BAB III
PENGEMBANGAN SISTEM

3.1. Metodelogi Pengembangan Sistem

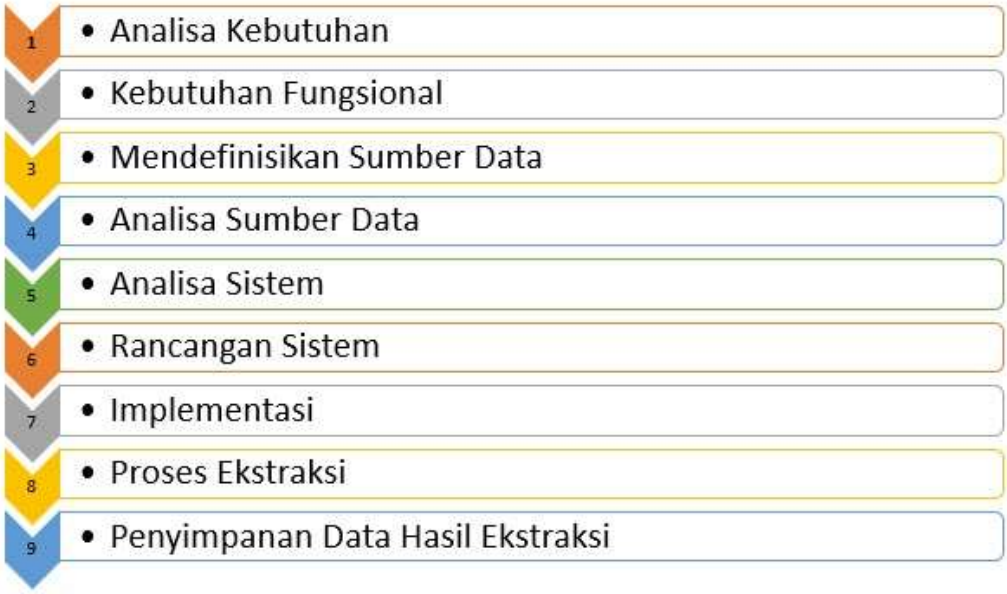
Pengembangan sistem *scraping* dan *crawling* dilakukan melalui beberapa tahapan. Tahapan-tahapan digunakan dalam pengembangan sistem *scraping* dan *crawling* ini mengacu pada journal *Conceptual Approach for Development of Web Scraping Application for Tracking Information* (Plamen Milev, 2017).

Langkah-langkah dalam metodelogi pengembangan sistem adalah sebagai berikut :

1. Analisa Kebutuhan. Tahap ini adalah tahap untuk melakukan evaluasi terhadap kebutuhan pengguna terkait dengan sistem yang akan dikembangkan.
2. Kebutuhan Fungsional. Menentukan fungsi-fungsi sistem yang dibutuhkan untuk memenuhi kebutuhan pengguna yang telah di evaluasi.
3. Mendefinisikan Sumber Data. Pada tahap ini akan ditentukan sumber data yang dibutuhkan oleh pengguna.
4. Analisa Sumber Data. Tahap ini merupakan tahap untuk menganalisa sumber data yang akan diekstraksi.
5. Analisa Sistem. Analasis sistem dibutuhkan untuk menentukan kebutuhan sistem dan menentukan modul-modul yang akan digunakan dalam sistem.
6. *Design Sistem*. *Design sistem* dibuat untuk memenuhi seluruh kebutuhan dan proses yang akan dibuat dalam sistem yang akan dibuat.
7. Implementasi dan pengujian. Tahap implementasi merupakan penjelasan dari proses implementasi yang dilakukan dalam pengembangan sistem. Setelah itu dilakukan pengujian sistem dengan cara *debugging*.

8. Proses Ekstraksi. Tahap ini menjelasakn bagaimana sistem melakukan proses ekstraksi pada sebuah *website.*
9. Penyimpanan Data Hasil Ekstraksi. Setelah proses ekstrkasi selesai, data hasil ekstraksi akan dikirim dan disimpan dalam *elasticsearch*.

Gambaran proses tahapan-tahapan pada pengembangan sistem *scraping* dan *crawling* dapat dilihat pada Gambar 3-1.



Gambar 3-1. Tahapan pengembangan sistem *scraping* dsn *crawling*

3.2. Analisa Kebutuhan

Sistem *scraping* dan *crawling* dikembangkan untuk memnuhi kebutuhan data oleh pengguna. Terkait dengan hal tersebut, maka kebutuhan pengguna adalah::

1. Pengguna membutuhkan sebuah sistem yang dapat mengekstrasksi data yang terdapat pada *website*
2. Pengguna membutuhkan sebuah sistem yang dapat melakukan *crawling website* berbasis *ajax*.
3. Data yang dibutuhkan pengguna merupakan data yang terstruktur

4. Data yang dibutuhkan pengguna merupakan data dari banyak halaman *website*
5. Data yang dibutuhkan merupakan data yang sudah tersimpan didalam database.

Dalam pengembangan sistem scraping dan crawling ini, data dari *webiste* yang dibutuhkan pengguna ada beberapa bagian. Bagian nama, alamat, *rating* dan star hotel dibutuhkan untuk mengidentifikasi data hotel dari setiap *review*. Sedangkan *rating*, nama, tanggal, tema dan teks *review* dibutuhkan untuk kepentingan analisis, grafik data dan yang lainnya.

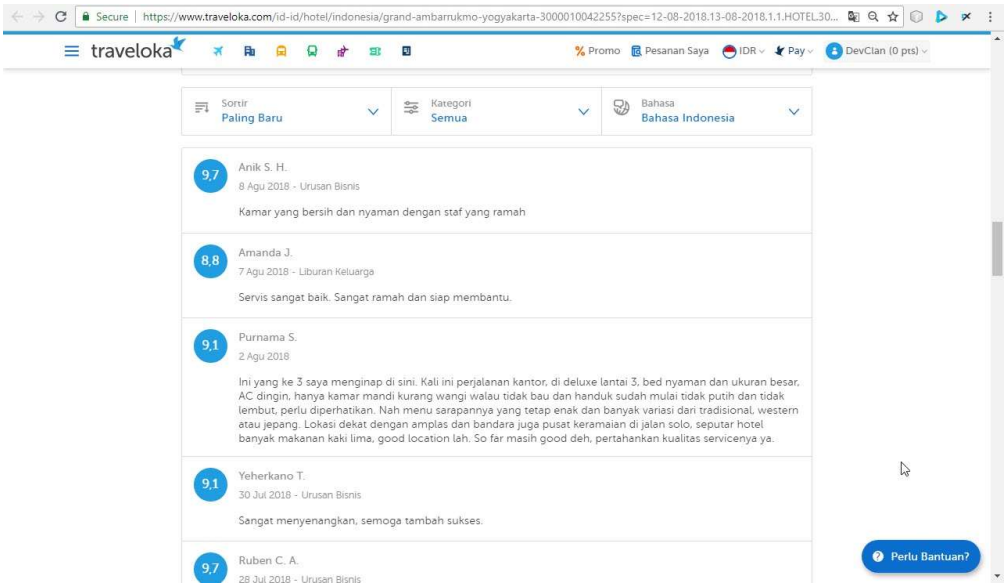
3.3. Kebutuhan Fungsional

Kebutuhan fungsional di tentukan berdasarkan analisa kebutuhan pengguna. Setelah menentukan apa saja yang dibutuhkan oleh pengguna, maka ditentukan kebutuhan fungsional. Kebutuhan fungsional dari sistem *scraping* dan *crawling* yang akan dikembangkan adalah:

1. Sistem dapat mengekstraksi data teks dari sebuah *website.*
2. Sistem dapat melakukan *crawling* pada *website* berbasis *ajax.*
3. Hasil data ekstraksi harus berupa data terstruktur.
4. Sistem dapat melakukan *crawling* untuk mengambil data secara pararel.
5. Data hasil ekstraksi dapat disimpan dalam sebuah database.

3.4. Mendefinisikan Sumber Data

Sumber data yang digunakan adalah data yang terdapat pada *website* Traveloka. Data yang diambil adalah data *review* semua hotel yang terdapat pada *website* traveloka dan data tersebut berupa data teks. Selain data *review* hotel, data lain yang diekstraksi adalah data informasi hotel yaitu: nama hotel, alamat hotel, bintang hotel dan rating hotel.



Gambar 3-2. Halaman *review* Traveloka

Tampilan *review* hotel pada *webiste* Traveloka dapat dilihat pada Gambar 3-2.Pada bagian *review* hotel, data yang akan diekstraksil berupa nama , tanggal, rating, tema dan teks *review*. Data *review* hotel yang ada di masing-masing halaman hotel akan diambil semuanya..

3.5. Analisa Sumber Data

Setelah mendefinisikan sumber data maka dilakukan analisis terhadap sumber data yang akan diekstraksi. Data yang akan diekstraksi terdapat pada teks *HTML* halaman *website* Traveloka. Namun, hanya beberapa bagian data teks yang akan diekstraksi dari satu halaman *website.* Maka, untuk mengekstraksi data-data yang diinginkan terlebih dahulu harus dipilih elemen-elemen html dimana terdapat data yang akan diekstraksi. Untuk memilih elemen-elemen tersebut dibutuhkan sebuah *selector*. *Selector* yang akan digunakan adalah *Xpath selector*.

Website traveloka menggunakan *javascript* dan *ajax*. *Ajax* digunakan pada halaman daftar hotel dan halaman *review* hotel. *Ajax* digunakan untuk perpindahan halaman daftar hotel dan daftar *review*. Oleh karena itu proses *scraping* dan *crawling* harus dilakukan menggunakan *driver browser* yang dapat mendukung *javascrip* dan *ajax*.

3.6. Analisa Sistem

Berdasarkan analisa kebutuhan dan analisa sumber data maka diputuskan untuk mengembangkan teknik *scraping* dan *crawling* untuk melakukan ekstraksi data pada *website* yang menggunakan *javascript* dan *ajax*. Web *Scraping* adalah teknik yang digunakan untuk mengambil data berupa teks. Sedangkan *crawling* merupakan suatu teknik yang digunakan untuk membuka *link* secara berulang atau terus menerus pada halaman-halaman web. Sebelum melakukan *scraping*, harus ditentukan terlebih dahulu target *website* yang akan di *scraping*. *Website* Traveloka merupakan target *scraping* untuk saat ini. Data yang akan diambil atau di *scraping* adalah data teks *review* hotel yang ada pada Traveloka diambil atau di *scraping* adalah data teks *review* hotel yang ada pada Traveloka.

Untuk melakukan *scraping* dan *crawling* akan menggunakan *Scrapy*. *Scrapy* adalah sebuah *framework* yang digunakan untuk melalukan *scraping* dan *crawling.* Proses *scraping* dan *crawling* dapat dilakukan secara bersamaan dengan menggunakan *framework* ini.

Pada *website* Traveloka terdapat *AJAX* yang terletak pada bagian daftar pencarian hotel dan *review* hotel. Sedangkan *downloader* pada *scrapy* tidak dapat memproses *AJAX*, karena itu *selenium driver* dibutuhkan pada proses *scraping* ini. *Selenium* digunakan untuk mengontrol *web driver* dan juga dapat memproses *AJAX.*

Dalam kasus ini, *scraping* tidak menggambil semua data yang ada dalam web, melainkan hanya beberapa elemen yang dibutuhkan terkait *review* hotel. Maka elemen-elemen yang akan diambil harus menggunakan sebuah *selector. Selector* yang digunakan untuk memilih elemen-elemen yang dibutuhkan adalah *Xpath*.

Setelah semua data yang diperlukan diekstraksi, maka dibutuhkan tempat untuk menyimpan semua data tersebut. *Elasticseach* adalah tempat akhir dimana data hasil *scraping* akan disimpan atau diproses lebih lanjut. Data yang telah di *scraping* akan masuk pada *item pipeline* dan disalurkan ke *elasticsearch*.

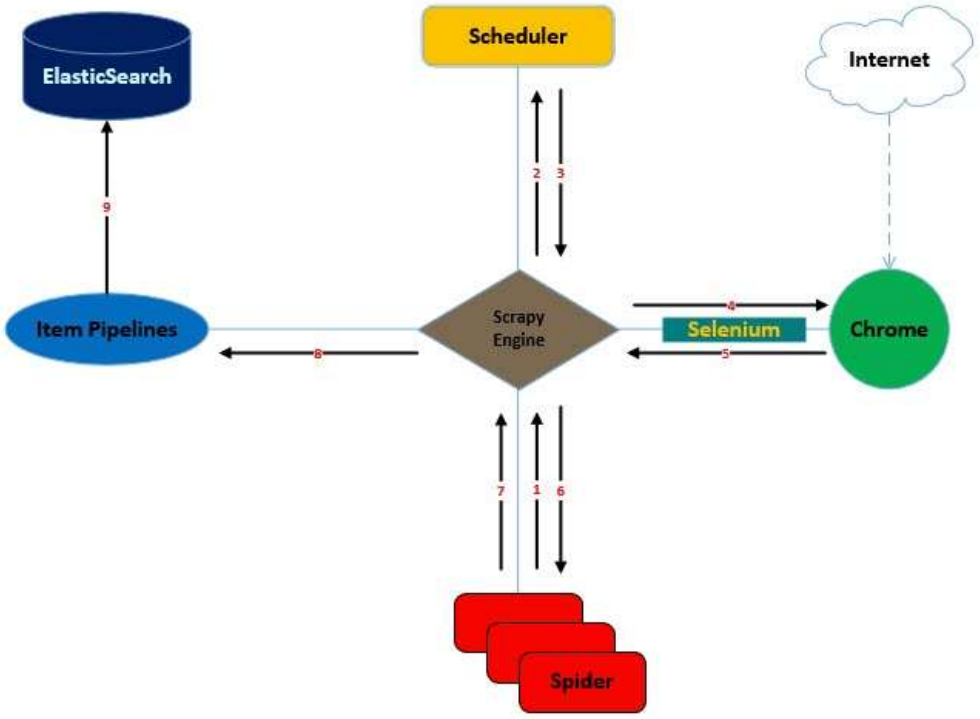
3.7. Rancangan Sistem

Scrapy merupakan *framework* utama dari pengembangan sistem *scraping* ini. Pertimbangan pemilihan *scrapy* adalah kemampuan *scrapy* untuk mengelola prosees secara paralel dimana dalam *scrapy* mempunya fitur *scheduler* dan *scrapy* juga dapat dikembangkan secara modular sehingga dapat diintegrasikan dengan modul modul lain yg dibutuhkan. Oleh karena itu beberapa kebutuhan yang tidak dapat dipenuhi oleh fitur scrapy dipenuhi dengan mengintegrasikan *scrapy* dengan beberapa modul yang lain.

Modul *chromedriver* digunakan untuk dapat mengunakan *browser chrome* yang dapat men-*support javascript* dan *ajax*. Agar *chromedriver* dapat berinterasi dengan *scrapy* maka diperlukan integrasi *scrapy* dengan *selenium*. *Selenium* ditempatkan sebagai modul *downloader* pada *scrapy*.

Pada bagian penyimpanan data hasil *scrapy* dipilih *elasticsearch*. *Elastiksearch* dipilih agar dapat menyimpan data hasil *scraping* dalam bentuk dokumen *json* dan sekaligus melakukan *indexing* data yg disimpan. Agar *elastissearch* dapat dihubingkan dengan *scrapy* maka digunakan modul *scrapyelasticseach*.

Dari berbagai modul yang digunakan diatas, dapat digambarkan rancangan arsitektur sistem *scraping* yang akan dikembangkan sebagaimana digambarkan pada Gambar 3-3.



Gambar 3-3. Arsitektur sistem

Penjelasan dari Gambar 3-3 adalah sebagai berikut :

a. *Scrapy Engine* dan *Scheduler* merupakan komponen yang terdapat pada *scrapy* yang berfungsi mengendalikan seluruh proses *scraping* dan *crawling*.
b. *Spider* adalah *class* yang berisi *code* yang nantinya akan melakukan proses *scraping* dan *crawling*
c. *Chrome* adalah *web browser* yang digunakan sebagai *downloader*.
d. *Selenium* digunakan untuk menghubungkan *scrapy* dan *web browser*.
e. *Item pipeline* menjadi perantara data yang dihasilkan untuk kemudian di simpan pada *elasticsearch*.
f. *Elasticsearch* adalah tempat menyimpan data hasil *scraping*.

3.8. Implementasi dan Pengujian

Pada tahap implementasi dilakukan proses pengembangan sistem berdasarkan arsitektur yang telah ditentukan. Sistem *scraping* diimplementasikan menggunakan bahasa *python*. *Framework* yang digunakan untuk melakukan *scraping* adalah *Scrapy*. Pengembangan proses *scraping* dan *crawling* dilakukan pada modul *spider*. Selain menggunakan *scrapy*, pengembangan sistem ini juga menggunakan *selenium driver* untuk dapat memproses *ajax* di dalam *website* traveloka. *Selector* yang digunakan untuk memilih elemen pada teks *HTML* yang akan diekstraksi adalah *Xptah selector*. Dan data hasil ekstraksi akan disimpan didalam *elasticsearch*.

Proses pengujian sistem ini dilakukan di dalam *spider* ketika sistem dijalankan. Penguji akan melakukan *debugging* untuk mengetahui apakah sistem berjalan dengan baik, dan memastikan data yang diekstraksi dapat terekstraksi dengan baik.

3.9. Proses Ekstraksi

Ekstraksi data adalah proses menjalankan *scrapy framework* yang telah dikembangkan untuk mengambil semua data *review* dari Traveloka. *Scrapy* mempunya fitur *concurency*, sehingga proses *download* halaman-halaman web dapat dilakukan secara paralel. Pada tahap ini diatur berapa jumlah *concurrent process download*. Pengaturan ini dengan memperhatikan kemampuan komputer dimana *scrapy* dijalankan, terutama kebutuhan *processor* dan memory-nya.

Data-data yang diekstrak dimasukan dalam object *item*. *Item-item* hasil ekstrasi akan diproses lebih lanjut pada modul modul yg ada pada *pipeline scrapy.*

3.10. Penyimpanan Data Hasil Ekstraksi

Untuk menyimpan data hasil ekstraksi dipilih menggunakan *elasticsearch*. *Elasticseach* adalah *noSQL database* atau tempat penyimpanan data dalam bentuk dokumen (CodePolitan, 2016). Pertimbangan pemilihan *elasticsearch* adalah data dapat disimpan dalam bentuk dokumen format *JSON* dengan skema yang fleksibel. Selain itu elascticsearch mempunyai *search engine* untuk mencari data data yang telah disimpan. Oleh karena itu penggunaan *elasticsearch* dapat mempermudah proses penyimpanan dan evaluasi data hasil ekstraksi.