BAB II

LANDASAN TEORI

2.1. Web Scraping

Menurut (Zhao, 2019), *Web Extraction* atau yang lebih dikenal dengan w*eb scraping* merupakan sebuah teknik untuk mengekstraksi data dari sebuah website dan menyimpannya ke sistem file atau basis data untuk suatu kepentingan. Biasanya, data pada web yang diekstrasi perlu melalui protokol *Hypertext Transfer Protocol (HTTP)* atau melalui web browser. Kegiatan *web scraping* bahkan sudah dilakukan baik secara manual oleh pengguna atau secara otomatis oleh bot.

2.2. Python

Bahasa pemrograman python merupakan salah satu bahasa pemrograman yang dibuat secara dinamis. Ada banyak librari python yang dibuat untuk keperluan *web scraping*. Beberapa diantaranya adalah *request, beautiful soup, lxml, selenium* dan *scrapy*. Selain untuk mengekstrak data *HTML* yang mentah untuk berbagai keperluan, *request* pada python dapat digunakan untuk mengirim form dan mengakses *API* (Prakash & Rashid, 2017).

Penggunaan bahasa python untuk keperluan komputasi ilmiah telah mendapatkan momentum dalam beberapa tahun terakhir. Menjadi fakta bahwa bahasa python memiliki kompleksitas dan mudah dibaca dengan dilengkapi librari ilmiah yang menjadi pendukung dalam penerapan karakteristiknya (Tejedor et al., 2016).

2.3. Metode Web Scraping

Web Scraping memiliki berbagai macam metode yang telah dikembangkan dalam berbagai penelitian, diantaranya:

a. XPath

XPath (XML Path Language) merupakan bahasa kueri untuk memilih bagian-bagian *(nodes)* dari sebuah dokumen *XML*. Selain untuk memilih, *XPath* juga dapat digunakan untuk menghitung nilai seperti: *string*, angka dan *bolean* pada sebuah file *XML* dan *HTML*. *World Wide Web Consortium (W3C)* bahkan telah menetapkan standardisasi dalam penggunaan *XPath*.

b. CSS Selector

CSS Selector merupakan sebuah metode untuk menemukan elemen *HTML* di halaman website dan mengekstrak data darinya. *CSS Selector* dideklarasikan sebagai bagian dari sebuah gaya markup yang berlaku untuk mencocokkan dengan tag dan atribut dalam markup tersebut.

c. HTML DOM

HTML DOM merupakan objek model standar untuk mendapatkan, mengubah, menambah atau menghapus elemen *HTML*. *DOM* bekerja dengan cara mendefinisikan objek dan properti dari semua elemen *HTML*, dengan metode untuk mengaksesnya. Sebuah web browser tidak mengharuskan untuk menggunakan *DOM* dalam menampilkan dokumen *HTML*. Namun dengan *DOM, Javascript* dapat mengakses semua elemen di dalam dokumen *HTML*.

d. Regular Expression (Regex)

Regular Expression (Regex) merupakan konstruksi bahasa untuk mencocokkan teks berdasarkan pola tertentu, terutama untuk kasus-kasus yang kompleks. *Regex* juga digunakan untuk mencocokkan pola-pola karakter tertentu dalam suatu kumpulan *string*. *Regex* memiliki dua macam karakter yaitu karakter biasa dan meta karakter.

2.4. Multiprocessing

Multiprocessing merupakan kemampuan suatu sistem untuk mendukung lebih dari satu prosesor pada saat bersamaan. Suatu program yang menggunakan *multiprocessing* akan dipecah menjadi rutinitas yang lebih kecil yang berjalan secara independent. Sistem operasi akan mengalokasikan komputasi ini ke prosesor yang meningkatkan kinerja sistem.

Penggunaan *multiprocessing* dengan menjadikan proses pengolahan menjadi paralel merupakan hal yang perlu untuk mencapai kinerja terbaik. Setiap tugas *multiprocessing* akan berjalan dalam prosesnya sendiri, setiap program yang berjalan di komputer diwakili oleh satu atau lebih proses (Abeykoon et al., 2017; Sherman & Hartog, 2016).

2.5. Penelitian Terkait

Penelitian yang dilakukan (Rizaldi & Putranto, 2017), yaitu membandingkan metode *web scraping* menggunakan *XPath* dan *CSS Selector* dengan parameter perhitungan waktu, jumlah item, pengukuran memori dan pengukuran *bandwith*. Kemudian penelitian yang dilakukan (Gunawan et al., 2019) masih dalam

perbandingan metode *web scraping* menggunakan metode *Regular Expression, HTML DOM* dan *XPath* dengan parameter pengukuran waktu, pengukuran memori dan pengukuran data.

Beberapa penelitian yang telah dilakukan sebelumnya menjadi landasan acuan untuk melakukan pengukuran jumlah objek data, penggunaan *CPU*, penggunaan memori, penggunaan *bandwith* dan *execution time* pada proses *web scraping* menggunakan empat metode, yaitu *XPath, CSS Selector, HTML DOM* dan *Regular Expression (Regex)* dengan penerapan *multiprocessing*.

Table II-1 Tabel Penelitian Terkait

1	Judul	*Parallelizing X-ray Photon Correlation Spectroscopy Software Tools using Python Multiprocessing*
	Penulis	Sameera K. Abeykoon, Meifeng Lin dan Kerstin Kleese van Dam (2017)
	Penelitian	Menggunakan modul multiprosesor Python untuk memparalelkan fungsi korelasi waktu dalam *scikit-beam*.
	Hasil	Memparalelkan perhitungan fungsi korelasi waktu dari perangkat lunak *XPCS* dalam paket *scikit-beam* menggunakan modul multiprosesor Python. Modul multiprosesor Python memperkenalkan cara untuk memparalelkan beban kerja.
	Keterkaitan	Konsep pemrograman paralel .
2	Judul	*Comparison of Web Scraping Techniques: Regular Expression, HTML DOM and Xpath*
	Penulis	Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan dan Firman Firdaus (2018)
	Penelitian	Membandingkan tiga metode *web scraping* dengan parameter pengukuran waktu, penggunaan memori dan penggunaan data
	Hasil	Metode *HTML DOM* lebih unggul pada parameter pengukuran waktu dan penggunaan data, sedangkan metode *Regex* unggul pada penggunaan memori.
	Keterkaitan	Teknik *web scraping* dan parameter yang digunakan.

3	Judul	*A review of programming languages for web scraping from software repository sites*
	Penulis	Mohan Prakash dan Dr. Ekbal Rashid (2017)
	Penelitian	Mereview empat bahasa pemrograman C, Java, PHP dan Python terkait perpustakaan dan metode penggunaanya pada *web scraping* dan *data extraction.*
	Hasil	Python merupakan bahasa terbaik yang dapat digunakan untuk *web scraping.* Hal ini karena adanya modul-modul yang memumpuni seperti *Scrapy*, *Selenium*, *Spiders* dan lain-lain.
	Keterkaitan	Bahasa pemrograman p*ython*
4	Judul	Implementasi *Web Scraping* dan *Text Mining* untuk Akuisisi dan Kategorisasi Informasi Laman Web Tentang Hidroponik
	Penulis	A Priyanto dan M R Ma'arif (2018)
	Penelitian	Mengumpulkan informasi pada laman web menggunakan *web scraping* dilanjutkan dengan melakukan pengelompokkan kedalam beberapa kategori menggunakan *text mining*.
	Hasil	Mengotomasi proses akuisisi informasi khususnya informasi-informasi yang bersumber dari artikel atau tulisan bebas di internet dan mengelompokannya kedalam beberapa kategori.
	Keterkaitan	Proses *web scraping*
5	Judul	Perbandingan Metode *Web Scraping* Menggunakan *CSS Selector* dan *Xpath Selector*
	Penulis	Taufiq Rizaldi dan Hermawan Arief (2017)
	Penelitian	Membandingkan dua metode *web scraping* dengan parameter pengukuran waktu, penggunaan memori, penggunaan data dan jumlah data
	Hasil	Penggunaan metode *XPATH* untuk *web scraping* situs berita menghasilkan artikel yang lebih lengkap dibandingkan dengan menggunakan metode *CSS Selector*. Metode *XPATH* juga lebih unggul dalam pengukuran waktu.
	Keterkaitan	Teknik *web scraping* dan parameter pengujian
6	Judul	*An Overview On Web Scraping Techniques And Tools*
	Penulis	Anand V. Saurkar, Kedar G. Pathare dan Shweta A. Gode (2018)
	Penelitian	Menganalisis teknik *web scraping* dan *tools* yang digunakan

	Hasil	Ada banyak teknik *web scraping* yang bisa digunakan diantaranya *Classical copy and paste*, *Hypertext Transfer Protocol (HTTP) Programming*, *Hyper Text Markup Language (HTML) Parsing*, *Document Object Model (DOM) Parsing*, *Web Scraping Software* dan *Computer vision web-page analysers* dan alat-alat yang digunakan pun beragam diantaranya *Mozenda, Visual Web Ripper, Web Content Extractor, Import.io* dan *Scrapy*
	Keterkaitan	Teknik *web scraping dan library*
7	Judul	*DECO: Polishing Python Parallel Programming*
	Penulis	Alex Sherman, Peter Den Hartog (2016)
	Penelitian	Mengusulkan penyederhanaan teknik pemrograman paralel tradisional yang meminimalkan interaksi programmer dan tidak memerlukan pengetahuan pemrograman paralel
	Hasil	Penyederhanaan teknik pemrograman bersamaan yang ditargetkan pada pemrogram dengan sedikit pemahaman tentang pemrograman bersamaan
	Keterkaitan	Konsep *python parallel programming*
8	Judul	*Web Scraping and Naïve Bayes Classification for Job Search Engine*
	Penulis	C Slamet, R Andrian, D S Maylawati, Suhendar, W Darmalaksana dan M A Ramdhani (2018)
	Penelitian	Penyederhanaan pencarian kerja melalui konstruksi dan kolaborasi teknik pengikisan web dan klasifikasi menggunakan *Naïve Bayes* di mesin pencari
	Hasil	Menghasilkan aplikasi yang efektif dan efisien bagi pengguna untuk mencari pekerjaan potensial yang sesuai dengan minat mereka.
	Keterkaitan	Konsep *web scraping*
9	Judul	Analisis *Web Scraping* Untuk Data Bencana Alam Dengan Menggunakan Teknik *Breadth-First Search* Terhadap 3 Media Online
	Penulis	Izatul Putri Sonya dan Dr. Prihandoko, Skom (2016)
	Penelitian	Melakukan *scraping* pada data yang tidak terstruktur dibeberapa media online dan mengklasifikasikannya menjadi data yang terstruktur menggunakan teknik *Breadth-First*
	Hasil	Menghasilkan data yang terstruktur berupa tabel dengan beberapa field yaitu no, hari/tanggal, waktu posting, judul, deskripsi, gambar, dan link halaman artikel.

	Keterkaitan	Konsep *web scraping*
10	Judul	*PyCOMPSs: Parallel computational workflows in Python*
	Penulis	Enric Tejedor, Yolanda Becerra, Guillem Alomar, Anna Queralt, Rosa M Badia, Jordi Torres, Toni Cortes dan Jesu´s Labarta (2015)
	Penelitian	Melakukan penelitian terhadap permasalahan umum yang dialami pengguna dalam menggunakan pemrograman paralel pada *Python*
	Hasil	Menyajikan PyCOMPSs, sebuah kerangka kerja yang memfasilitasi pengembangan alur kerja komputasi paralel dalam Python.
	Keterkaitan	Konsep parallel programing
11	Judul	*Comparison Of Python Libraries Used For Web Data Extraction*
	Penulis	Erdinç Uzun, Tarik Yerlikaya Dan Oğuz Kirat (2018)
	Penelitian	Membandingkan tiga pustaka ekstraksi terkenal yang berbeda termasuk *BeautifulSoup, lxml* dan *regex*
	Hasil	Hasil percobaan menunjukkan bahwa *regex* mencapai hasil terbaik dengan rata-rata 0,071 ms. Namun, sulit untuk menghasilkan aturan ekstraksi yang benar untuk *regex* ketika jumlah elemen dalam tidak diketahui, dan *lxml* menunjukkan hasil terbaik dengan rata-rata 9.074 ms
	Keterkaitan	Penggunaan *library web scraping*.
12	Judul	*Web Scraping*
	Penulis	Bo Zhao (2019)
	Penelitian	Menganalisa *web scraping* secara terperinci
	Hasil	Mendeskripsikan *web scraping* secara terperinci
	Keterkaitan	Konsep *web scraping*.

2.6. State of the Art

Beberapa penelitan sudah dilakukan berhubungan dengan penelitian yang sedang dilakukan, peneliti sudah memperluas *state of the art* dari parameter dan teknik yang digunakan pada penelitian sebelumnya, serta penggunaan *paralel programing* yang mampu mengeksekusi beberapa perintah secara bersamaan dengan lebih efisien.

Selain itu, para peneliti sudah menjawab beberapa pertanyaan yang dibutuhkan untuk memulai penelitian ini, dengan mengekplorasi setiap kelebihan untuk menangkap parameter yang dibutuhkan, mengembangkan teknik-teknik yang sebelumnya telah digunakan agar mendapatkan hasil yang lebih efektif. Beberapa penelitian lainnya berfokus pada penggunaan teknik *web scraping*. Tantangan dan peluang pada penelitian ini yaitu penggunaan *web scraping* pada kasus yang berbeda seperti melakukan *scraping* pada beberapa *URL* bersamaan dan melakukan *scraping* dengan banyak objek data. Hal tersebut dapat dilihat pada matrik penelitian berikut:

Table II-2 Tabel Matriks Penelitian

1	Pengarang	Taufiq Rizaldi, Hermawan Arief
	Tahun	2017
	Judul	Perbandingan Metode Web Scraping Menggunakan CSS Selector dan Xpath Selector
	Penjelasan	Penelitian yang dilakukan membandingkan teknik *web scraping CSS Selector* dan *XPath* pada *weblog* http://blog.detik.com dengan membagi tiga kategori yang digunakan untuk *scraping* yaitu, komunitas, hiburan dan kuliner. Bahasa pemrograman yang digunakan *python* dengan menggunakan *framework scrapy*. Parameter yang

		diguanakan dalam pengujian yaitu pengukuran waktu, penggunaan memori dan jumlah item.
	Kesimpulan	Untuk penggunaan memori baik *CSS Selector* dan *XPath* tidak memiliki perbedaan yang begitu signifikan, sedangkan untuk waktu yang dibutuhkan *XPath* memiliki waktu pemrosesan yang lebih singkat dari *CSS Selector*.
	Kekurangan	Objek penelitian yang digunakan tidak diperjelas apakah dari setiap kategori yang dipilih hanya judul artikel saja yang diambil atau isi dari setiap konten yang muncul dari kategori tersebut. Penggunaan *framework scrapy* dirasa kurang tepat, karena *scrapy* berfungsi sebagai *engine* yang sudah mencakup tujuh komponen kompleks *Scheduler, Item Pipeline, Downloader, Downloader Middleware, Spiders, Spiders Middleware* sehingga akan mengabaikan sedikit teknik yang digunakan.
	Keterkaitan	Penggunaan teknik *web scraping* dan parameter
2	Pengarang	Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan, Firman Firdaus
	Tahun	2018
	Judul	*Comparison of Web Scraping Techniques: Regular Expression, HTML DOM and Xpath*
	Penjelasan	Penelitian yang dilakukan melakukan perbandingan teknik *web scraping* dengan teknik *HTML DOM, Regular Expression,* dan *XPath* pada website *http://testing-ground.scraping.pro*. Java menjadi bahasa pemrograman yang digunakan pada penelitian ini. Parameter yang digunakan pengukuran waktu, penggunaan data dan penggunaan memori.
	Kesimpulan	Kinerja *web scraping* metode *regular expression* paling kecil dalam penggunaan *memory* dibandingkan metode *HTML DOM*, dan *XPath*. Sedangkan *HTML DOM* membutuhkan waktu paling sedikit dan penggunaan data paling kecil dibanding metode *Regex* dan *XPath*.
	Kekurangan	Tidak adanya pengukuran objek yang didapatkan dan pengukuran *CPU,* tidak dijelaskan mengapa bahasa pemrograman yang digunakan *java.*
	Keterkaitan	Penggunaan teknik *web scraping* dan parameter