IMPLEMENTASI WEBSCRAPING UNTUK PENCARIAN PRODUK DENGAN METODE EXTENSIBLE PATH LANGUAGE (XPATH) PADA HALAMAN WEBSITE E-COMMERCE

Rifqi Rosyidi[1)]
[1]Sistem Informasi, STMIK PPKIA Pradnya Paramita (Rifqi Rosyidi)
Email: rief.rosyidi@gmail.com

Abstract

Information Overload is the condition wich too much amount of information received by humans, causing difficulties in receiving and processing information. This phenomenon also occurs, one of the example is on online store sites. The large number of products is accompanied by a large number of online stores that are offers a wide variety of products in large quantities so that cause confusion and doubt in doing shopping activities on line. Things that are usually done by users or consumers in doing shopping activities sometimes tend to open several shopping sites simultaneously together to search for the same product, while many sites shopping that is available out there so that there are doubts in choosing the product so time tends to be spent just doing searches and sometimes can't find the right product. To overcome this then a Product Search System was built by implementing Webscraping using the XPath method for extraction data. We can reconstruct data from various shopping sites into our system as well as from the test results that by utilizing Webscraping the retrieved data will be faster and the search process will be more efficient.

Keywords: *Webscraping, Webdata Extraction, XPath, E-Commerce.*

1. PENDAHULUAN

A. Latar Belakang

Perkembangan internet yang semakin maju di lingkungan masyarakat memberikan kemudahan dalam mendapatkan berbagai macam informasi, informasi apapun bisa didapatkan dengan sangat mudah dalam waktu yang relatif singkat. Banyaknya sumber informasi yang ada di satu sisi akan memberikan manfaat, karena antara satu sumber dengan sumber lainnya dapat saling melengkapi informasi yang disajikan. Namun, di sisi lain informasi dengan jumlah yang sangat banyak dan beragam akan menyebabkan timbulnya Information Overload.

Information Overload adalah banyaknya jumlah informasi yang diterima oleh manusia sehingga menimbulkan kesulitan dalam penerimaan dan pengolahan informasi (A. Priyanto & Ma'arif, 2018). Fenomena Information Overload salah satunya terjadi pada informasi situs jual-beli/bisnis online.

Sekarang ini, perusahaan yang bergerak dibidang e-commerce terus menawarkan berbagai macam produk dan brand, mulai dari ukuran, bentuk, warna, harga dan variasi yang lainnya, Mayoritas peneliti telah mengindikasikan bahwa dengan banyaknya data dan informasi yang disajikan menyebabkan konsumen mengalami Information Overload. Information Overload didefinisikan sebagai efek negatif yang disebabkan karena terlalu banyaknya informasi terhadap kapasitas permrosesan tiap-tiap individu atau konsumen.

Untuk mengatasi fenomena Information Overload ini, kita tidak mungkin menghilangkan data atau informasi begitu saja, namun dengan adanya Information Overload ini kita dapat memanfaatkannya untuk melakukan ekstraksi data ataupun data mining dalam rangka menemukan pola bisnis yang dilakukan oleh para pesaing. Pola bisnis dapat dianalisis lebih lanjut untuk memahami peluang dan tantangan yang ada.

Salah satu metode yang dapat diterapkan yaitu pengimplementasian Web Scraping dengan menggunakan XPath pada halaman website e-commerce yang diharapkan dapat mempermudah konsumen dalam melakukan pencarian produk. Karena dengan menggunakan teknik Web Scraping

atau yang dikenal juga sebagai Ekstraksi Data, kita dapat melakukan ekstraksi data dari beberapa website E-Commerce yang kemudian memudahkan kita dalam melakukan pencarian produk, karena dengan semakin mudahnya konsumen menemukan produk yang diinginkan, maka konsumen akan terhindar dari kebingungan dalam pemilihan produk.

B. Rumusan Masalah

Berdasarkan uraian dalam latar belakang maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Belum adanya website yang dapat menghimpun pencarian produk secara efisien.
2. Data yang disajikan dalam website e-commerce masih semi terstruktur dan terkadang terdapat informasi yang berlebihan dalam suatu produk.

C. Tujuan Penelitian

Berdasarkan permasalahan yang telah di rumuskan diatas, maka tujuan penelitian ini adalah membangun sebuah sistem pencarian produk menggunakan metode XPath pada halaman website e-commerce.

D. Batasan Masalah

Demi tercapainya tujuan penelitian yang telah di uraikan diatas, maka ruang lingkup pada penelitian ini adalah:

1. Sistem yang dibangun adalah khusus untuk Sistem Pencarian.
2. Implementasi menggunakan Bahasa Pemrograman Python, menggunakan Framework Django.

E. Kontribusi Penelitian

Kontribusi yang dihasilkan dari penelitian ini adalah:

1. Membangun sebuah aplikasi Pencarian Produk menggunakan Web Scraping dengan Metode XPath.
2. Meningkatkan efisiensi dapat mempermudah konsumen dalam mengelola informasi dari produk yang dicari dari website situs belanja online.

2. KAJIAN LITELATUR

A. Web Scraping

Web Scraping (juga diistilahkan Screen Scraping, Web Data Extraction, Web Harvesting) adalah teknik yang digunakan untuk mengekstraksi data dalam jumlah besar dari situs web di mana data diekstraksi dan disimpan ke file lokal di komputer Anda atau ke database dalam format tabel (spreadsheet).

Web Scraping merupakan teknik penting yang digunakan untuk mengekstrak data yang tidak terstruktur (unstructured data) dari sebuah website dan mentransformasikan data tersebut menjadi sebuah data yang terstruktur. (Anand V. Saurkar, Kedar G. Pathare, Shweta A. Gode, 2018).

B. HTML

HTML merupakan sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi dan dapat juga digunakan sebagai link-link menuju halaman web yang lain dengan kode tertentu.

HyperText Markup Language atau HTML adalah bahasa yang digunakan pada dokumen web sebagai bahasa untuk pertukaran dokumen web (Sibero, 2013).

C. XML

XML merupakan teknologi dengan aplikasi dunia nyata, khususnya untuk manajemen, tampilan, dan organisasi data. XML bekerja dengan tujuan markup dari setiap jenis data tetapi dengan kompleksitas yang di eliminasi, XML tidak benar – benar merupakan bahasa, tetapi lebih pada sintaks yang digunakan untuk menjelaskan markup lain (Hunter, 2007) .

D. XPATH

Xpath atau dikenal dengan istilah Extensible Path Language, XML Path Language adalah bahasa query untuk memilih node dari dokumen XML. Xpath dapat digunakan untuk menghitung nilai (misalnya, string, angka, atau nilai Boolean) dari isi dokumen XML. Xpath didefinisikan oleh World Wide Web Consortium(W3C) (Abdilah, 2015). Bahasa Xpath didasarkan pada representasi pohon dokumen XML, dan menyediakan kemampuan untuk menavigasi di sekitar pohon XML, memilih node dengan berbagai kriteria. Dalam penggunaannya yang populer, meskipun tidak dalam spesifikasi resmi, ekspresi XPath sering disebut hanya sebagai XPath.

Implementasi XPath

XPath biasanya digunakan pada bahasa pemrograman maupun kerangka kerja yang sering menangani data berupa dokumen XML atau bahkan dokumen markup lainnya (HTML). Beberapa contoh adalah libxml, lxml, .NET framework, Java, Python dan JavaScript.

Sintaks dan Semantik XPath

a) Axis (definisi antara relasi tree memilih node dan node saat ini)

Tabel 2.1 XPath Axis

Axis Name	Sintaks Singkat	Keterangan
ancestor		Select semua ancestor (parent, grandparent) dari node terkini
ancestor-or-self		Select semua ancestor (parent, grandparent) dari node terkini atau node itu sendiri
atttribute	@::node	Select semua attribute pada node terkini
children	node	Select semua child dari node terkini
descendant	/	Select semua descendant (children, grandchildren) dari node terkini
descendant-or-self	//	Select semua descendant (children, grandchildren) dari node terkini atau node itu sendiri
parent	..	Select parent dari node terkini
preceeding		Select semua node yang muncul sebelum node terkini dalam dokumen kecuali ancestor
following		Select semuanya setelah node terakhir
follow-sibling		Select semua sibling yang ada dari node terkini
self	.	Select dari node terkini

Sebagai contoh menggunakan **axis atribut**, `[//a/@href]`, memilih atribut a href dalam elemen mana saja (//) di pohon dokumen. Sedangkan contoh lain untuk Ekspresi **self** (.) (singkatan untuk self :: node ()) paling umum digunakan dalam predikat untuk merujuk ke node yang saat ini dipilih. Misalnya, `h3[.='Elemen H3']` memilih elemen yang disebut h3 pada node yang dipilih saat ini dengan text “Elemen H3”.

b) Node test (mengidentifikasi nilai dalam sebuah axis).

Node didalam Xpath memiliki beberapa jenis node meliputi : element, attribute, text, namespace, document nodes, processing-instruction, dan comment. Lihat contoh berikut ini:

1. Nodes Example

Berikut merupakan contoh struktur nodes:

```
<?xml version="1.0" encoding="UTF-8" ?>
<bookstore> <!-- Root Element Node -->
    <author>Rifqi Rosyidi</author> <!-- Element Node -->
</bookstore>

lang="en" <!-- Attribute Node -->
```

Gambar 2. 1 Contoh Node

2. Atomic Value

Atomic Value merupakan node yang tidak memiliki parent dan tidak memili anak, contoh:

```

Rifqi Rosyidi

"en"

```

Gambar 2. 2 Contoh Atomic Value

3. Nodes Relationship

Nodes Relationship merupakan Relasi antara node yang satu dengan yang lain dalam struktur markup suatu dokumen. Terdapat beberapa nodes relationship seperti, *Parent, Children, Siblings, Ancestor, Descendants.*

4. Nodes Syntax Example

Berikut merupakan contoh sederhana struktur file xml, seperti pada gambar 2.1.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<bookstore>

    <book>
        <title lang="en">My Title</title>
        <price>40000</price>
    </book>

    <book>
        <title lang="en">My Second Title</title>
        <price>45000</price>
    </book>

</bookstore>
```

Gambar 2.3 Contoh Struktur XML untuk Selecting Nodes.

Berikut ini contoh penulisan sintaks nodes sederhana dari ekspresi XPath :

Tabel 2.2 Sintaks Nodes

Ekspresi XPath	Keterangan
bookstore	Memilih semua node dengan nama "bookstore".
/bookstore	Memilih element root "bookstore" Catatan : Jika sebuah path diawali dengan slash (/) mewakili path mutlak suatu element.
/bookstore/book	Memilih semua elemen "book" pada children "bookstore".
//book	Memilih semua elemen "book" dimanapun berada pada file dokumen.
/bookstore//book	Memilih semua elemen "book" dari turunan "bookstore" dimanapun berada dibawah elemen "bookstore".
//@lang	Memilih semua atribut yang memiliki atribute "lang".

c) Nol atau lebih predikat (untuk mempersempit node yang dipilih).

1. Predikat

Predikat digunakan untuk mencari node dengan kondisi yang lebih spesifik. Penggunaannya selalu ada didalam kurung. Contoh penggunaan daftar predikat bisa dilihat pada Tabel 2.3 dibawah ini.

Tabel 2.3 Nol atau Lebih Predikat

Ekspresi XPath	Hasil
/bookstore/book [1]	Memilih element pertama dari "book" dengan child dari "bookstore".
/bookstore/book [last()]	Memilih element terakhir dari "book" dengan child dari "bookstore".
/bookstore/book [last()-1]	Memilih element terakhir-1 dari "book" dengan child dari "bookstore".
/bookstore/book [position()>3]	Memilih element dari posisi 1-2 dari "book" dengan child dari "bookstore".
//title[@lang]	Memilih semua "title" yang memiliki atribut "lang".
//title[@lang=" en"]	Memilih semua "title" yang memiliki atribut "lang"="en".
/bookstore/book [price>35000]	Memilih semua "book" yang memiliki "price" diatas 35000.
/bookstore/book [price>35000]/title	Memilih semua "title" dengan syarat "book" yang memiliki "price" diatas 35000.

2. Memilih Beberapa Path

Kita dapat memilih beberapa path menggunakan ekspresi " | ", berikut contoh penggunaannya seperti pada Tabel 2.4 dibawah ini.

Tabel 2.4 Memilih Beberapa Path

Ekspresi XPath	Hasil
//book/title | //book/price	Memilih semua elemets "title" dan semua element "price"
/bookstore/book /title | //price	Memilih semua elemets "title" dan semua element "price"

Berdasarkan contoh penggunaan sintaks XPath yang telah disajikan, menunjukkan bahwa dengan implementasi XPath, kita dapat mengekstrak data yang ada secara lebih mudah dan data yang diambil dapat lebih informatif dan terstruktur.

3. METODE PENELITIAN

Pada ini menjelaskan tentang gambaran tahapan yang dilakukan untuk menganalisis data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang tahapan dari penelitian.

A. Analisis Permasalahan

Permasalahan yang sering terjadi pada saat melakukan pencarian produk di

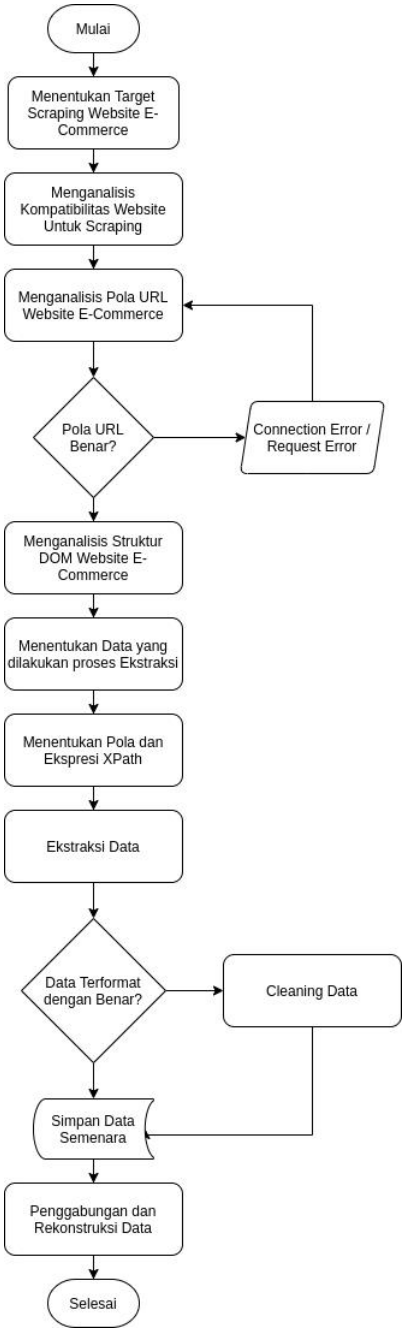
situs belanja online yaitu banyaknya jumlah situs belanja online serta produk yang ditawarkan sangat banyak dan beragam yang kemudian menimbulkan kebingungan bagi konsumen untuk mencari produk yang diinginkan sehingga mengakibatkan kita harus melakukan pencarian produk di tiap-tiap situs belanja online dan berdampak menghabiskan terlalu banyak waktu hanya untuk melakukan pencarian produk diberbagai situs belanja online.

Selain produk yang ditawarkan cukup banyak, kita terkadang sulit membandingkan antara produk yang satu dengan produk lain yang serupa terutama antar situs belanja karena selain struktur produk yang berbeda dari tiap situs belanja, ketika melakukan perbandingan antara produk yang satu dengan yang lain terkadang menimbulkan keraguan pada saat membeli produk tersebut dikarenakan faktor faktor tertentu seperti kualitas maupun harga yang beragam dari situs belanja.

B. Solusi yang Diusulkan

Berdasarkan analisis tersebut, solusi yang diusulkan adalah membuat Sistem Informasi Pencarian Produk dengan teknik webscraping pada website e-commerce dengan menggunakan metode XPath untuk membantu dalam melakukan proses pencarian produk dari berbagai macam situs belanja online.

Untuk menerapkan solusi yang telah dijelaskan sebelumnya, maka perlu adanya kerangka kerja atau framework yang sistematis. Framework merupakan model alur kerja sistem yang akan dilakukan dalam penyelesaian masalah yang akan dibahas. Berikut merupakan framework yang tertera pada gambar 3.1.



Gambar 3.1 Model Sistem yang Diusulkan

1) Menentukan Target Scraping Website *E-Commerce*.

Terdapat banyak sekali website e-commerce yang tersedia, pada tahapan awal ini kita menentukan target website yang akan kita lakukan proses scraping. Pada mulanya target awal website scraping yaitu seperti Bukalapak, Tokopedia, Elevenia, Shopee, OLX, Uqishop, Zalora, dan Sorabel.

2) Menganalisis Kompatibilitas Website *E-Commerce.*

Menganalisis Kompatibilitas Website E-Commerce dilakukan untuk

melakukan pengecekan apakah website tersebut dapat dilakukan scraping atau tidak. Langkah sederhana yang dapat dilakukan yaitu dengan mengunjungi website tersebut dan mengakses file robots.txt setelah base http url, https://example.com/robots.txt.

a) Bukalapak

```
User-agent: *
Disallow: /*/admin_link$
Disallow: /*/edit_label
Disallow: /*/one_click_form$
Disallow: /*/seller_actions$
Disallow: /*/tiny_link_to_cart$
Disallow: /*/user_manage$
Disallow: /account_settings/
Disallow: /api/v2/
Disallow: /ca/
Disallow: /campaign/
Disallow: /cart/
Disallow: /css/
Disallow: /components/products/related_deal/
Disallow: /fb_login
Disallow: /feed/feed.rss
Disallow: /gplus_login
Disallow: /js/
Disallow: /messages/
Disallow: /online/
Disallow: /palaver/
Disallow: /payment/
Disallow: /products/*/bookmarker$
Disallow: /products/*/recommended_bid_value
Disallow: /products/*_-*
Disallow: /products/load_image_gallery
Disallow: /product_discussion/vote/*
Disallow: /product_reviews/votes/*
Disallow: /product_reviews/vote/*
Disallow: /recomendations/*/owner_actions
Disallow: /system/
Disallow: /system2/
Disallow: /system3/
Disallow: /system4/
Disallow: /users/*/inactivity
Disallow: /users/login_menu
Disallow: /users/*/report
Disallow: /v2/
Disallow: /u/*/label/bukamobil
Disallow: /cdn-cgi/
```

Gambar 3.2 File Robots.txt Bukalapak

Selain robots.txt pada website bukalapak, kita dapat mengecek robots.txt dari beberapa webite e-commerce lainnya seperti olx, lazada, elevenia, uqishop, sorabel, dll. Dari beberapa File Robots.txt diatas dapat disimpulkan bahwa ada beberapa situs belanja yang tidak memperbolehkan bots, scraper maupun crawler untuk mengakses situs untuk melakukan operasi pencarian produk, salah satu contoh pada situs shopee, pada web tersebut User Agent: * artinya semuanya yang mengakses ada beberapa yang tidak diperbolehkan seperti Dissallow: /search*searchPrefills untuk pencarian produk sehinnga menyebabkan website tersebut tidak dapat kita lakukan scraping secara langsung dengan memasukkan parameter request, selain shopee ada website yang lain yang tidak dapat kita lakukan scraping seperti Bukalapak, Tokopedia maupun situs belanja yang lain. Sehingga mengerucutkan penelitian ini pada 4 Website yaitu Elevenia, Olx, Uqishop, dan Sorabel.

3) **Menganalisis Pola Url Target Website E-Commerce**

Dari ke 4 website tersebut kemudian kita kembangkan dengan melakukan analisis dengan melakukan pencarian pada tiap-tiap website dan menemukan pola url yang cocok untuk mendukung dalam proses pencarian. Pola url tersebut dapat dilihat seperti pada Tabel 3.1.

Tabel 3.1 Pola URL Website

No	Website	Base URL	Joined Keyword
1	OLX	https://www.olx.co.id	/items/q-{search_keyword}
2	Elevenia	https://www.elevenia.co.id	/search?q={search_keyword}
3	UqiShop	https://www.uqishop.com	/?s={search_keyword}&post_type=product
4	Sorabel	https://www.sorabel.com	/products/search?query={search_keyword}

Dari Tabel diatas dapat kita cermati bahwa tiap-tiap website mempunyai ciri khas tersendiri dalam menangani request pencarian pada website tersebut. Kita dapat melakukan hal yang sama untuk website e-commerce yang lain, namun harus menyesuaikan dengan target url website masing-masing yang akan kita lakukan proses scraping karena pola tiap-tiap url berbeda.

4) Menganalisis Struktur DOM

Menganalisis Struktur DOM dilakukan untuk mengetahuit Tree HTML yang terkandung pada website yang akan kita analisa, pada tahapan ini kita melakukan analisis struktur untuk mengetahui jenis tag, atribut, properti maupun data yang akan kita ambil, data yang akan kita ambil yaitu terkait produk, sehinnga kita hanya melakukan analisis

aterhadap struktur DOM produk tersebut, berikut merupakan struktur DOM dari tiap-tiap website yang akan kita lakukan proses scraping.

a) OLX

Berikut merupakan struktur dom pada halaman pencarian OLX:

Gambar 3.10 Potongan Struktur DOM Halaman Pencarian Produk OLX

Selanjutnya kita dapat melakukan analisis juga terhadap web lainnya seperti elevenia, uqshop & sorabel.

5) Menentukan Data yang Akan Dilakukan Proses Ekstraksi

Data yang akan dilakukan ekstraksi dari ke-4 website tersebut kita melakukan ekstraksi data-data terkait yang penting seperti nama produk, gambar produk, harga produk, asal produk, link produk dan source produk, tahap ini merupakan lanjutan tahap lanjutan setelah menganalisis struktur DOM, kita melakukan penentuan bagian struktur yang akan kita ambil terkait data-data terkait yang penting tersebut.

6) Menentukan Pola dan Ekspresi XPath

Dari data yang akan telah kita tentukan diatas kita mendapatkan informasi mengenai tag serta data yang kita inginkan, langkah selanjutnya yaitu menemukan pola dan ekspresi XPath yang cocok agar kita mendapatkan data dan informasi yang kita inginkan, berikut ini merupakan pola dan ekspresi XPath pada tiap tiap situs e-commerce.

a) OLX

Berikut ini merupakan tabel pola dan ekspresi XPath berdasarkan data yang akan kita ambil pada halaman OLX.

Tabel 3.2 Expresi XPath Halaman OLX

No	Data Produk	Ekspresi XPath
1	Gambar	//figure[@class="_2grx4"]//img//@src
2	Nama	//span[@class="_2tW1I"]/text()
3	Harga	//span[@class="_89yzn"]/text()
4	Asal	//div[@class="_1KOFM"]/span[@class="tjgMj"]/text()
5	Link	//span[@class="_2tW1I"]/ancestor::a/@href

b) Elevenia

Berikut ini merupakan tabel pola dan ekspresi XPath berdasarkan data yang akan kita ambil pada halaman Elevenia.

Tabel 3.3 Expresi XPath Halaman Elevenia

No	Data Produk	Ekspresi XPath
1	Gambar	//div[@class="group"]/a/img/@src
2	Nama	//a[@class="pordLink notranslate"]/text()
3	Harga	//div[@class="prc"]/strong/text()
4	Asal	//ul[@class="sellerPlace"]/li/text()
5	Link	//a[@class="pordLink notranslate"]/@href

c) Uqishop

Berikut ini merupakan tabel pola dan ekspresi XPath berdasarkan data yang akan kita ambil pada halaman Uqishop.

Tabel 3.4 Expresi XPath Halaman Uqishop

No	Data Produk	Ekspresi XPath
1	Gambar	//div[@class="product-image-box"]/a/img/@src
2	Nama	//h2[@class="woocommerce-loop-product__title"]/text()
3	Harga	//span[@class="price"]/ins/span/text()
4	Asal	-
5	Link	//a[@class="product-detail-link"]/@href

d) Sorabel

Berikut ini merupakan tabel pola dan ekspresi XPath berdasarkan data yang akan kita ambil pada halaman Sorabel.

Tabel 3.5 Expresi XPath Halaman Sorabel

No	Data Produk	Ekspresi XPath
1	Gambar	//div[@class="group"]/a/img/@src
2	Nama	//a[@class="pordLink notranslate"]/text()
3	Harga	//div[@class="prc"]/strong/text()
4	Asal	//ul[@class="sellerPlace"]/li/text()
5	Link	//a[@class="pordLink notranslate"]/@href

7) Ekstraksi Data

Setelah mendapatkan pola dan ekspresi XPath, kita dapat melakukan request ke masing-masing website dan melakukan ekstraksi data sesuai dengan keyword yang kita inginkan, proses ekstraksi dilakukan secara otomatis oleh sistem, dengan mendeteksi apakah ada GET request dari sistem.

Pada saat melakukan ekstraksi data, traffic dan koneksi internet juga cukup berpengaruh dalam kecepatan melakukan ekstraksi data. Sistem yang dibangun menggunakan Framework Djagno, library lxml, requests untuk menangani XPath, sedangkan Framework Django saat ini belum mendukung untuk operasi multiple request secara bersamaan (Asyncronous) sehingga proses ekstraksi dilakukan satu persatu (Syncronous) selelah setiap request dilakukan.

Data yang telah diekstrak tadi jika terdapat jenis data atau tipe data yang tidak sesuai selama proses ekstraksi maka kita lakukan proses cleaning data terlebih dahulu agar data antar situs belanja dapat kita presentasikan dalam bentuk yang konsisten.

8) Cleaning Data

Selanjutnya kita melakukan analisa apakah data sudah dalam bentuk dan terformat seperti yang kita inginkan, salah satu contohnya yaitu pada saat melakukan ekstraksi data terutama data dengan format yang berbeda misalkan harga, terkadang pada saat ekstraksi terdapat data harga dengan format yang berbeda-beda seperti Rp. 10.000,00- , 10.0000, 10000, Rp 10.000, 00, 10.000,00. maka perlu adanya cleaning data agar semua data yang kita peroleh dari masing-masing situs belanja tersebut terformat dengan baik dan benar, sehingga memudahkan kita dalam menampilkan data maupun dalam proses sorting kedepan.

9) Simpan Data

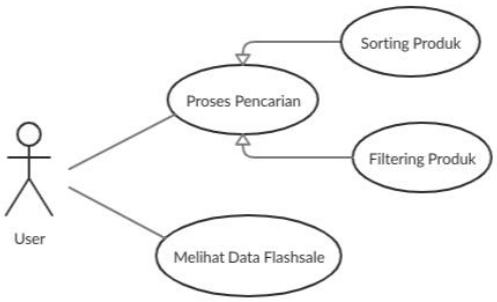
Jika data telah selesai dilakukan proses cleaning, langkah selanjutnya yaitu kita simpan data sementara tersebut, dalam hal ini data yang kita simpan secara bertahap karena request dilakukan secara bergantian (Syncronous),

10) Penggabungan dan Rekonstruksi Data

Jika data sudah disimpan, maka data tersebut akan kita rekonstruksi ulang susunan urutan datanya dengan menggunakan sorting sequence matcher atau sorting berdasarkan relevansi agar data yang kita gabung tidak berurutan sesuai urutan produk pada situs e-commerce, melainkan kita sorting ulang sehingga data tersebut kita sesuaikan dengan produk yang kita cari. Setelah data digabungkan dan kita rekonstruksi kemudian kita dapat menampilkan data hasil pencarian tersebut ke dalam sistem yang akan kita bangun.

Use Case Diagram

Permodelan Use Case Diagram dapat dilihat pada gambar 3. 13.



Gambar 3.13 Use Case Diagram Sistem Pencarian

Penjelasan dari Use Case Diagram adalah sebagai berikut:

a) *User*

User berperan sebagai pengguna sistem pada Sistem Implementasi Webscraping untuk Pencarian Produk dengan menggunakan Metode XPath pada Halaman Website e-commerce, User dapat melakukan pencarian produk, melakukan sorting, filter dari pencarian produk, melihat flashsale atau diskon dari situs belanja. Yang berperan sebagai user adalah Pengguna maupun Konsumen yang sedang ingin melakukan kegiatan belanja online.

b) Use Case

1. Proses Pencarian

Proses melakukan pencarian produk dengan menginputkan keyword pencarian yang diinginkan, dan memberikan informasi hasil pencarian.

2. Sorting Produk

Proses melakukan sorting hasil pencarian produk berdasarkan relevansi, harga tertinggi atau harga terendah dari pencarian produk.

3. Filtering Produk

Proses melakukan filter hasil pencarian produk berdasarkan source marketplace.

4. Melihat Data Flashsale

Proses melihat data produk dari kategori flashsale, promo, diskon atau daily deals berdasarkan data yang telah diekstrak dari situs e-commence.

4. HASIL DAN PEMBAHASAN

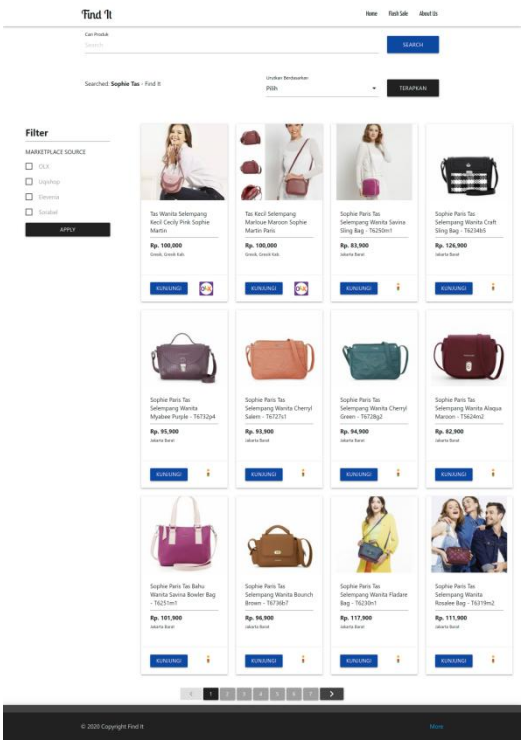
Rangkaian hasil penelitian dan pengujian pada sistem. Pengujian dilakukan untuk mengetahui fungsionalitas dari sistem penjurusan dan rangkaian hasil penelitian digunakan untuk menganalisis hasil penelitian sehingga menghasilkan kesimpulan yang sesuai dengan tujuan penelitian.

A. Antarmuka Halaman

Implementasi antarmuka merupakan tampilan dari sistem yang dibangun. Hasil keluaran setiap halaman pada saat pengujian sistem ditampilkan sebagai berikut:

1. Halaman Pencarian

Halaman Pencarian merupakan halaman yang akan muncul ketika kita melakukan pencarian suatu produk, pada halaman ini selain melakukan pencarian, kita juga dapat menerapkan beberapa filter seperti sorting produk mulai dari harga, relevansi maupun filter berdasarkan source marketplace. Tampilan halaman pencarian dapat dilihat pada Gambar 4. 2 berikut:



Gambar 4.2 Antarmuka Halaman Search

B. Hasil Perbandingan

Hasil perbandingan digunakan untuk menganalisis perbandingan antara teknik yang lama dengan teknik yang telah kita bangun baik dari segi struktur objek suatu produk maupun kecepatan request antara website situs belanja online dengan sistem yang telah kita bangun dengan webscraping yang menggunakan XPath dengan Framework Django.

Perbandingan kecepatan dari implementasi webscraping dengan metode XPath untuk mendukung proses pencarian dapat dilihat pada tabel 4.7 sebagai berikut:

Tabel 4. 7 Tabel Perbandingan Kecepatan

Keyword	**Average Request Time**				**Sistem Baru**
	OLX	*Uqishop*	*Eleve nia*	*Sorabel*	
Tas	7.8s	3.0s	9.8s	16.3s	11.6s
Sepatu	7.2s	3.4s	8.8s	19.1s	11.9s
Laptop	9.2s	3.7s	10.2s	17.6s	9.2s
Pulpen	11. 5s	3.3s	9.5s	16.6s	9.4s
Baju	7.0s	3.9s	10.1s	20.1s	10.8s
Gamis	12. 6s	3.1s	8.4s	17.1s	7.2s

Celana	10.1s	3.3s	11.4s	20.1s	8.9s
Jaket	11.3s	3.1s	11.9s	20.9s	10.9s
Dompet	15.1s	4.3s	9.5s	17.2s	7.9s
Rata-rata	**10.2s**	**3.5s**	**9.96s**	**18.4s**	***9.6s***
Perbandingan	**Added Request Total Time: 42.06s Average Request Total Time: 10.5s**				**9.6s**

Jika kita lihat dari tabel 4.7 tentang perbandingan kecepatan diatas mengenai request time untuk mengetahui kecepatan loading website khususnya pada halaman pencarian maka dapat dilihat bahwa jika kita melakukan request satu persatu secara bergantian pada tiap-tiap situs belanja secara data dari tiap-tiap situs belanja dapat kita jumlahkan sehingga kita akan mendapatkan total time (added) sebanya 42,06s. sedangkan bila kita bandingkan dengan sistem yang telah dibangun. waktu yang diperlukan 4,4 kali lebih cepat yaitu hanya memerlukan 9,6s. bila kita dibandingkan dengan cara mengunjungi satu persaru halaman situs belanja online.

Meskipun demikian jika kita melakukan request bersamaan ke semua website dengan membuka beberapa tab sekaligus dan melakukan request sekaligus maka dapat diasumsikan average total request yaitu 10,5s. Jika kita bandingkan dengan sistem yang baru maka kita akan mendapatkan selisih sebanyak 0,9s. mungkin selisihnya tidak terlalu banyak, namun dengan menggunakan sistem yang baru ini kita dapat melakukan pencarian sekaligus ke situs-situs tersebut hanya cukup dengan satu sentuhan saja.

5. KESIMPULAN

A. Kesimpulan

Berdasarkan analisis hasil pengujian pada eksperimen yang telah dilakukan, maka dapat diambil kesimpulan bahwa dengan menggunakan XPath dalam Pencarian Produk pada halaman website E-Commerce dengan teknik Webscraping ini mampu melakukan ekstraksi data dari beberapa situs belanja online, proses pencarian akan jauh lebih cepat dibandingkan dengan melakukan pencarian tersendiri pada tiap-tiap situs belanja online.

B. Saran

Dalam rangka mencapai performa yang lebih baik pada penelitian di masa mendatang, maka dapat dilakukan pengembangan sebagai berikut:

1. Pada E-Marketplace Bukalapak, Lazada, Blibli, Shopee tidak dapat discraping data produknya karena terdapat markup class banyak yang sama dan keamanan pada script JS yang membuat scraping data produk tidak berhasil namun pada saat penelitian, ada salah satu tools webscraping juga yang menerapkan XPath, yaitu Selenium yang memungkinkan kita melakukan scraping pada beberapa website tersebut karena dengan menggunakan selenium kita mengemulasikan browser secara langsung maupun headless (berjalan dibackground) sehingga data tersebut bisa kita diambil karena menunggu semua proses rendering HTML nya selesai sebagaimana yang ditampilkan dibrowser, akan tetapi proses mengekstraknya cukup lama sehingga kurang compatible untuk diimplementasikan dalam penelitian ini. Namun jika kita mengedepankan data yang diambil maka ada baiknya dalam penelitian kedepan akan lebih baik menerapkan Selenium dalam proses scraping.

6. REFERENSI

Priyanto, A., M, Rifqi, Implementasi Web Scraping Dan Text Mining Untuk Akuisisi Dan Kategorisasi Informasi Laman Web Tentang Hidroponik. *Indonesian Journal of Information Sytems* (2018).

Anand V. Saurkar, Kedar G. Pathare, Shweta A. Gode. AN OVERVIEW ON WEB SCRAPING TECHNIQUE AND TOOLS, *International Journal on Future Revolution in Computer Science & Communication Engineering* (2018).

Sibero, AlexanderF.K . 2013. Web Programming Power Pack. Yogyakarta : Mediakom.