BAB II

TINJAUAN PUSTAKA

2.1. 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).

Gambar 2. 1 Basic Architecture Of Web Scraping

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).

2.2. Django

Gambar 2. 2 Logo Django

Django adalah Web Framework yang dibangun dengan menggunakan bahasa pemrograman Python tingkat tinggi yang mendorong rapid development atau pengembangan cepat dan desain pragmatis yang bersih. Django dapat digunakan secara gratis dan open source. (Djangoproject, 2020)

Django memberikan para developer kemungkinan untuk membuat apa pun yang diminta, hanya dengan tingkat imajinasi dan keterampilan pengembang sebagai batasan. Django juga memberikan akses developer ke berbagai macam library dan tools yang telah dibuat sebelumnya untuk mempercepat proses pengembangan

2.3. 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.

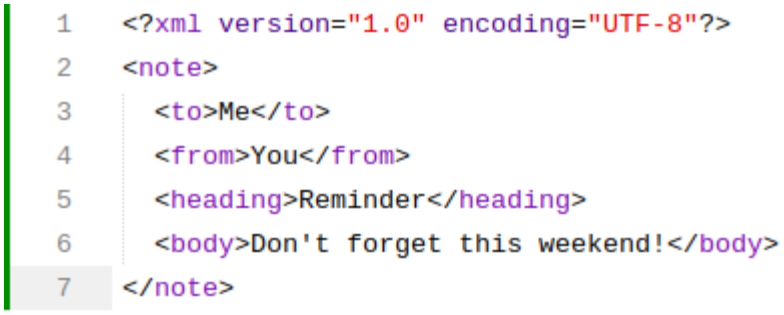
```
1   <!DOCTYPE html>
2   <html>
3       <head>
4           <title>Page Title</title>
5       </head>
6       <body>
7           <h1>This is a Heading</h1>
8           <p>This is a paragraph.</p>
9       </body>
10  </html>
```

Gambar 2. 3 Contoh Struktur Sederhana HTML

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

2.4. 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) .



Gambar 2. 4 Contoh Strukur Sederhana XML

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur, serta menyediakan format tertentu untuk dokumen – dokumen yang mempunyai data terstruktur.

2.5. 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.

2.5.1. Implementasi XPath

XPath adalah bahasa query mengekstrak data dari dokumen XML serta memberikan operasi dasar untuk memanipulasi *string*, angka maupun *boolean*. *HTML DOM Tree* dapat direpresentasikan dalam format XML, yang berarti XPath dapat diterapkan untuk representasi DOM dari halaman website juga. Data ditangani oleh XPath dapat berkisar dari teks, *specific node*, *list node* dan atribut. XPath menawarkan berbagai fasilitas yang terdiri dari *string function*, *comparisson function* dan *mathematical function*. (Nieland, Joacim., 2017).

2.5.2. Sintaks dan Semantik XPath

Xpath memiliki sintaks untuk memilih tempat lokasi. Sebuah tempat lokasi ini mempunyai beberapa langkah untuk mendapatkan langkah lokasi. Langkah lokasi ini terdapat tiga komponen yaitu:

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

Axis Name	Sintaks Singkat	Keterangan
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 7 jenis node meliputi : element, attribute, text, namespace, processing-instruction, comment, dan document nodes. Lihat contoh berikut ini :

a) Contoh Nodes

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<bookstore>
    <book>
        <title lang="en">My Title</title>
        <author>Rifqi Rosyidi</author>
        <year>2020</year>
        <price>40000</price>
    </book>
</bookstore>
```

Gambar 2. 5 Contoh Dokumen XML

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

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

Gambar 2. 6 Contoh Node

b) Atomic Value

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

```
Rifqi Rosyidi

"en"
```

Gambar 2. 7 Contoh Atomic Value

c) Nodes Relationship

Nodes Relationship merupakan Relasi antara node yang satu dengan yang lain dalam struktur markup suatu dokumen.

a. Parent

Setiap elemen memiliki atribute dan satu parent.

```
<book> <!-- Parent -->
    <title lang="en">My Title</title>
    <author>Rifqi Rosyidi</author>
    <year>2020</year>
    <price>40000</price>
</book>
```

Gambar 2. 8 Nodes Relationship - Parent

b. Children

Children merupakan turunan dari parent.

```
<book>
    <title lang="en">My Title</title> <!-- Child -->
    <author>Rifqi Rosyidi</author> <!-- Child -->
    <year>2020</year> <!-- Child -->
    <price>40000</price> <!-- Child -->
</book>
```

Gambar 2. 9 Nodes Relationship - Children

c. Siblings

Siblings memiliki parent yang sama. Seperti contoh pada gambar 2.7. title, author, year dan price adalah sibling.

```xml
<book>
    <title lang="en">My Title</title> <!-- Siblings -->
    <author>Rifqi Rosyidi</author> <!-- Siblings -->
    <year>2020</year> <!-- Siblings -->
    <price>40000</price> <!-- Siblings -->
</book>
```

Gambar 2. 10 Nodes Relationship - Siblings

d. Ancestor

Ancestor merupakan semua silsilah parent yang berkaitan dengan element.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<bookstore> <!-- Ancestor -->
    <book> <!-- Ancestor -->
        <title lang="en">My Title</title>
        <author>Rifqi Rosyidi</author>
        <year>2020</year>
        <price>40000</price>
    </book>
</bookstore>
```

Gambar 2. 2 Nodes Relationship - Ancestor

e. Descendants

Descendant merupakan semua turunan children yang berkaitan dengan element.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<bookstore>
    <book>
        <title lang="en">My Title</title> <!-- Descendants -->
        <author>Rifqi Rosyidi</author> <!-- Descendants -->
        <year>2020</year> <!-- Descendants -->
        <price>40000</price> <!-- Descendants -->
    </book>
</bookstore>
```

Gambar 2. 12 Nodes Relationship - Descendants

d) Contoh Sintaks Nodes Sederhana

Berikut merupakan contoh sederhana struktur file xml, seperti yang terlihat pada gambar 2.13.

```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. 13 Contoh Struktur XML untuk Selecting Nodes

Berikut ini contoh penulisan sintaks nodes sederhana dari ekspresi XPath :

Tabel 2. 2 Sintaks Nodes XPath.

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).

a) Predikat

Predicated digunakan untuk mencari node dengan kondisi yang lebih spesifik. Penggunaan predicate 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.

b) 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 mengimplementasikan XPath, kita dapat mengekstrak data yang ada secara lebih mudah dan data yang diambil dapat lebih informatif dan terstruktur.

2.6. Robots.txt

File Robots.txt digunakan untuk memberikan informasi kepada mesin pencari mengenai halaman mana yang dapat diakses atau tidak yang terdapat pada situs website. Rata-rata situs website menempatkan file robots.txt di *root directory* pada *web server*. File Robots.txt terdapat tiga bagian, yaitu: 'User-agent', 'Disallow', dan sitemap parameter.

```
User-agent: *
Disallow: /wp-admin/
Disallow: /wp-includes/

Sitemap: http://www.example.com/sitemaps.xml
```

Gambar 2. 3 Contoh File Robots.txt

Simbol tanda bintang (*) pada 'User-agent' berarti semua mesin pencari mengindeks website ini. 'Disallow' memblokir crawler/bots pada spesifik folder tertentu dan Sitemap parameter menampilkan link sitemap website tersebut.

2.7. UML (Unified Model Language)

UML (Unified Modeling Language) adalah metodologi kolaborasi antara metoda-metoda Booch, OMT (Object Modeling Technique), serta OOSE (Object Oriented Software Enggineering) dan beberapa metoda lainnya, merupakan

metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa pemrograman yang berorientasi objek (OOP) (Nugroho, 2009).

2.7.1. Use Case Diagram

Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, Use Case menjelaskan interaksi yang terjadi antara aktor atau user dengan inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah Use Case direpresentasikan dengan urutan langkah yang sederhana.

Tabel 2. 5 Tabel Use Case

No	Simbol	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case
2	<< include>> ---------->	*Include*	Menspesifikasikan bahwa use case sumber secara eksplisit.
3	<<extend>> <----------	*Extend*	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
4		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

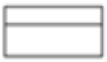
No	Simbol	Nama	Keterangan
6		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Sumber: Satzinger & Jackson, 2012.

2.7.2. Class Diagram

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki 3 bagian utama yaitu attribute, operation, dan name. kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem.

Tabel 2. 6 Tabel Class Diagram.

No	Simbol	Nama	Keterangan
1		*Generalization*	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (ancestor).
2		*Message*	Himpunan dari objek yang berbagi atribut serta operasi yang sama.
3		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
4		*Realization*	Operasi yang benar benar dilakukan oleh suatu objek.
5		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

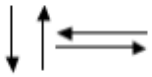
No	Simbol	Nama	Keterangan
6	———	*Association*	Menghubungkan antara objek satu dengan objek lainnya.

Sumber: Satzinger & Jackson, 2012.

2.7.3. Activity Diagram

Activity Diagram adalah diagram yang menggambarkan worlflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa a*ctivity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2. 7 Tabel Activity Diagram.

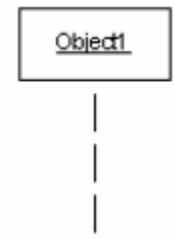
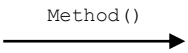
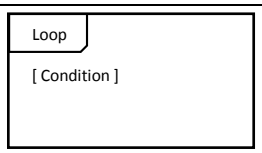
No	Simbol	Nama	Keterangan
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk atau diakhiri.
5		*Decision*	Digunakan untuk menggambarkan suatu keputusan yang harus diambil pada kondisi tertentu.
6		*Line Connector*	Digunakan untuk menghubungkan satu simbol dengan simbol lainn

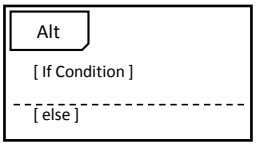
Sumber: Satzinger & Jackson, 2012.

2.7.4. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Banyaknya *sequence diagram* yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak.

Tabel 2. 8 Tabel Sequence Diagram

No	Simbol	Nama	Keterangan
1		*Object Life Line*	Merepresentasikan potongan waktu sebagaimana kebawah, Garis vertikal putus-putus ini menunjukkan peristiwa berurutan yang terjadi pada suatu objek selama proses terjadi.
3		*Activation*	Bar yang di atur sesuai dengan lifelines yang mengindikasikan periode dari waktu ketika aktor melakukan interaksi
4		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi
5		*Asyncronous Return Message*	Direpresentasikan dengan panah bergaris, pesan ini merupakan respon dari message
6		*Loop*	Digunakan untuk memodelkan jika terdapat skenario atau kondisi tertentu yang terjadi secara berulang.

No	Simbol	Nama	Keterangan
7		*Alternative*	Memodelkan pilihan antara dua atau lebih message, disimbolkan dengan rectangle dengan snip single rectangle didalamnya dan dipisahkan dengan garis putus-putus