

## BAB IV IMPLEMENTASI dan PEMBAHASAN

### 4.1. 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 4- 1*.

Tabel 4- 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*.

```

class HotelReview(scrapy.Item):
    hotelName = scrapy.Field()
    hotelStar = scrapy.Field()
    hotelAddress = scrapy.Field()
    hotelRating = scrapy.Field()
    reviewRating = scrapy.Field()
    reviewDate = scrapy.Field()
    reviewName = scrapy.Field()
    reviewTheme = scrapy.Field()
    reviewText = scrapy.Field()

```

## 4.2. Analisa Web Traveloka

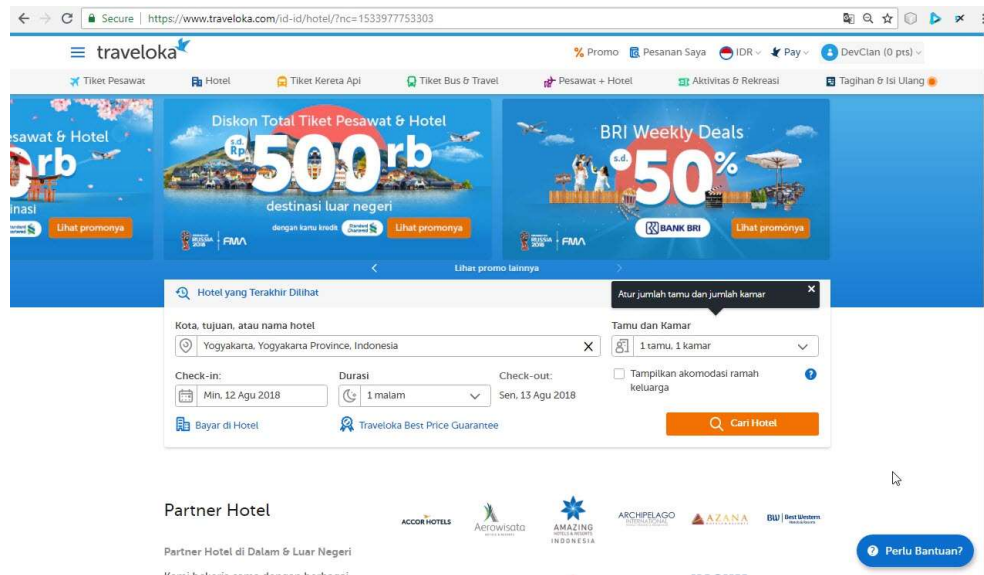
*Website* Traveloka perlu dianalisa untuk mengetahui hubungan atau *link* antar halaman dan letak data yang akan diekstrasi. Hubungan *link* antara halaman diperlukan untuk melakukan proses *crawling*. Sedangkan letak data yang akan diekstrasi diperlukan untuk melakukan proses *scraping*.

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 4-1.

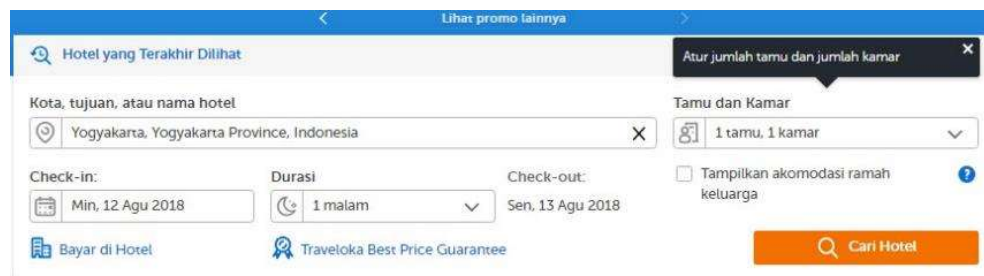


Gambar 4-1. Struktur web Traveloka

Ketika membuka *website* Traveloka dengan *url* <https://www.traveloka.com/id-id/hotel>, halaman yang tampil adalah halaman pencarian hotel seperti pada Gambar 4-2.

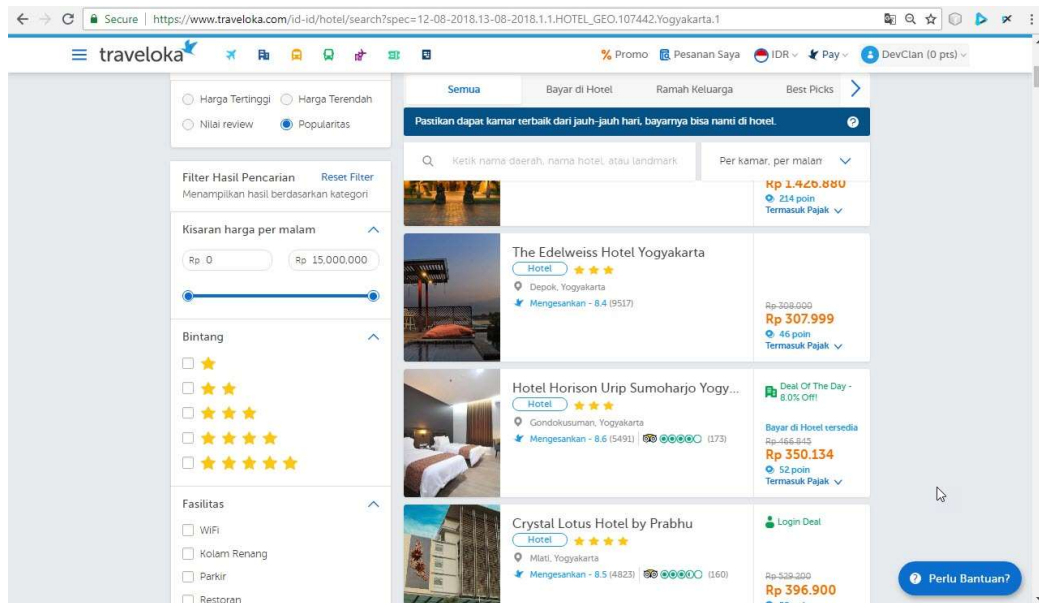


Gambar 4-2. Halaman pencarian hotel



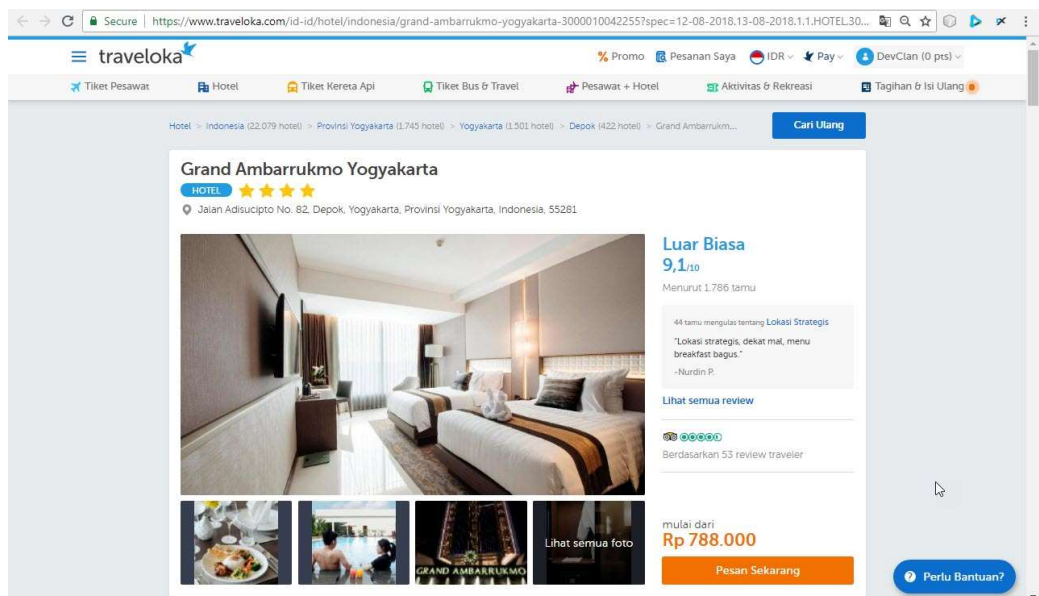
Gambar 4-3. Form pencarian hotel

Pada halaman pencarian hotel, *form* pencarian harus diisi untuk mendapatkan daftar hotel yang sesuai dengan yang diinginkan. *Form* pencarian hotel dapat dilihat pada Gambar 4-3. Isi *form* pencarian dengan mengganti nama lokasi menjadi Yogyakarta dan tekan *button* Cari Hotel, maka akan menampilkan daftar hotel yang ada di Yogyakarta seperti pada Gambar 4-4.



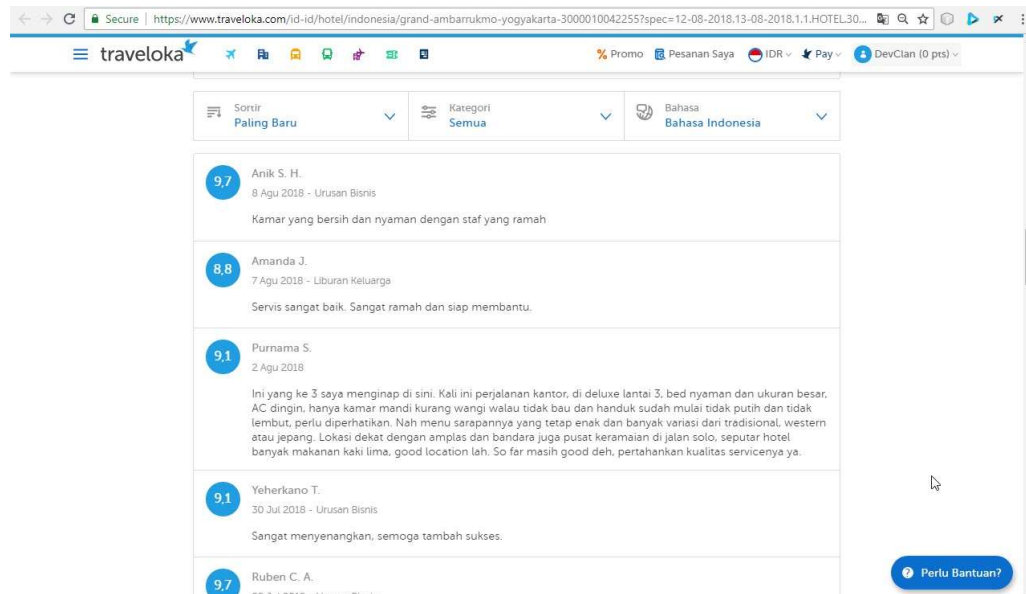
Gambar 4-4. Halaman daftar hotel

Setelah itu untuk dapat melihat *review* masing-masing hotel, harus masuk kedalam halaman hotel tertentu. Klik bagian salah satu hotel dalam daftar hotel dan akan muncul halaman hotel seperti pada Gambar 4-5.



Gambar 4-5. Halaman hotel Traveloka

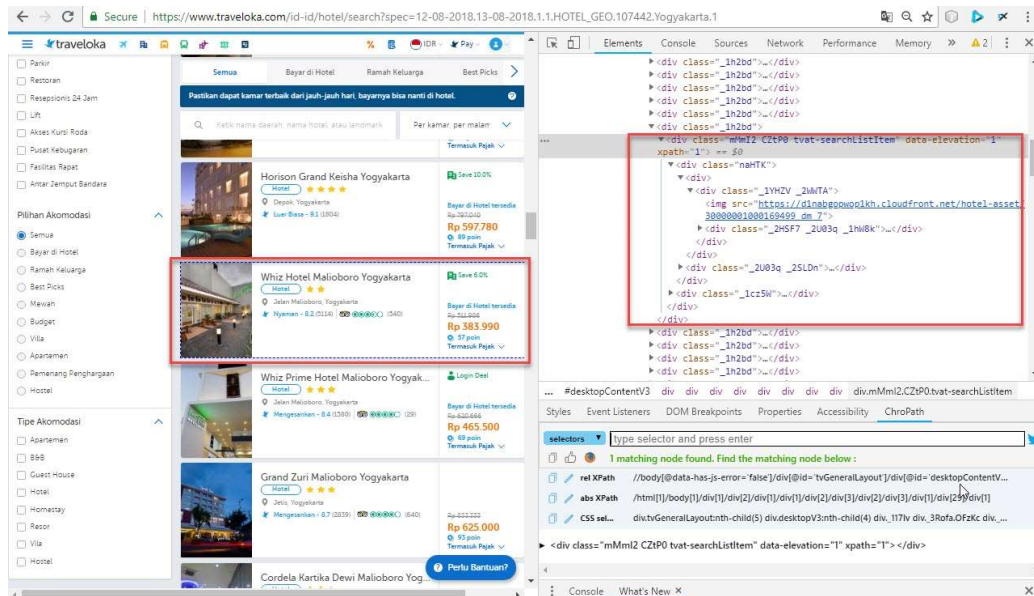
*Review* dari sebuah hotel terdapat pada bagian bawah pada setiap halaman hotel seperti Gambar 4-6.



Gambar 4-6. Review hotel 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*.

Untuk mendapatkan *review* dari masing-masing hotel, maka halaman dari tiap-tiap hotel harus dibuka terlebih dahulu. Untuk membuka halaman hotel pada daftar hotel melalui *selenium* diperlukan elemen dari masing-masing hotel seperti pada Gambar 4-7.



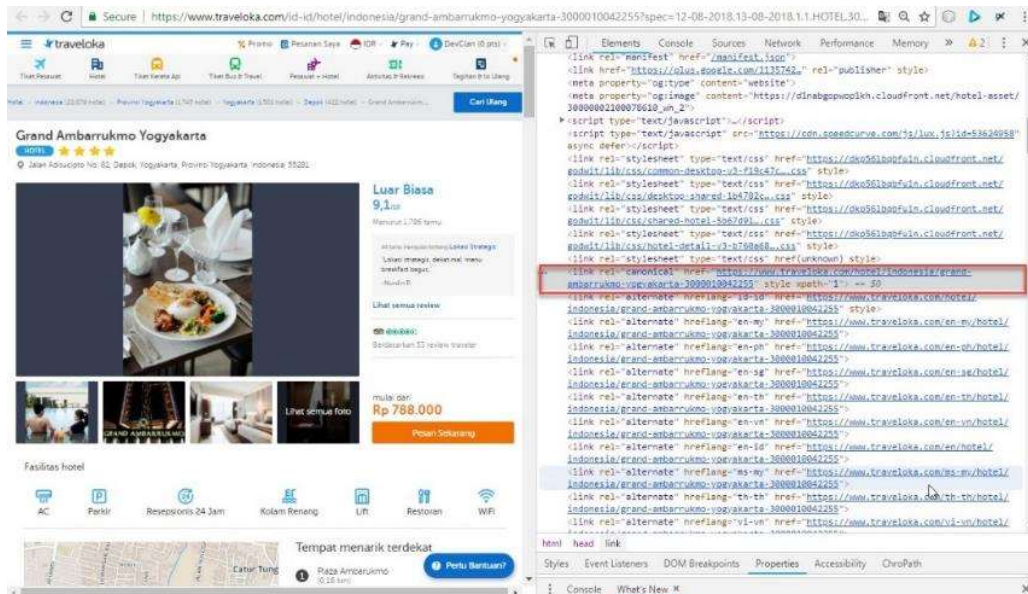
Gambar 4-7. Elemen url hotel

Gambar 4-8 merupakan elemen yang digunakan untuk membuka halaman pada masing-masing hotel.



Gambar 4-8. Elemen membuka halaman hotel

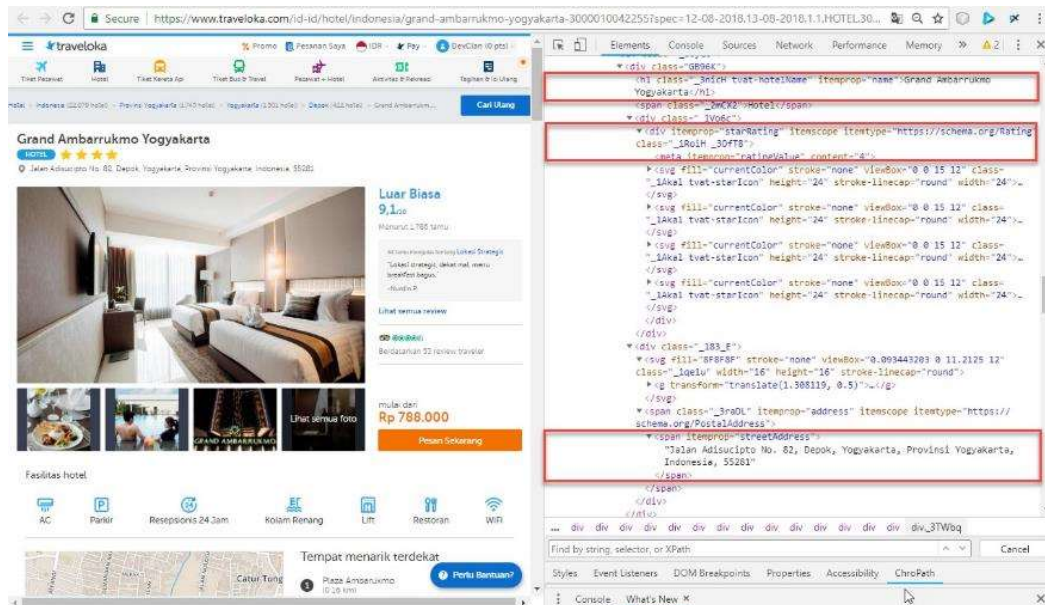
Setelah masuk pada halaman hotel, maka yang diperlukan adalah mengambil *link url* dari hotel tersebut tanpa menggunakan tanggal pencarian hotel. Untuk mengambil link url tersebut elemen yang digunakan dapat dilihat pada Gambar 4-9.

Gambar 4-9. Elemen *url* hotel

```
... <link rel="canonical" href="https://www.traveloka.com/hotel/indonesia/grand-ambarrukmo-yogyakarta-3000010042255" style xpath="1"> == $0
```

Gambar 4-10. Elemen *canonical*

Elemen *Canonical* pada Gambar 4-10. Merupakan elemen yang terdapat *link url* hotel tanpa kata kunci pencarian atau tanggal pencarian. Selanjutnya, setelah link hotel yang telah di dapat itu dibuka, muncul halaman hotel yang akan diekstraksi datanya. Untuk mengekstraksi halaman hotel juga perlu dipilih elemen-elemen yang akan digunakan seperti pada Gambar 4-11.



Gambar 4-11. Elemen data hotel

```

▼<div class="GB96K">
  <h1 class="_3nicH tvat-hotelName" itemprop="name">Grand Ambarukmo
  Yogyakarta</h1>

```

Gambar 4-12. Elemen nama hotel

```

<div class="_3TWbq" style>9,1</div>
</div>

```

Gambar 4-13 Elemen *rating* hotel

Gambar 4-12 merupakan elemen untuk nama hotel, sedangkan Gambar 4-13 merupakan elemen yang digunakan untuk mengambil *rating* hotel.

```

▼<div itemprop="starRating" itemscope itemtype="https://schema.org/Rating"
class="_1RoiH_30fT8">
  <meta itemprop="ratingValue" content="4">

```

Gambar 4-14. Elemen bintang hotel

```

▼<span itemprop="streetAddress">
  "Jalan Adisucipto No. 82, Depok, Yogyakarta, Provinsi Yogyakarta,
  Indonesia, 55281"
</span>

```

Gambar 4-15. Elemen alamat hotel





```

▼<div class="_3Q18j _14ETA" itemprop="review" itemscope
itemtype="https://schema.org/Review">
  ▼<div class="Xc61c" itemprop="reviewRating" itemscope
itemtype="https://schema.org/Rating">
    <meta itemprop="bestRating" content="10">
    <span itemprop="ratingValue" content="9.7">9,7</span>
  </div>
  ▼<div class="r44Gh">
    ▼<div class="TbFQ4">
      <div class="NkdVJ" itemprop="author">Anik S. H.
      </div>
      ▼<div class="_3ydMH">
        <span itemprop="datePublished" content="2018-08-
08">8 Agu 2018</span>
        ▶<span>...</span>
      </div>
    </div>
    <div class="_1mI1m" itemprop="reviewBody">Kamar yang
bersih dan nyaman dengan staf yang ramah</div>
  </div>
</div>

```

Gambar 4-17. Elemen *review*

Pada bagian *ajax* yang terledak di daftar *list* hotel dan daftar *review* pada tiap hotel, terdapat *button next* yang akan di *crawling* oleh *scrapy* untuk membuka list berikutnya. Elemen pada *button next* dapat dilihat pada Gambar 4-18.

```
<div class="fyGvr tvat-pageButton" id="next-button">Next</div>
```

Gambar 4-18. Elemen *button next*

Semua elemen-elemen tersebut akan digunakan dalam proses *scraping* dan *crawling* untuk membuka halaman hotel, memilih data-data yang akan diekstraksi dan untuk melakukan melakukan *crawling* pada *list* data berikutnya yang akan diekstraksi.

*Xpath* untuk membuka halaman hotel, elemen canonical dan proses *crawling* dapat dilihat pada Tabel 4- 2. Sedangkan *xpath* untuk data yang akan diekstraksi dapat dilihat pada Tabel 4- 3.

Tabel 4- 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 4- 3. Tabel *XPath* 2

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']

### 4.3. Instalasi Scrapy Framework

Setelah mengetahui struktur dari *website* Traveloka, proses instalasi semua modul dilakukan ditahap ini. Implementasi menggunakan 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* dengan perintah :

```
Pip install Scrapy
```

Selain melakukan instalasi pada *scrapy*, instalasi *selenium* dan *elasticsearch* juga dilakukan pada tahap ini dengan perintah :

```
Pip install Scrapy-Selenium
```

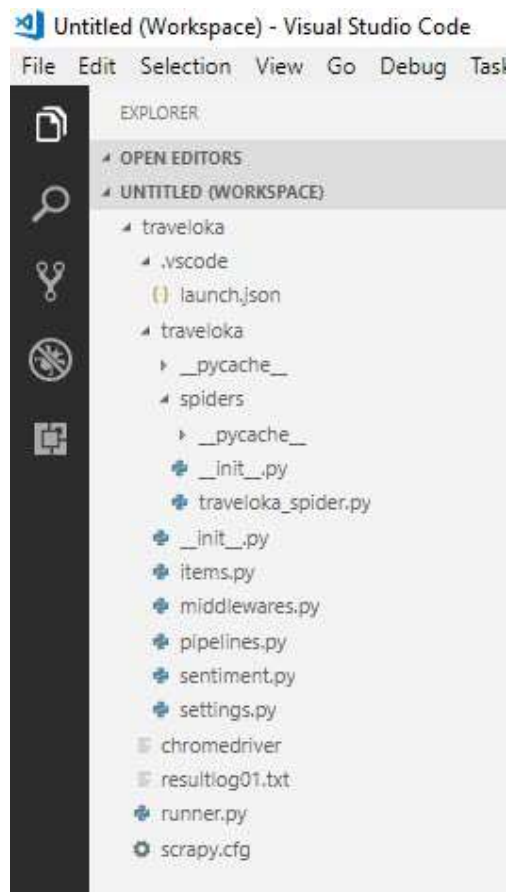
```
Pip install ScrapyElasticSearch
```

Setelah semua modul di *install*, maka yang selanjutnya yang dilakukan adalah membuat *Scrapy Project* pada *Workspace Visual Studio Code* dengan menuliskan perintah :

```
scrapy startproject traveloka
```

*File Directory project* akan muncul ketika proses *startproject* berhasil dilakukan. *File directory project* terletak di dalam *workspace* pada bagian *explorer* *Visual Studio Code* seperti pada Gambar 4-19.

Setelah membuat *Scrapy Project*, selanjutnya adalah membuat satu *file spider* pada *folder spider*. *Spider* yang dibuat adalah *traveloka\_spider.py*.



Gambar 4-19. *File directory scrapy project*

Gambar 4-19 merupakan tampilan direktori dari *project scrapy* yang dibuat. Direktori *spiders* merupakan tempat untuk *file source code spider*. 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*.

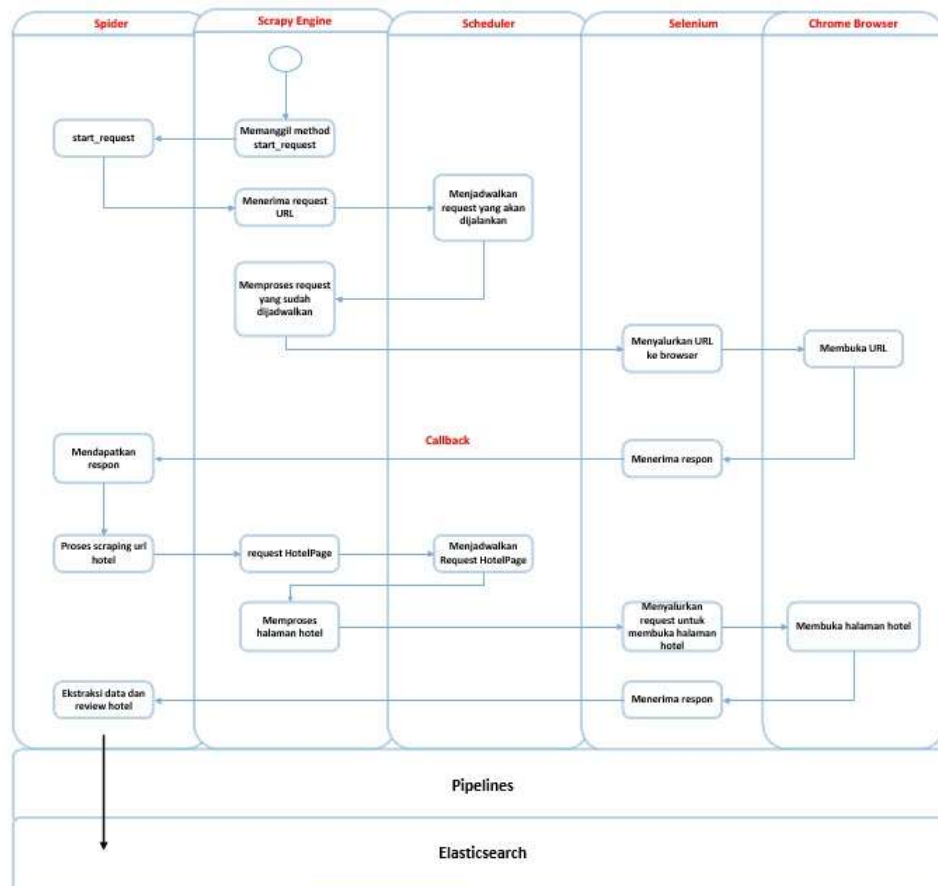
Selain itu juga terdapat *file setting.py*. *Setting.py* merupakan *file* untuk konfigurasi *framework scrapy*.

#### **4.4. 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 *elasticsearch* sebagai tempat penyimpanan data yang telah diekstraksi.

##### **4.4.1. Activity Diagram**

Pengembangan teknik *scraping* dan *crawling* diawali dengan merencanakan proses *scraping* dan *crawling*. Perencanaan proses dituangkan dalam diagram aktifitas. Pada diagram aktifitas dijabarkan proses yang terjadi pada semua modul modul yang digunakan.



Gambar 4-20. Activity diagram

Gambar 4-20 merupakan *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 *method strat\_request*. *Method start\_request* dalam *spider* berisi *link url* dari daftar *list* hotel. *Scheduler* menjadwalkan *request* yang masuk melalui *scrapy engine*. Setelah menerima *http request* yang telah dijadwalkan oleh *scheduler*, *scrapy engine* akan mengirim *request* tersebut ke *selenium* dan dilanjutkan ke *chrome driver* untuk diproses oleh *browser chrome*. Setelah itu *spider* akan menerima *http respon website* yang akan diakses dari *chrome*. *Spider*

akan mendapatkan respon melalui *method callback* yang telah didefinisikan pada *request scrapy request*. *Method callback* berikutnya adalah *method hotelList*.

*Method* selanjutnya adalah *method hotelList*. *Method hotelList* digunakan untuk mengambil *link url (canonical)* dari semua hotel yang ada. Setiap hotel dari daftar hotel dibuka terlebih dahulu menggunakan fitur *click* dari *selenium*. Proses klik dilakukan menggunakan *javascript* untuk *action click 'arguments[0].click()*. Klik dengan *javascript* digunakan karena respon aksi klik diproses dengan *javascript* untuk membuka halaman hotel. 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* melalui *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*. *Elasticsearch* adalah tempat akhir dimana data hasil *scraping* akan disimpan atau diproses lebih lanjut.

#### 4.4.2. Konfigurasi Scrapy

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

```
BOT_NAME = 'traveloka'
SPIDER_MODULES = ['traveloka.spiders']
NEWSPIDER_MODULE = 'traveloka.spiders'

ROBOTSTXT_OBEY = False
CONCURRENT_REQUESTS = 8
DOWNLOAD_DELAY = 10
```

```

CONCURRENT_REQUESTS_PER_DOMAIN = 16
CONCURRENT_REQUESTS_PER_IP = 16

COOKIES_ENABLED = False

DEFAULT_REQUEST_HEADERS = {
    'Accept-Language': 'id',
}

```

Pada *setting.py* konfigurasi yang dilakukan berupa *disable robotstxt*, *concurrent request*, *download delay*, *disable cookies* dan bahasa yang digunakan *browser*. *Disable robotstxt* digunakan agar *scrapy* dapat melakukan *crawling* halaman-halaman tanpa mengikuti *setting* akses di *web server*. *Concurrent request* digunakan untuk menentukan jumlah akses *web browser* secara bersamaan. *Download delay* digunakan untuk memberikan jeda antar akses web. Dan *disable cookies* digunakan supaya akses web tanpa melibatkan *cookies* sehingga akses web jadi lebih cepat. Untuk web Traveloka dapat diakses tanpa menggunakan *cookies*.

#### 4.4.3. Spider Url Hotel

Proses ekstraksi *url* hotel dilakukan terlebih dahulu. Terdapat dua *method* yang akan diproses sebelum ekstraksi pada *review* hotel dilakukan, yaitu *method start\_request* dan *method hotelList*.

```

class TravelokaSpider(scrapy.Spider):
    name = "traveloka"

    def start_requests(self):
        urls = [
            'https://www.traveloka.com/id-id/hotel/search?spec=15-08-2018.16-08-2018.1.1.HOTEL_GEO.100154.Lampung%20Selatan.1'
        ]
        for url in urls:
            yield scrapy.Request(url, self.parse_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.

```

def parse_hotelList(self, response):
    print('===== start get hotel list data =====')

```



```

options = Options()
options.add_argument('--headless')
options.add_argument('--lang=id')
options.add_argument('--disable-gpu')
dv = webdriver.Chrome('E:/TOKOPEDIA/chromedriver.exe',
    chrome_options=options)
dv.get(response.url)
wait = WebDriverWait(dv,20)
dv.implicitly_wait(7)
listURL = []

wait.until(lambda dv:
dv.find_element_by_xpath("//div[@class='mMmI2 CZtP0 tvat-
searchListItem']"))
    nextPage = True
    while nextPage:
hotels = dv.find_elements_by_xpath("//div[@class='mMmI2 CZtP0
tvat-searchListItem']")
        try:
            for hotel in hotels:
                hotel.click()
                mainwindow = dv.window_handles[0]
                hotelwindow = dv.window_handles[1]
                dv.switch_to_window(hotelwindow)

wait.until(lambda dv:
dv.find_elements_by_xpath("//link[@rel='canonical']"))
hotelLink =
dv.find_element_by_xpath("//link[@rel='canonical']").get_attribute
('href')

                listURL.append(hotelLink)
                dv.close()
                dv.switch_to_window(mainwindow)

nextButton = dv.find_element_by_xpath("//div[@id='next-button']")
dv.execute_script('arguments[0].click();', nextButton)
wait.until(lambda dv:
dv.find_element_by_xpath("//div[@class='mMmI2 CZtP0 tvat-
searchListItem']"))

            except Exception as e:
                print(e)
                nextPage = False

print('===== finish get data LIST
HOTEL =====')
    dv.quit()
    del dv

        for url in listURL:
yield scrapy.Request(url, self.parse_hotelPage)

del listURL

```

*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 mengklik hotel di dalam daftar hotel, halaman hotel akan di tampilkan pada *tab* baru. Setelah itu, proses ekstraksi 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 menghapus *listURL*.

#### 4.4.4. 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.

```
def parse_hotelPage(self, response):
    print('=====  
start hotel page =====')
    options = Options()
    options.add_argument('--headless')
    options.add_argument('--lang=id')
    options.add_argument('--disable-gpu')
    page =webdriver.Chrome('E:/TOKOPEDIA/chromedriver.exe',
        chrome_options=options)
    page.get(response.url)
    wait = WebDriverWait(page,20)
    page.implicitly_wait(7)

    hotellName=page.find_element_by_xpath("//h1[@itemprop='name']").text

    hotelRatingElemen =
    page.find_elements_by_xpath("//div[@class='_3TWbq']")
    if (len(hotelRatingElemen) > 0):
        hotelRating =
        float(hotelRatingElemen[0].text.replace(',','.'))
```

```

else:
    hotelRating = 0

    hotelStarElemen =
    page.find_elements_by_xpath("//meta[@itemprop='ratingValue']")
if (len(hotelStarElemen) > 0):
    hotelStar =
    float(hotelStarElemen[0].get_attribute('content'))
else:
    hotelStar = 0

    hotelAddress =
    page.find_element_by_xpath("//span[@itemprop='streetAddress']").text

try:
    wait.until(lambda page:
    page.find_elements_by_xpath("//div[@itemprop='review']")
    )
    getReview = True

except:
    getReview = False

counter = 0
nextPage = True
while nextPage and getReview:
    reviews
    =page.find_elements_by_xpath("//div[@itemprop='review']")
    )
    hotelReviews = []
    for review in reviews:
        counter += 1
        reviewContent = review.text.split("\n")
        dtContent = reviewContent[2].split(' - ')
        hotelReview = HotelReview()
        hotelReview['hotelName'] = hotelName
        hotelReview['hotelStar'] = hotelStar
        hotelReview['hotelAddress'] = hotelAddress
        hotelReview['hotelRating'] = hotelRating
        hotelReview['reviewRating'] = reviewContent[0]
        hotelReview['reviewName'] = reviewContent[1]
        hotelReview['reviewDate'] =
        self.getReviewDate(dtContent[0])
        if(len(dtContent)>1):
            hotelReview['reviewTheme'] = dtContent[1]
            hotelReview['reviewText'] = reviewContent[3]
            hotelReviews.append(hotelReview)

    for hotelReview in hotelReviews:
        yield hotelReview

    nextButtonElemen =
    page.find_elements_by_xpath("//div[@id='next-button']")
    if(len(nextButtonElemen)>0):

```

```

        page.execute_script('arguments[0].click();',
        nextButtonElemen[0])
        page.implicitly_wait(5)
        wait.until(lambda page:
        page.find_elements_by_xpath("//div[@itemprop='review']")
        )
        else:
            nextPage = False
            del hotelReviews

print('===== end hotel page =====')
page.quit()
del page

```

*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 ekstraksi, sistem akan menutup dan menghapus semua halaman hotel.

```

def getReviewDate(self, dt):
    tanggal = dt.split(" ")

    if(tanggal[1] == "Jan"):
        bl = 1
    elif(tanggal[1] == "Feb"):
        bl = 2
    elif(tanggal[1] == "Mar"):

```

```

        bl = 3
    elif(tanggal[1] == "Apr"):
        bl = 4
    elif(tanggal[1] == "Mei"):
        bl = 5
    elif(tanggal[1] == "Jun"):
        bl = 6
    elif(tanggal[1] == "Jul"):
        bl = 7
    elif(tanggal[1] == "Agu"):
        bl = 8
    elif(tanggal[1] == "Sep"):
        bl = 9
    elif(tanggal[1] == "Okt"):
        bl = 10
    elif(tanggal[1] == "Nov"):
        bl = 11
    elif(tanggal[1] == "Des"):
        bl = 12

    tgl = tanggal[0] + " " + str(bl) + " " + tanggal[2]
    hasil = datetime.strptime(tgl, '%d %m %Y').strftime('%Y-%m-%dT%H:%M:%S%z')
    return hasil

```

*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*.

#### 4.4.5. 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 `elasticsearch` agar data yang bisa disimpan di dalam `elasticsearch`. Angka 500 pada `ScrapyElasticsearch` menunjukkan prioritas untuk proses di `pipelines`. Selanjutnya terdapat konfigurasi `elasticsearch` untuk `server`, `index`, `format index date`, `username`, `password`, `type` dan `buffer length`.

```

class TravelokaPipeline(object):
    def process_item(self, item, spider):
        return item

```

```

ITEM_PIPELINES = {

```

```

        'traveloka.sentiment.ReviewSentimentPolarity':300,
        'scrapyelasticsearch.scrapyelasticsearch.ElasticSearchPipeline': 500
    }

ELASTICSEARCH_SERVERS = ['139.162.60.42']
ELASTICSEARCH_INDEX = 'reviewHotel'
ELASTICSEARCH_INDEX_DATE_FORMAT = '%Y-%m'
ELASTICSEARCH_USERNAME = 'elastic'
ELASTICSEARCH_PASSWORD = 'TeloGosong'
ELASTICSEARCH_TYPE = 'items'
ELASTICSEARCH_BUFFER_LENGTH = 50

```

## 4.5. Pengujian

Pengujian sistem scraping dan crawling dibagi menjadi dua tahap yaitu pengujian fungsional dan pengujian proses. Pengujian fungsional bertujuan untuk dapat mengetahui apakah *method* yang dibuat pada *spider* dapat berfungsi dengan baik untuk melakukan *scraping* dan *crawling*. Sedangkan pengujian proses adalah untuk menguji keseluruhan proses hingga mendapatkan hasil data terstruktur sesuai yang diharapkan.

### 4.5.1. 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. Pada proses *debugging* dilakukan pengamatan pada variabel-variabel yang digunakan untuk menampung data hasil *scraping*. Selain itu juga dilakukan pengamatan pada proses pemanggilan *method callback* yang dibuat di *class spider*.

The image shows a Python IDE with a Selenium script for scraping hotel URLs. The 'VARIABLES' pane on the left shows a list named 'listURL' containing two URLs from traveloka.com. The script code on the right shows the logic for finding and extracting these URLs. The terminal at the bottom shows debug logs for the Selenium session.

```

48 wait.until(lambda dv: dv.find_element_by_xpath("//div[@class='mM12 C2tP0 tvat-searchListItem']"))
49 nextPage = True
50 while nextPage:
51     #ambil semua element hotel 1
52     hotels = dv.find_elements_by_xpath("//div[@class='mM12 C2tP0 tvat-searchListItem']")
53     try:
54         for hotel in hotels:
55             hotel.click()
56             mainWindow = dv.window_handles[0] #tab bawaan browser
57             hotelWindow = dv.window_handles[1] #new tab
58             dv.switch_to_window(hotelWindow)
59             dv.wait.until(lambda dv: dv.find_elements_by_xpath("//link[@rel='canonical']"))
60             hotelLink = dv.find_element_by_xpath("//link[@rel='canonical']").get_attribute('href')
61             listURL.append(hotelLink)
62             dv.close()
63             dv.switch_to_window(mainWindow)
64
65             nextButton = dv.find_element_by_xpath("//div[@id='next-button']")
66             dv.execute_script('arguments[0].click();', nextButton)
67             wait.until(lambda dv: dv.find_element_by_xpath("//div[@class='mM12 C2tP0 tvat-searchListItem']"))
68
69         except Exception as e:
70             print(e)
71             nextPage = False
72
73     print('----- finish get data LIST HOTEL -----')
74     dv.quit()
75     del dv
76
77 for url in listURL:

```

The terminal output shows the following logs:

```

w ("sessionId": "fd781be6ed64fd2fed8a8c83a0ca7b6")
[0906/112500.315:WARNING:spdy_session.cc(3058)] Received HEADERS for invalid stream 9
2018-09-06 11:25:00 [selenium.webdriver.remote.remote_connection] DEBUG: b'{"sessionId": "fd781be6ed64fd2fed8a8c83a0ca7b6", "status": "Finished Request"}'
2018-09-06 11:25:01 [selenium.webdriver.remote.remote_connection] DEBUG: POST http://127.0.0.1:2173/session/fd781be6ed64fd2fed8a8c83a0ca7b6/
{"name": "CDwindow-296D0877F83C738FCDS2B957D4805481", "sessionId": "fd781be6ed64fd2fed8a8c83a0ca7b6"}
2018-09-06 11:25:01 [selenium.webdriver.remote.remote_connection] DEBUG: b'{"sessionId": "fd781be6ed64fd2fed8a8c83a0ca7b6", "status": "Finished Request"}'
2018-09-06 11:25:01 [selenium.webdriver.remote.remote_connection] DEBUG: Finished Request

```

Gambar 4-21. Pengujian ekstraksi *list url* hotel

Pada Gambar 4-21 menunjukkan hasil *list url* hotel yang sudah berhasil diekstraksi. Proses *scraping* dan *crawling list url* hotel dilakukan pada *method HotelList*. *List url* hotel yang berhasil di ekstraksi akan dimasukkan kedalam *list* yaitu *listURL*. *List url* hotel yang sudah diekstraksi dapat dilihat dalam *listURL* pada bagian *variables*.

Selanjutnya pengujian untuk ekstraksi data hotel juga dilakukan dengan cara *debugging*. Proses ekstraksi data hotel dilakukan pada *method HotelPage*. *List url* yang sudah di ekstraksi akan dibuka oleh *browser* setelah itu ekstraksi pada data hotel dilakukan oleh sistem. Pada Gambar 4-22 di dalam *local* terlihat hasil data hotel yang telah berhasil diekstraksi.

```

85 print('===== start hotel page =====')
86 options = Options()
87 options.add_argument('--headless')
88 options.add_argument('--lang-id')
89 options.add_argument('--disable-gpu')
90 page = webdriver.Chrome('E:/TOKOPEDIA/chromedriver.exe', chrome_options=options)
91 page.get(response.url)
92 wait = WebDriverWait(page,20)
93 page.implicitly_wait(?)
94
95 hotelName = page.find_element_by_xpath("//h1[@itemprop='name']").text
96
97 hotelRatingElemen = page.find_elements_by_xpath("//div[@class='_3Thbq']")
98 if (len(hotelRatingElemen) > 0):
99     hotelRating = float(hotelRatingElemen[0].text.replace(',','.'))
100 else:
101     hotelRating = 0
102
103 hotelStarElemen = page.find_elements_by_xpath("//meta[@itemprop='ratingValue']")
104 if (len(hotelStarElemen) > 0):
105     hotelStar = float(hotelStarElemen[0].get_attribute('content'))
106 else:
107     hotelStar = 0
108
109 hotelAddress = page.find_element_by_xpath("//span[@itemprop='streetAddress']").text
110
111 try:
112     wait.until(lambda page: page.find_elements_by_xpath("//div[@itemprop='review']"))
113     getReview = True
114 except:

```

**VARIABLES**

- Local
  - options: <selenium.webdriver.chrome.options.Options o...
  - page: <selenium.webdriver.chrome.webdriver.WebDriver ...
  - wait: <selenium.webdriver.support.wait.WebDriverWait ...
  - hotelName: 'Airy Krakatau Kahai Beach Batu Balak 99 ...'
  - hotelRatingElemen: [<selenium.webdriver....18639-2?>]
  - hotelRating: 7.9
  - hotelStarElemen: [<selenium.webdriver....18639-3?>]
  - hotelStar: 4.8
  - hotelAddress: 'Jl Raya Pesisir Desa Batu Balak No. 99 ...'
  - getReview: <undefined>
  - counter: <undefined>
  - nextPage: <undefined>
  - reviews: <undefined>
  - hotelReviews: <undefined>
  - review: <undefined>
  - reviewContent: <undefined>
  - dtContent: <undefined>
  - hotelReview: <undefined>
  - nextbuttonElemen: <undefined>
  - Options: <class 'selenium.webdriver.chrome.options.op...
  - webdriver: <module 'selenium.webdriver' from 'C:\Pyt...
  - WebDriverWait: <class 'selenium.webdriver.support.wai...
  - HotelReview: <ItemMeta>

**DEBUG CONSOLE**

```

2018-09-06 11:35:43 [selenium.webdriver.remote.remote_connection] DEBUG: b'({"sessionId":"25208ec0d21ee7cec2bce1f267573092","":
"Jl Raya Pesisir Desa Batu Balak No. 99 Kec. Rajabasa, Rajabasa, Lampung Selatan, Provinsi Lampung, Indonesia, 35551"})'
2018-09-06 11:35:43 [selenium.webdriver.remote.remote_connection] DEBUG: Finished Request
2018-09-06 11:35:43 [selenium.webdriver.remote.remote_connection] DEBUG: POST http://127.0.0.1:2473/session/25208ec0d21ee7cec2
lements ("using": "xpath", "value": "//div[@itemprop='review']", "sessionId": "25208ec0d21ee7cec2bce1f267573092")
2018-09-06 11:35:52 [selenium.webdriver.remote.remote_connection] DEBUG: b'({"sessionId":"25208ec0d21ee7cec2bce1f267573092","
":["ELEMENT":"0.5699652575518639-15"],["ELEMENT":"0.5699652575518639-16"],["ELEMENT":"0.5699652575518639-17"],["ELEMENT":"0.56
"],["ELEMENT":"0.5699652575518639-19"],["ELEMENT":"0.5699652575518639-20"],["ELEMENT":"0.5699652575518639-21"],["ELEMENT":"0.56
22"],["ELEMENT":"0.5699652575518639-23"],["ELEMENT":"0.5699652575518639-24"]])'
2018-09-06 11:35:52 [selenium.webdriver.remote.remote_connection] DEBUG: Finished Request

```

Gambar 4-22. 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. Proses ekstraksi *review* pada halaman hotel terjadi pada halaman yang sama pada setiap data hotel yang diekstraksi.

Pada Gambar 4-23 dapat dilihat hasil pengujian untuk proses ekstraksi *review* hotel. Pada bagian *variables*, didalam *review content* adalah data *review* yang telah berhasil diekstraksi berupa nama, tanggal, tema, rating dan teks review.



```

118 hotelName: 'Airy Krakatau Kahai Beach Batu Balak 99
119 hotelRatingElemen: [<selenium.webdriver....18639-2*>]
120 hotelRating: 7.9
121 hotelStarElemen: [<selenium.webdriver....18639-3*>],
122 hotelStar: 4.8
123 hotelAddress: 'Jl Raya Pesisir Desa Batu Balak No. 99.
124 getReview: True
125 counter: 2
126 nextPage: True
127 reviews: [<selenium.webdriver....8639-15*>], <seleniu
128 hotelReviews: [{'hotelAddress': 'Jl...Keluarga'}, {'h
129 review: <selenium.webdriver.remote.webelement.WebElem
130 reviewContent: ['6,2', 'Rheza S.', '23 Jun 2018 - Lib
131 [0]: '6,2'
132 [1]: 'Rheza S.'
133 [2]: '23 Jun 2018 - Liburan Keluarga'
134 [3]: '1. Kebersihan kamar mandi, lingkungan pantai,
135 dtContent: ['23 Jun 2018', 'Liburan Keluarga']
136 hotelReview: {'hotelAddress': 'Jl Raya Pesisir Desa B.
137 nextButtonElemen: <undefined>
138 Options: <class 'selenium.webdriver.chrome.options.op
139 webdriver: <module 'selenium.webdriver' from 'C:\Pyt
140 webdriverwait: <class 'selenium.webdriver.support.wai
141 HotelReview: <ItemMeta>
142 Arguments
143 WATCH
144 CALL STACK
145 BREAKPOINTS
146 All Exceptions
147 Uncaught Exceptions
148 traveioka_spider.py traveioka/spiders 95
149 traveioka_spider.py traveioka/spiders 118
150 traveioka_spider.py traveioka/spiders 119
151 traveioka_spider.py traveioka/spiders 144
152 traveioka_spider.py traveioka/spiders 145
153 traveioka_spider.py traveioka/spiders 148
154 traveioka_spider.py traveioka/spiders 162
155 Scrapy (traveioka) Python 3.6 (32-bit)

```

```

118 counter = 0
119 nextPage = True
120 while nextPage and getReview:
121     reviews = page.find_elements_by_xpath("//div[@itemprop='review']")
122     hotelReviews = []
123     for review in reviews:
124         counter += 1
125         reviewContent = review.text.split("\n")
126         dtContent = reviewContent[2].split(' - ')
127         #object item
128         hotelReview = HotelReview()
129         hotelReview['hotelName'] = hotelName
130         hotelReview['hotelStar'] = hotelStar
131         hotelReview['hotelAddress'] = hotelAddress
132         hotelReview['hotelRating'] = hotelRating
133         hotelReview['reviewRating'] = reviewContent[0]
134         hotelReview['reviewName'] = reviewContent[1]
135         hotelReview['reviewDate'] = self.getReviewDate(dtContent[0])
136         if(len(dtContent)>1):
137             hotelReview['reviewTheme'] = dtContent[1]
138             hotelReview['reviewText'] = reviewContent[3]
139             hotelReviews.append(hotelReview)
140     for hotelReview in hotelReviews:
141         yield hotelReview
142     nextButtonElemen = page.find_elements_by_xpath("//div[@id='next-button']")
143     if(len(nextButtonElemen)>0):
144         page.execute_script('arguments[0].click();', nextButtonElemen[0])
145         page.implicitly_wait(5)
146         nextButtonElemen = page.find_elements_by_xpath("//div[@id='next-button']")
147         if(len(nextButtonElemen)>0):
148             page.execute_script('arguments[0].click();', nextButtonElemen[0])
149             page.implicitly_wait(5)
150         else:
151             nextPage = False
152             del hotelReviews
153     print('==== end hotel page =====')
154     page.quit()
155     del page
156 def getReviewDate(self, dt):
157     tanggal = dt.split(" ")
158     if(tanggal[1] == "Jan"):

```

Gambar 4-23. Pengujian ekstraksi review hotel

```

131 hotelReview['hotelAddress'] = hotelAddress
132 hotelReview['hotelRating'] = hotelRating
133 hotelReview['reviewRating'] = reviewContent[0]
134 hotelReview['reviewName'] = reviewContent[1]
135 hotelReview['reviewDate'] = self.getReviewDate(dtContent[0])
136 if(len(dtContent)>1):
137     hotelReview['reviewTheme'] = dtContent[1]
138     hotelReview['reviewText'] = reviewContent[3]
139     hotelReviews.append(hotelReview)
140 for hotelReview in hotelReviews:
141     yield hotelReview
142 nextButtonElemen = page.find_elements_by_xpath("//div[@id='next-button']")
143 if(len(nextButtonElemen)>0):
144     page.execute_script('arguments[0].click();', nextButtonElemen[0])
145     page.implicitly_wait(5)
146     nextButtonElemen = page.find_elements_by_xpath("//div[@id='next-button']")
147 else:
148     nextPage = False
149     del hotelReviews
150 print('==== end hotel page =====')
151 page.quit()
152 del page
153 def getReviewDate(self, dt):
154     tanggal = dt.split(" ")
155     if(tanggal[1] == "Jan"):
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
```

hotel. Jika terdapat *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 4-25.

```

132 hotelReview['hotelRating'] = hotelRating
133 hotelReview['reviewRating'] = reviewContent[0]
134 hotelReview['reviewName'] = reviewContent[1]
135 hotelReview['reviewDate'] = self.getReviewDate(dtContent[0])
136 if(len(dtContent)>1):
137     hotelReview['reviewTheme'] = dtContent[1]
138     hotelReview['reviewText'] = reviewContent[3]
139     hotelReviews.append(hotelReview)
140
141 for hotelReview in hotelReviews:
142     yield hotelReview
143
144 nextButtonElemen = page.find_elements_by_xpath("//div[@id='next-button']")
145 if(len(nextButtonElemen)>0):
146     page.execute_script('arguments[0].click();', nextButtonElemen[0])
147     page.implicitly_wait(5)
148     wait.until(lambda page: page.find_elements_by_xpath("//div[@itemprop='review']"))
149 else:
150     nextPage = False
151     del hotelReviews
152
153 print('==== end hotel page =====')
154 page.quit()
155 del page
156
157 def getReviewDate(self, dt):
158     tanggal = dt.split(" ")
159     if(tanggal[1] == "Jan"):
160
161
162
163
164
165
166
167
168
169
170
171

```

DEBUG | Scrapy (traveioka) | traveioka\_spider.py x | Python Debug Conso

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

'reviewRating': '6,9',
'reviewSentimentPolarity': 0,
'reviewText': 'Pemandangannya bagus, jalan menuju hotel tidak bagus dan
'gelap, merasa tidak aman',
'reviewTheme': 'Liburan Romantis'
2018-09-06 11:42:02 [selenium.webdriver.remote.remote_connection] DEBUG: POST http://127.0.0.1:2473/session/25208ec0d21ee7ce
lements ("using": "xpath", "value": "//div[@id='next-button']", "sessionId": "25208ec0d21ee7cec2bce1f267573092")
2018-09-06 11:42:08 [selenium.webdriver.remote.remote_connection] DEBUG: b{"sessionId":"25208ec0d21ee7cec2bce1f267573092",":
[]}'
2018-09-06 11:42:08 [selenium.webdriver.remote.remote_connection] DEBUG: Finished Request

```

Scrapy (traveioka) Python 3.6 (32-bit) Ln 151, Col 1 Spaces: 4 UTF-8

Gambar 4-25. Pengujian *crawling* 2

#### 4.5.2. 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*.

Name	CPU	Memory	Disk	Network
Google Chrome	0%	1.3 MB	0 MB/s	0 Mbps
chromedriver.exe (32 bit)	0%	3.7 MB	0 MB/s	0 Mbps
Google Chrome	0%	74.7 MB	0 MB/s	0 Mbps
Google Chrome	0%	4.7 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.6 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.5 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.6 MB	0 MB/s	0 Mbps
chromedriver.exe (32 bit)	0%	3.8 MB	0 MB/s	0 Mbps
Google Chrome	3.2%	88.3 MB	0 MB/s	0 Mbps
Google Chrome	0%	4.7 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.6 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.5 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.6 MB	0 MB/s	0 Mbps
chromedriver.exe (32 bit)	0%	3.7 MB	0 MB/s	0 Mbps
Google Chrome	4.0%	109.1 MB	0 MB/s	0 Mbps
Google Chrome	0%	4.7 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.6 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.5 MB	0 MB/s	0 Mbps
Google Chrome	0%	1.6 MB	0 MB/s	0 Mbps
chromedriver.exe (32 bit)	0.4%	4.3 MB	0 MB/s	0 Mbps

Gambar 4-26. Pengujian proses *concurrent*

Pengujian pada *setting concurrent* dapat dilihat pada *task manager* pada Gambar 4-26 ketika sistem dijalankan. *Concurrent* request yang diatur pada *setting.py* adalah 8, yaitu menunjukkan 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.Time
outError': 1,
   'downloader/exception_type_count/twisted.web._newclient.Resp
onseNeverReceived': 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.ResponseNeverRece
ived': 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)

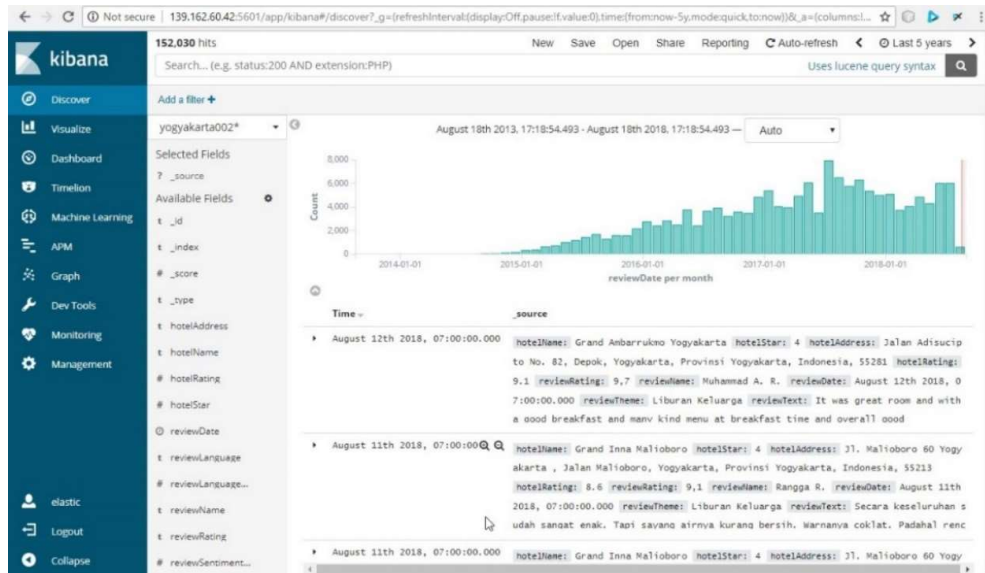
```

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.

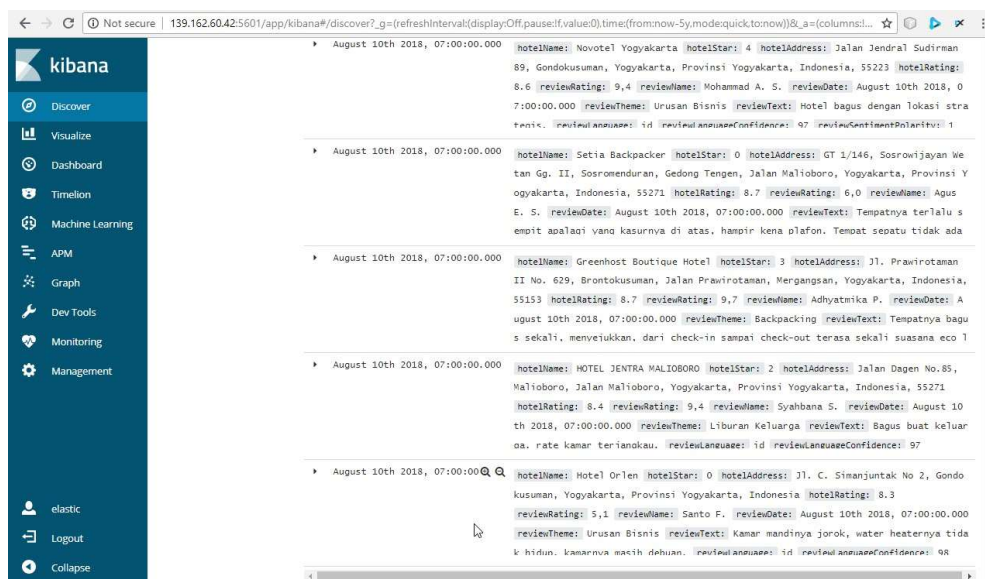
#### 4.6. Hasil

Data hasil ekstraksi akan disimpan pada *elasticsearch* melalui *pipelines*. Gambar 4-27 dan Gambar 4-28 adalah hasil data yang sudah berada pada

*elasticsearch*. Semua data yang didapat dari hasil ekstraksi akan masuk dalam *index elasticsearch* melalui *items pipelines*. Data yang diekstraksi adalah data hotel (nama, alamat, *rating* dan bintang hotel) dan *review* hotel (nama, tanggal, tema, *rating*, teks *review*) yang ada di Yogyakarta. Dibagian kiri atas halaman terdapat jumlah semua data yang telah diekstraksi.

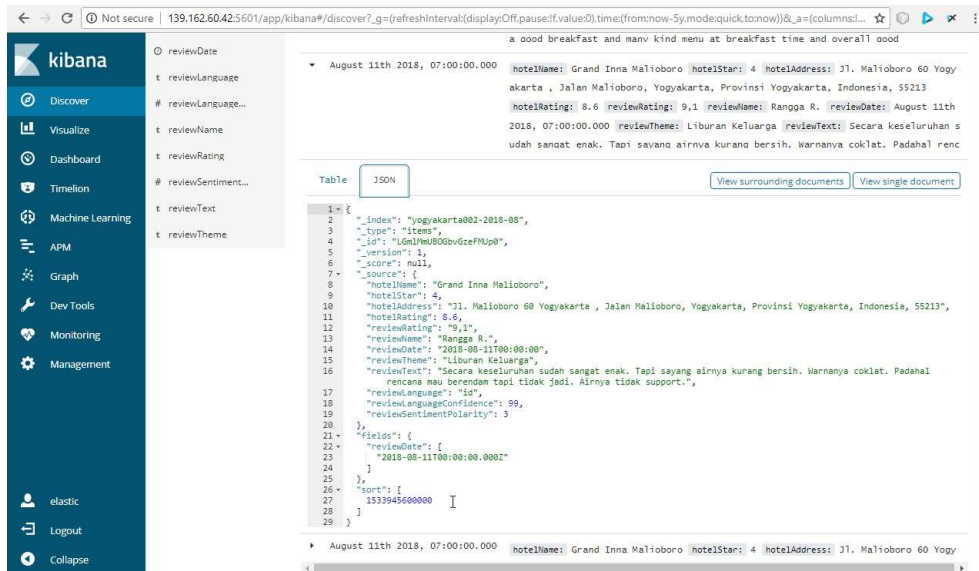


Gambar 4-27. Hasil ekstraksi 1



Gambar 4-28. Hasil ekstraksi 2

Gambar 4-28 merupakan tampilan data yang ada di *elasticsearch*. Data ditampilkan berdasarkan *review* hotel. Masing-masing data *review* hotel dapat dibuka dan akan menampilkan data json sesuai isi dari data *review* tersebut. Data yang disimpan di dalam *elasticsearch* telah berhasil disimpan berdasarkan model data yang telah dibuat.



The screenshot shows the Kibana interface with a search query: "a ood breakfast and manv kind menu at breakfast time and overall ood". The search results are displayed in a table view, showing a single document for "Grand Inna Malioboro" with a rating of 4 stars. The document is displayed in JSON format, showing fields like "\_index", "\_type", "\_id", "\_score", "\_source", and "reviewText".

```

1 {
2   "_index": "yogyakarta002-2018-08",
3   "_type": "items",
4   "_id": "L6nJmU00DvGzefNU0p",
5   "_version": 1,
6   "_score": null,
7   "_source": {
8     "hotelName": "Grand Inna Malioboro",
9     "hotelStar": 4,
10    "hotelAddress": "Jl. Malioboro 60 Yogyakarta , Jalan Malioboro, Yogyakarta, Provinsi Yogyakarta, Indonesia, 55213",
11    "hotelRating": 8.6,
12    "reviewRating": "9,1",
13    "reviewName": "Rangga R.",
14    "reviewDate": "2018-08-11T00:00:00",
15    "reviewTheme": "Liburan Keluarga",
16    "reviewText": "Secara keseluruhan sudah sangat enak. Tapi sayang airnya kurang bersih. Warnanya coklat. Padahal rencana mau brendan tapi tidak jadi. Airnya tidak support.",
17    "reviewLanguage": "id",
18    "reviewLanguageConfidence": 99,
19    "reviewSentimentPolarity": 3
20  },
21  "fields": {
22    "reviewDate": [
23      "2018-08-11T00:00:00.000Z"
24    ]
25  },
26  "sort": [
27    [
28      1533945600000
29    ]
30  ]
31 }

```

Gambar 4-29. Hasil ekstraksi (*json*)

Masing-masing data *review* hotel yang ada pada *elasticsearch* dapat dilihat dalam *format json* seperti pada Gambar 4-29. Data hasil ekstraksi yang ditampilkan di *elasticsearch* merupakan data yang terstruktur dengan model data sesuai dengan yang telah didefinisikan pada *class item*.