merupakan dokumen teks *html* yang datanya tidak terstruktur. Data *review* hotel yang ada pada dokumen *html* juga bercampur dengan teks-teks lain yang bukan merupakan data. Selain itu akses data pada halaman web menggunakan *ajax* dimana perubahan halaman web dilakukan tanpa semua halaman web. Proses untuk mengekstraksi dan mengolah informasi membutuhkan data yang terstruktur dalam sebuh model data.

Oleh karena itu, dibutuhkan sebuah sistem yang dapat melakukan ekstraksi data dari sebuah *website* berbasis *ajax*, melakukan *crawling* web berbasis *ajax* dan menghasilkan data terstruktur dalam sebuah model data.

Pengembangan sistem *scraping* dan *crawling* dibuat untuk mengekstraksi data dari *website*. Pengembangan diutamakan pada proses ekstraksi data untuk menghasilkan data terstruktur sesuai dengan model data yg telah ditentukan. Hal-hal yang tidak termasuk dalam pengembangan ini adalah:

1. Topologi jaringan dari sistem *scraping* ke *website* yang diproses
2. Peoses pengolahan data hasil ekstraksi
3. Basis data yang digunakan untuk menyimpan hasil ekstraksi

Pengembangan sistem *scraping* dan *crawling* ini bertujuan untuk membangun sistem yang dapat melakukan proses *scraping* dan *crawling* web berbasis *ajax* untuk menghasilkan data *review* hotel terstruktur. Data hasil ekstraksi dapat disimpan kedalam database untuk proses lebih lanjut.

Manfaat dari dikembangkannya sistem *scraping* dan *crawling* ini adalah agar lebih mudah mendapatkan data terstruktur dari sebuah *website*. Mengekstraksi data menggunakan sistem ini sangat membantu agar lebih cepat mendapatkan data dibandingkan melakukan pengambilan data secara manual. Dengan tersedianya data yang dibutuhkan, pemerintah dapat menentukan kebijakan-kebijakan terkait pengembangan sektor pariwisata melalui data atau informasi yang dihasilkan dari proses ekstrasi ini.

Manfaat lain dari proses ekstraksi ini adalah untuk menghasilkan data yang dibutuhkan dari *website* untuk kepentingan perusahaan atau individu. Ketika suatu perusahaan akan membuat atau mengembangkan suatu perangkat atau aplikasi, dibutuhkan analisa mengenai apa yang akan dibuat. Analisa tersebut dapat berupa data yang didapat dari calon pengguna secara langsung, maupun data dari para pengguna yg ada pada perangkat atau aplikasi yang pernah dibuat. Data tersebut dibutuhkan untuk mengukur kebutuhan calon pengguna, melihat respon dari

pengguna di aplikasi sebelumnya dan lain-lain. Dapat juga untuk melihat seberapa baik aplikasi yang sebelumnya dibuat, dalam hal ini bertujuan agar perusahaan dapat mengembangkan aplikasi menjadi lebih baik dan lebih nyaman digunakan oleh para pengguna.

ISI MAKALAH

1. Metodelogi Pengembangan Sistem

Pengembangan sistem *scraping* dan *crawling* dilakukan melalui beberapa tahapan. Tahapan-tahapan digunakan dalam pengembangan sistem *scraping* dan *crawling* ini mengacu pada journal *Conceptual Approach for Development of Web Scraping Application for Tracking Information* (Plamen Milev, 2017).

Langkah-langkah dalam metodelogi pengembangan sistem adalah sebagai berikut :

a. Analisa Kebutuhan. Tahap ini adalah tahap untuk melakukan evaluasi terhadap kebutuhan pengguna terkait dengan sistem yang akan dikembangkan.

b. Kebutuhan Fungsional. Menentukan fungsi-fungsi sistem yang dibutuhkan untuk memenuhi kebutuhan pengguna yang telah di evaluasi.

c. Mendefinisikan Sumber Data. Pada tahap ini akan ditentukan sumber data yang dibutuhkan oleh pengguna.

d. Analisa Sumber Data. Tahap ini merupakan tahap untuk menganalisa sumber data yang akan diekstraksi.

e. Analisa Sistem. Analasis sistem dibutuhkan untuk menentukan kebutuhan sistem dan menentukan modul-modul yang akan digunakan dalam sistem.

f. *Design Sistem*. *Design sistem* dibuat untuk memenuhi seluruh kebutuhan dan proses yang akan dibuat dalam sistem yang akan dibuat.

g. Implementasi dan pengujian. Tahap implementasi merupakan penjelasan dari proses implementasi yang dilakukan dalam pengembangan sistem. Setelah itu dilakukan pengujian sistem dengan cara *debugging.*

h. Proses Ekstraksi. Tahap ini menjelasakn bagaimana sistem melakukan proses ekstraksi pada sebuah *website.*

i. Penyimpanan Data Hasil Ekstraksi. Setelah proses ekstrkasi selesai, data hasil ekstraksi akan dikirim dan disimpan dalam *elasticsearch*.

2. Model Data

Tujuan utama dari *scraping* adalah mendapatkan data terstruktur dari sumber data yang tidak terstruktur. Model data dibutuhkan dalam menentukan struktur data yang akan di ekstrasi. Data yang ada pada struktur data mengikuti data yang dibutuhkan pada analisa kebutuhan data. Data yang dibutuhkan dan yang akan diekstraksi dari website dapat dilihat pada Tabel 1.

Tabel 1. Tabel data ekstraksi

No	Data	Tipe Data
1.	Nama Hotel	Text
2.	Bintang Hotel	Decimal
3.	Alamat Hotel	Text
4.	Rating Hotel	Decimal
5.	Rating Review	Decimal
6.	Tanggal Review	Datetime
7.	Name Review	Text
8.	Tema Review	Text
9.	Teks Review	Text

Model data ini diimplementasikan dalam *class HotelReview*. *Class HotelReview* merupakan turunan dari *class scrapy.Item* dan ditempatkan pada *file items.py*. Di dalam *items*.py terdapat *class HotelReview* yang berfungsi untuk memetakan data yang telah di scraping untuk disalurkan ke *pipelines* dan disimpan kedalam *elasticsearch*.

3. Analisa Web Traveloka

Website Traveloka perlu dianalisa untuk mengetahui hubungan atau *link* antar halaman dan letak data yang akan diekstrasi. Hubungan *link* antara halaman diperlukan untuk melakukan proses *crawling*. Sedangkan letak data yang akan dikestrasi diperlukan untuk melakukan proses *scraping*.

Ada 3 halaman utama yang menjadi perhatian dan letak sumber data. Pertama halaman *home* sebagai awal akses ke web Traveloka, kedua halaman hotel *list* dimana terdapat daftar semua *link* hotel, dan ketiga adalah halaman hotel dimana terdapat *review* hotel. Hubungan antar 3 halaman tersebut dapat dilihat pada Gambar 1.

Gambar 1. Struktur web Traveloka

Pada halaman web Traveloka bagian daftar hotel dan daftar *review* pada tiap hotel merupakan data *Ajax*. Sebelum melakukan ekstraksi harus ditentukan elemen mana saja yang didalamnya terdapat data yang ingin diekstraksi. Untuk memilih elemen-elemen pada teks *html*, diperlukan sebuah *selector*. *Selector* yang akan digunakan adalah *Xpath selector*.

X*path* untuk membuka halaman hotel, elemen canonical dan proses crawling dapat dilihat pada *Tabel 2*. Sedangkan *xpath* untuk data yang akan diekstraksi dapat dilihat pada

Tabel 3.

Tabel 2. Tabel Xpath 1

No	Nama	XPath
1.	Untuk membuka halaman hotel	//div[@class='mMmI2 CZtP0 tvat-searchListItem']
2.	Elemen Canonical	//link[@rel='canonical']
3.	Button-next	//div[@id='next-button']

Tabel 3. Tabel XPath

No	Data	(XPath)
1.	Nama Hotel	//h1[@itemprop='name']
2.	Bintang Hotel	//meta[@itemprop='ratingValue']
3.	Alamat Hotel	//span[@itemprop='streetAddress']
4.	Rating Hotel	//div[@class='_3TWbq']
5.	Rating Review	//div[@itemprop='review']
6.	Tanggal Review	//div[@itemprop='review']
7.	Name Review	//div[@itemprop='review']
8.	Tema Review	//div[@itemprop='review']
9.	Teks Review	//div[@itemprop='review']

4. Instalasi Scrapy Framework

Setelah mengetahui struktur dari *website* Traveloka, proses instalasi semua modul dilakukan ditahap ini. Implementasi menggunakna bahasa python, sehingga semua modul di-*install* di dalam *PIP* (*Python Package Management*). Sebelum melakukan instalasi pada *scrapy*, *install extention python* terlebih dahulu pada *Visual Studio Code*. *Python* yang digunakan adalah *python 3.6*. Scrapy di *install* pada terminal *Visual Studio Code*.

Selain melakukan instalasi pada *scrapy*, instalasi *selenium* dan *elasticsearch* juga dilakukan pada tahap ini. Setelah semua modul di *install*, maka yang selanjutnya yang dilakukan adalah membuat *Scrapy Project* pada *Workspace Visual Studio Code* dengan menuliskan perintah :

File Directory project akan muncul ketika proses *startproject* berhasil dilakukan. *File directory project* terletak di dalam *workspace* pada bagian *explorer Visual Studio Code*. Setelah membuat *Scrapy Project*, selanjutnya adalah membuat satu *file spider* pada *folder spider*. *Spider* yang dibuat adalah *traveloka_spider.py*.

Pengembangan teknik *scraping* dan *crawling* dilakukan dengan membuat *class spider* pada *file traveloka_spider.py* yg diletakkan pada direktori *spiders*. *File item.py* berisi satu class HotelReview yang merupakan implementasi dari model data hasil scraping. Selain itu juga terdapat file setting.py.

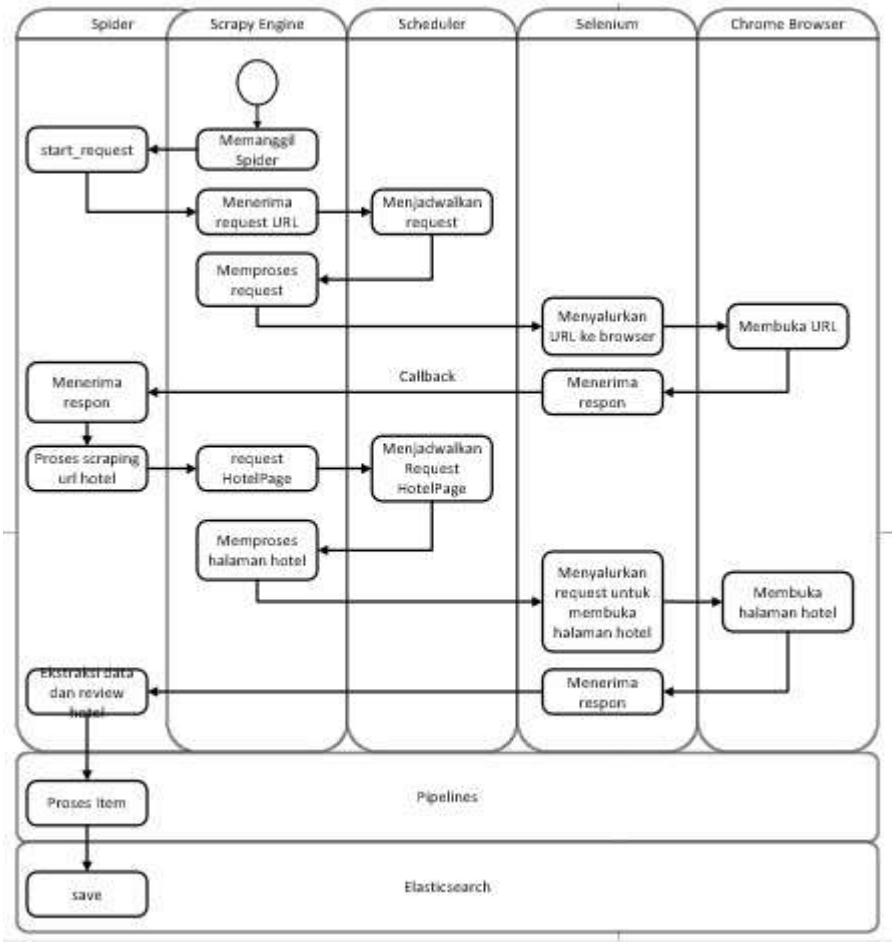
Setting.py merupakan file untuk konfigurasi framework scrapy.

5. Pengembangan scrapy

Pengembangan *scrapy* dimulai dari proses pada *activity diagram*, melakukan konfigurasi pada *scrapy*, membuat model data pada *items*, membuat *code scraping* dan *crawling* pada *spider* dan konfigurasi pada *elasticserach* sebagai tempat penyimpanan data yang telah diekstraksi.

a. Activity Diagram

Gambar 2 merupaka *Activity Diagram* proses *scraping* dan *crawling*. Proses ekstraksi data dilakukan di dalam *spider*. Spider merupakan sebuah *class* yang dibuat dalam *scrapy project. Code* yang terdapat dalam *spider* dimulai dari sebuah request sampai respon. Proses pertama adalah *engine scrapy* memanggil *spider* melalui *strat_request*. *Method start_request* dalam *spider* berisi *link url* dari daftar *list* hotel.



Gambar 2. *Activity diagram*

Scheduler menjadwalkan *request* yang masuk melalui *scrapy engine*. Setelah menerima *request* yang telah dijadwalkan oleh *scheduler*, *scrapy engine* akan mengirim *request* tersebut ke *selenium* dan dilanjutkan ke *chrome driver* untuk diproses oleh *chrome*. Setelah itu *spider* akan menerima respon dari *chrome*. *Spider* akan melakukan *callback* dan melanjutkan ke *method* selanjutnya.

Method selanjutnya adalah *method hotelList*. *Method hotelList* digunakan untuk mengambil *link url* (*canonical*) dari semua hotel yang ada di Yogyakarta. Setiap hotel dari daftar hotel dibuka terlebih dahulu menggunakan fitur *click* dari *selenium*. Setelah halaman hotel dibuka, lalu *link url* hotel dapat diambil dengan menggunakan *xpath selector*.

Proses itu terjadi berulang sampai sistem mendapatkan semua *link url* dari daftar hotel.

Jika semua *link url* hotel sudah masuk ke dalam *listURL*, maka *spider* akan mengirim *list url* ke *scheduler* melalu *scrapy engine* untuk dijadwalkan oleh *scheduler*. Setelah itu, *list* yang sudah dijadwalkan oleh *scheduler* akan dikirim ke *selenium* melalui *scrapy engine* untuk selanjutnya di proses di dalam *chrome driver*. Halaman hotel akan dibuka berdasarkan *list url* yang diterima oleh *chrome* dan akan diekstraksi data yang dibutuhkan sesuai yang elemen yang di tulis pada *xpath*. Lalu proses ekstraksi dilakukan oleh sistem secara berulang sampai semua *review* hotel di ekstraksi.

Pada saat ekstraksi data review hotel berlangsung, saat satu data *review* berhasil di ekstraksi data tersebut akan masuk ke dalam *item pipelines* dan oleh *item pipelines* akan diteruskan ke *elasticsearch*. *Elasticseach* adalah tempat akhir dimana data hasil *scraping* akan disimpan atau diproses lebih lanjut.

b. Konfigurasi Scrapy

Konfigurasi *scrapy* di dilakukan di dalam *setting.py*. Konfigurasi pada *scrapy* dilakukan untuk mengubah mengatur pada *browser*.

Pada *setting.py* konfigurasi yang dilakukan berupa *disable robotstxt, concurrent request, download delay, disable cookies* dan bahasa yang digunakan *browser*.

c. Spider Url Hotel

Proses ekstraksi *url* hotel sebelum mendapatkan *review* hotel terjadi di dalam *traveloka_spider.py*. Terdapat dua *method* yang akan diproses sebelum ekstraksi pada *review* hotel dilakukan, yaitu *method start_request* dan *method hotelList.*

Method start_request digunakan untuk melakukan *start_request. Start_request* dimulai dengan memasukan *url* yang akan dibuka pada *browser. Url* yang digunakan pada *start_request* adalah *url* daftar *list* hotel yang ada di Yogyakarta untuk kemudian di proses pada *method* selanjutnya untuk mengambil *url* dari masing-masing hotel.

Method diatas merupakan *method hotelList* yang digunakan untuk mengambil *list url* hotel (*canonical*). Pada bagian *options* diatur *options* untuk *webdriver*. *Browser* yang digunakan adalah *headless browser*. Pada *method* ini, sistem akan membuka satu *browser* untuk membuka daftar hotel. Pada saat meng-klik hotel di dalam daftar hotel, halaman hotel akan di tampilkan pada *tab* baru. Setelah itu, proses ekstrkasi pada elemen *canonical* akan dilakukan. Setelah mendapat *url* hotel, *url* tersebut akan disimpan di dalam *listURL* dan sistem akan menutup *tab* halaman hotel tersebut.

Setelah daftar hotel pada bagian *ajax* pertama selesai, sistem akan membaca *button next* dan otomatis melakukan *crawling* untuk mendapatkan *url* hotel pada halaman selanjutnya. Jika masih terdapat *next-page*, maka proses ekstraksi *url* hotel akan diteruskan sampai sistem mendapatkan semua *url* hotel yang terdapat pada daftar hotel. Setelah semua *url* hotel telah masuk ke dalam *listURL*, maka sistem akan mengirim semua *url* hotel yang terdapat pada *listURL* ke *scheduler*. Lalu sistem akan mengahpus *listURL*.

d. Spider Review Hotel

Setelah semua *url* hotel di dapatkan, sistem akan menjalankan *method hotelPage* di dalam *traveloka_spider* untuk melakukan ekstraksi pada *review* di masing-masing hotel.

Method ini berfungsi untuk mengekstraksi data hotel dan *review* hotel yang dibutuhkan. Pada bagian *options* diatur *options* untuk *webdriver*. *Browser* yang digunakan adalah *headless browser*.Setelah semua *url* hotel masuk di *scheduler*, *scheduler* akan mengatur jadwal untuk masing-masing *url* hotel yang akan dibuka oleh *browser*. Dalam *setting.py* diatur *concurrent request* untuk *browser*, yaitu *browser* dapat membuka 8 *url* hotel dari *scheduler* untuk dibuka secara bersamaan dalam 8 *browser*.

Setelah *url* hotel dibuka oleh *browser*, sistem akan memulai ekstraksi pada data hotel berupa nama hotel, alamat hotel, *rating* hotel, dan bintang hotel. Lalu ekstraksi selanjutnya dilakukan pada *review* hotel. Semua data *review* hotel yang berupa teks akan diekstraksi seperti nama, *rating*, tanggal, tema dan teks *review*. Setelah selesai mengekstraksi daftar *review* yang pertama, sistem akan melakukan *crawling* untuk membuka daftar *review* selanjutnya dan mengekstraksi *review* selanjutnya sampai pada *review* terakhir di daftar *review* hotel tersebut. Proses itu terjadi pada masing-masing halaman hotel yang akan di ekstraksi.

Ketika sistem telah mendapatkan hasil ekstraksi dari halaman hotel, data hasil ekstraksi akan dikirim ke *items pipelines* untuk

selanjutnya di simpan dalam *elasticsearch*. Dan saat semua halaman hotel sudah selesai di esktraksi, sistem akan menutup dan menghapus semua halaman hotel.

Method getReviewDate merupakan *method* yang berfungsi untuk mengubah format tanggal pada hasil ekstraksi menjadi format tanggal yang bisa diterima oleh elasticsearch. Pada *method hotelPage*, data yang di ekstraksi semuanya adalah berupa data tesk termasuk tanggal, jadi dibutuhkan *method getReviewDate* untuk mengubah format tanggal yang awalnya teks menjadi *datetime*.

e. Konfigurasi Modul Elasticsearch

Konfigurasi modul *elasticsearch* dilakukan di dalam *setting.py*. Didalam *pipelines.py* terdapat satu *class* yang digunakan untuk menerima *object* berupa *items* dari *object* yang telah dibuat pada *items.py*. Pada *setting.py* dilakukan konfigurasi *pipelines* untuk *elasticseacrh* agar data yang bisa disimpan di dalam *elasticsearch*. Angka 500 pada *ScrapyElasticsearch* menunjukan prioritas untuk proses di *pipelines*. Selanjutnya terdapat konfigurasi *elasticsearch* untuk *server, index, format index date, username, password, type* dan *buffer length*.

f. Pengujian

Pengujian sistem scraping dan crawling dibagi menjadi dua tahap yaitu pengujian fungsional dan pengujian proses.

Pengujian Fungsional Scraping dan Crawling

Pengujian Sistem *Scraping* dan *Crawling* dilakukan dengan cara *Debugging*. Proses *Debugging* dilakukan dengan cara memberikan *break point* pada *code* yang ada di dalam *traveloka_spider* untuk melihat apakah data berhasil di ekstraksi dan proses *crawling* berjalan dengan baik.

Gambar 3. Pengujian ekstraksi *list url* hotel

Pada Gambar 3 menunjukan hasil *list url* hotel yang sudah berhasil diekstraksi. *List url* hotel yang sudah diekstraksi dapat dilihat dalam *listURL* pada bagian *variables*.

Selanjutnya pengujian untuk ekstraksi data hotel juga dilakukan dengan cara *debugging*. Pada Gambar 4 di dalam *local* terlihat hasil data hotel yang telah berhasil diekstraksi.

Gambar 4. Pengujian ekstraksi data hotel

Setelah melakukan ekstraksi data pada halaman hotel, proses selanjutnya adalah ekstraksi pada data *review* hotel. Proses ekstraksi *review* pada halaman hotel berhasil dilakukan.

Pada Gambar 5 dapat dilihat hasil pengujian untuk proses ekstraksi *review* hotel. Pada bagian *variables*, didalam *review content* adalah data *review* yang telah berhasil diekstraksi.

Gambar 5. Pengujian ekstraksi *review* hotel

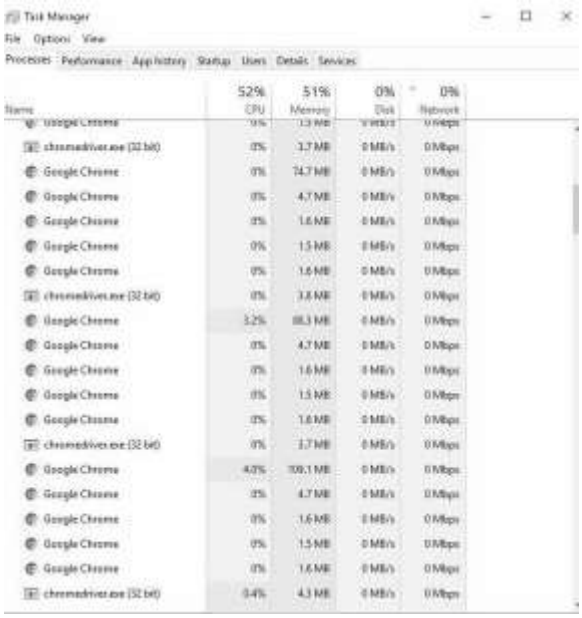
Gambar 6. Pengujian *crawling* 1

Pada Gambar 6 menujukukan proses *crawling* pada sistem berhasil dilakukan. Proses *crawling* dilakukan pada saat ekstraksi pada *url* hotel dan *review* hotel. Jika terdapar *button-next* atau halaman selanjutnya yang terdapat dalam *ajax*, maka sistem akan otomatis melakukan *crawling* untuk memuat daftar selanjutnya. Namun, jika tidak terdapat *button-next*, maka sistem tidak akan melakukan *crawling* dan langsung menuju proses selanjutnya seperti pada Gambar 7.

Gambar 7. Pengujian *crawling* 2

Pengujian Proses Scraping dan Crawling

Pengujian pada proses *scraping* dan *crawling* dilakukan saat sistem dijalankan. Pengujian ini dilakukan dengan cara melihat pada *task manager* terkait *concurrent* dan melihat hasil akhir dari proses *scraping* dan *crawling* pada *terminal*.



Gambar 8. Pengujian proses *concurrent*

Pengujian pada *setting concurrent* dapat dilihat pada *task manager* pada Gambar 8 ketika sistem dijalankan. *Concurrent* request yang diatuar pada *setting.py* adalah 8, yaitu menunjukan 8 *browser* dapat dijalankan secara bersamaan untuk membuka *url* hotel dan melakukan proses ekstraksi pada halaman hotel.

```
2018-08-14 07:57:47 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'downloader/exception_count': 70,
'downloader/exception_type_count/twisted.internet.error.TimeoutError': 1,
'downloader/exception_type_count/twisted.web._newclient.ResponseNeverReceived': 69,
 'downloader/request_bytes': 155522,
 'downloader/request_count': 491,
 'downloader/request_method_count/GET': 491,
 'downloader/response_bytes': 36882418,
 'downloader/response_count': 421,
 'downloader/response_status_count/200': 421,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2018, 8, 14, 0, 57, 47, 748515),
 'item_scraped_count': 174770,
 'log_count/DEBUG': 1165453,
 'log_count/ERROR': 14,
 'log_count/INFO': 743,
 'log_count/WARNING': 3226,
 'memusage/max': 136798208,
 'memusage/startup': 64970752,
 'request_depth_max': 1,
 'response_received_count': 421,
 'retry/count': 62,
 'retry/max_reached': 8,
'retry/reason_count/twisted.web._newclient.ResponseNeverReceived': 62,
 'scheduler/dequeued': 491,
 'scheduler/dequeued/memory': 491,
 'scheduler/enqueued': 491,
 'scheduler/enqueued/memory': 491,
 'spider_exceptions/NoSuchElementException': 1,
 'spider_exceptions/ValueError': 3,
 'start_time': datetime.datetime(2018, 8, 13, 17, 36, 4, 295672)}
2018-08-14 07:57:47 [scrapy.core.engine] INFO: Spider closed (finished)
```

Laporan hasil *scraping* dan *crawling* akan muncul pada tampilan terminal setelah proses selesai dilakukan. Pada laporan tersebut terdapat detail hasil proses *scraping* dan *crawling* berupa proses berhasil dilakukan, jumlah *items* yang telah di ekstraksi sebanyak 174770, jumlah antrian pada scheduler sebanyak 491, crawling yang berhasil dilakukan sebanyak 421, crawling yang gagal dilakukan sebanyak 8, dan crawling yang melakukan retry sebanyak 62.

g. Hasil

Data hasil ekstraksi akan disimpan pada *elasticsearch* melalui *pipelines*. Gambar 9 adalah hasil data yang sudah berada pada *elasticsearch*. Semua data yang didapat dari hasil ekstraksi akan masuk dalam *index elasticsearch*. Data yang diekstraksi adalah data hotel dan *review* hotel yang ada di Yogyakarta. Dibagian kiri atas halaman terdapat jumlah semua data yang telah diekstraksi.

Gambar 9 Hasil ekstraksi

Masing-masing data *review* hotel yang ada pada *elasticseacrh* dapat dilihat dalam *format json* seperti pada Gambar 10. Data hasil ekstraksi merupakan data terstruktur dengan model data sesuai dengan yang telah didefinisikan pada *class item*.

Gambar 10 Hasil ekstraksi (*json*)

KESIMPULAN

Pengembangan sistem *scraping* dan *crawling* telah berhasil dilakukan dengan beberapa proses pengujian pada sistemnya. Dari proses *scraping* dan *crawling* yang telah dilakukan, kesimpulan yang telah didapat adalah :

1. Proses *scraping* telah berjalan dengan baik berdasarkan pengujian yang telah dilakukan.
2. Proses *crawling* sudah dapat dilakukan pada sistem untuk melakukan *crawling* pada *button next* di setiap daftar hotel dan daftar *review* hotel.
3. Penyimpanan data dilakukan di *elasticsearch* melalui *item pipelines*.
4. Semua proses *concurrent* berjalan dengan baik sesuai konfigurasi yang dilakukan pada *setting.py*.

UCAPAN TERIMA KASIH

Dalam melakukan pengembangan sistem penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terima kasih yang tak terhingga kepada:

1. Bapak M. Helmi Zain Nuri, S.T., M.T. selaku pembimbing pertama.
2. Bapak Chayadi Oktomy Noto S., S.T., M.Eng., ITILF. selaku pembimbing kedua.
3. Bapak Dr. Ir. Dwijoko Purbohadi, M.T. selaku dosen penguji.
4. Semua pihak yang tidak dapat penulis sebutkan satu-persatu, yang telah membantu secara langsung maupun tidak langsung.

PENULISAN PUSTAKA DAN DAFTAR PUSTAKA

Dyah Sugandini. (2018). Anteseden Loyalitas Konsumen pada Industry Perhotelan. 197.

Carlson, J. (2018). Customer engagement behaviours in social media: capturing innovation opportunities. *Journal of Service Marketing*.

Kementerian Pariwisata. (2016). *Laporan Akuntabilitas Kinerja Kementerian Pariwisata tahun 2016.* Biro Perencanaan dan Keuangan Sekretariat Kementerian.

Ni Nyoman Ayu Wiratini M. (2018). Analisis Faktor-Faktor yang Mempengaruhi Niat Kunjungan Kembali Wisatawan pada Daya Tarik Wisata di Kabupaten Bandung.

Plamen Milev. (2017). Conceptual Approach for Development of Web Scraping Application for Tracking Information.

Wilton, T. d. (1998). Kandampully.

PENULIS:

Selvi Oktaria
Teknik Informatika, Teknik, Universitas Muhammadiyah Yogyakarta, Yogyakarta.
Email: selvioktaria123@gmail.com
.