Jurnal Ilmiah
ILMU KOMPUTER
Universitas Udayana

Vol. X, No. 2, September 2017 **ISSN 1979 - 5661**

PERBANDINGAN METODE WEB SCRAPING MENGGUNAKAN CSS SELECTOR DAN XPATH SELECTOR

Taufiq Rizaldi [1], Hermawan Arief [2]

Jurusan Teknologi Informasi, Program Studi Manajemen Informatika,
Politeknik Negeri Jember
E-mail: taufiq_r@polije.ac.id[1], hermawan_arief_putranto@yahoo.com[2]

ABSTRAK

Pemanfaatan data atau berita yang tersebar di internet untuk meningkatkan peluang keberhasilan dalam sebuah usaha melalui analisa trend pasar adalah hal yang sangat umum pada saat ini. Penjelajahan Web *(Crawl*) dan ekstraksi data dari web (*Scraping*) menjadi salah satu hal yang penting, agar tidak terjadi data yang kurang sempurna, dan data yang diterima adalah data yang paling baru. *CSS Selector* dan *Xpath* merupakan salah satu metode yang umum digunakan dalam melakukan proses crawling. Terdapat perbedaan dari jumlah data yang terambil, besar file output dan waktu pemrosesan dari kedua metode tersebut, dimana *Xpath* memiliki keunggulan pada jumlah data yang terambil dan waktu pemrosesnya yang berakibat pada ukuran file *output* yang lebih besar. Sedangkan untuk penggunaan *memory* pada kedua metode pada proses *crawling* tidak memiliki perbedaan yang signifikan.
Kata Kunci: *Web Crawling, Web scraping, Scrapy, Xpath, CSS Selector*

ABSTRACT

Utilization of data or news spread across the internet to increase the chances of success in a business through the analysis of market trends is a very common thing at the moment. Web Crawling and data extraction from the web (Scraping)becomes an important thing, in order to avoid catching imperfect or incomplete data, and to ensure data that we received are the most recent data. CSS Selector and Xpath is one of the commonly methods that used for data crawling. There is a difference in the amount of data captured, the output file size and processing time of both methods, where Xpath has an advantage over the amount of data retrieved and the processing time resulting Xpath have more larger output file sizes. As for the memory use for both methods on the crawling process does not have a significant difference.
Keywords: *Web Crawling, Web scraping, Scrapy, Xpath, CSS Selector*

1 PENDAHULUAN

Internet adalah sebuah tempat berkumpulnya sejumlah besar informasi di dunia, baik itu teks, media atau data dalam format lain yang biasanya ditampilkan dalam sebuah halaman web. Kemudahan untuk mengakses data tersebut sangat penting bagi keberhasilan sebagian besar bisnis di dunia modern. Bagi perusahaan yang bergerak dibidang pemasaran, data tersebut bisa digunakan untuk mengetahui trend pasar yang berkembang saat ini, sehingga bisa diketahui strategi pemasaran yang paling tepat untuk tiap produk. Bagi perusahaan yang berbasis *Ecommerce* juga bisa memanfaatkan data tersebut untuk analisis pasar atau sekedar perbandingan harga dengan kompetitor *ecommecre* lain.

Keberadaan data dalam jumlah besar dan beragam juga mendorong beberapa

peneliti untuk menggali informasi yang tersirat atau melakukan proses analisis pada fenomena tersebut. Diantaranya adalah penelitian tentang analisis sentiment berbasis *ontology* untuk mengukur persepsi produk yang menggunakan *microblogging tweeter* sebagai sumber datanya (Akbar et-al, 2015). Pada penelitian tersebut, data yang berupa kicauan (*tweet*) diperoleh melalui layanan yang disediakan oleh *microblogging tweeter* yang disebut *Tweeter API.* Namun tidak semua website yang ada di internet memberikan layanan tersebut, dan itu menjadi masalah tersendiri apabila ada yang ingin menakses data mereka. Berikutnya adalah penelitian tentang klasifikasi dokumen berita yang memuat konten *E-Government* menggunkan metode *Naïve Bayes Classifier* (Wijaya, 2016). Dalam penelitian tersebut, digunakan portal berita nasional www.jawapos.com sebagai sumber data yang digunakan untuk proses *training*.

Sayangnya sebagian besar data di internet memiliki hak akses yang sangat terbatas. Tidak seperti *microblogging tweeter* yang memiliki *Tweeter API*, sebagian besar situs web di internet tidak menawarkan opsi untuk menyimpan data yang mereka tampilkan ke penyimpanan lokal komputer, atau ke situs web pribadi. Untuk mengakses data dari situs-situs seperti itu dibutuhkan teknik khusus yaitu *scraping*.

Web scraping merupakan teknik yang digunakan untuk mengekstrak sejumlah besar data dari situs web dimana data yang sudah diekstraksi disimpan ke sebuah file lokal di komputer atau ke *database* dalam format tabel (*spreadsheet*). Inilah yang memungkinkan user untuk mengeksplorasi isi dari situs web tanpa mengunjungi situs web yang bersangkutan, sehingga user bisa melakukan berbagai bentuk analisis tanpa mengganggu *resource* situs web yang bersangkutan.

Banyak *tool* yang bisa digunakan untuk melakukan *web scraping*, salah satu yang populer adalah scrapy. Scrapy adalah sebuah *framework* aplikasi yang digunakan untuk menjelajahi (*crawling*) situs web dan mengekstrak data terstruktur sehingga dapat digunakan untuk berbagai aplikasi lain yang bermanfaat, seperti *data mining*, pemrosesan informasi atau arsip sejarah (Loukas-Kouzis, 2016). Walaupun bersifat *open source*, namun scrapy merupakan *framework web scraping* yang handal dan fleksibel, sehingga hanya dibutuhkan sedikit penyesuaian apabila kita ingin menjelajahi beberapa situs yang berbeda. Hal yang biasa dilakukan pada saat melakukan *web scraping* adalah mengekstraksi data dari halaman web yang berbentuk dokumen html. Scrapy memiliki mekanisme tersendiri untuk mengekstrak data dari dokumen html yang disebut *selector* karena mereka "memilih" bagian tertentu dari dokumen HTML yang

ditentukan baik oleh ekspresi XPath maupun CSS. XPath adalah bahasa untuk memilih simpul (*node*) dalam dokumen XML, yang juga bisa digunakan dengan HTML. CSS adalah bahasa untuk menerapkan style pada dokumen HTML. Dalam *paper* ini akan disajikan perbandingan hasil dari penggunaan kedua mekanisme tersebut, sehingga dapat diketahui manakah yang paling baik digunakan, apabila kita ingin mengekstrak data dari sebuah situs yang tidak menyediakan layanan ekstraksi data.

2 RUMUSAN MASALAH

Dari uraian diatas, permasalahan yang diangkat dalam penelitian ini adalah bagaimana membuat mekanisme yang dapat membandingkan penggunaan XPATH dan CSS *selector* sebagai metode *web scraping* menggunakan scrapy. Hal ini membawa kedalam permasalahan yang lebih rinci, yang pertama adalah, bagaimana mengimplementasikan Scrapy sebagai web scraper yang akan menjelajah situs tertentu. Kedua, bagaimana membuat spider yang mengimplementasikan kedua metode seleksi tersebut. Yang ketiga adalah bagaimana menyajikan data yang dihasilkan dari penerapan metode XPATH dan CSS *Selector* sehingga diketahui metode mana yang paling tepat untuk digunakan.

3 KAJIAN PUSTAKA

3.1 *Web Crawler*

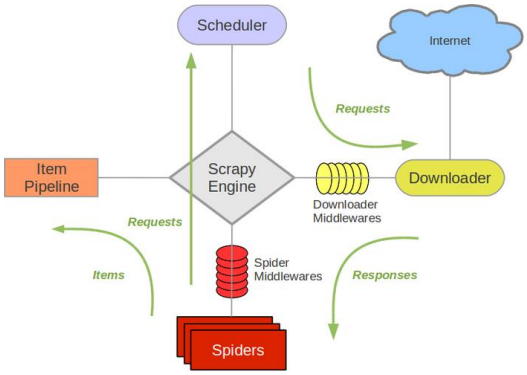
Web Crawler adalah suatu program atau *script* otomatis yang relatif simple, yang dengan mengunakan metode tertentu melakukan scan ke semua halaman-halaman Internet untuk membuat index dari data yang dicarinya. Pada umunya *crawling* diterapakn pada web yang banyak disebut *Web Crawling. Web Crawling* pada umumnya digunakan pada search engine yang dilakukan oleh sekelompok kompuer yang dikluster dimana setiap komputer menjalankan beberapa *thread* (Hatzi, 2014).

Prosedur dari *Web Crawling* dimulai dari memilih satu set URL yang akan dilakukan proses *crawling* dimana URL tersebut menyediakan banyak link ke halaman yang penting untuk proses *Crawling*. *The crawler* akan men-*download* konten ari halaman tersebut ke dalam penyimpanan local seperti *hardisk* dll. Secara bersaman thread yang ada pada Crawler mencari link baru dari halaman yang terhubung dengan halaman yang telah diunduh, link baru membentuk yang disebut dengan *crawler's frontier*. Tujuan utama dari *crawler's frontier* adalah mendapatkan halaman baru sebanyak mungkin dan mengupdate tingkat kebaruan halaman yag telah diunduh.

3.2 Scrapy

Scrapy adalah sebuah *framework* yang digunakan untuk melakukan proses *crawling* dan meng*extract* data yang

tersruktur. Scrapy digunakan pada proses data mining, pemrosesan informasi dan pengarsipan *history*. Scrapy dibangun dengan menggunakan pyhton yang disupport dengan twisted (Jing Wang, 2012). Terdapat tujuh komponen utama pada scrappy seperti yang ditunjukan pada gambar 1, yaitu *Scheduler, Item Pipeline, Downloader, Downloade Middleware, Spiders, Spiders Middleware*.



Gambar 1 . Arsitektur Scrapy

Scrapy Engine bertanggung jawab untuk mengendalikan arus data antar semua komponen sistem. *Downloader* bertangung jawab untuk mengambil halaman web yang diminta dan memasukananya kedalam *engine*. *Spiders* adalah sebuah class yang dibuat oleh user untuk memindah respon yang didapat dari *engine* dan mengektrak item dari respon tersebut. *Pipeline* bertanggung jawab untuk memproses item setelah item tersebut terekstrak oleh *spiders*. *Downloader middlewares* adalah perantara atau jembatan yang berada diantara *engine* dan *downloader* yang bertugas memproses *request* dari *engine* ke *downloader* dan memberikan respon dari *downloader* ke *engine*. *Downloader middlewares* menyediakan mekanisme yang sesuai untuk memperluas fungsi *Scrapy* dengan memasukkan kode yang dapat dirubah sesuai dengan kebutuhan.

4 METODOLOGI

4.1 Objek Penelitian

Dalam penelitian ini, objek penelitiannya adalah sebuah *weblog* yang bernama blogdetik (http://blog.detik.com/). Blog ini disediakan oleh salah satu situs berita polpuler di Indonesia detik.com (http://www.detik.com) sebagai wadah untuk menampung karya tulis seluruh *blogger* di Indonesia, baik yang sudah memiliki blog sendiri maupun yang belum.

Ada tiga kategori dalam blog ini yang digunakan sebagai objek *scraping* yaitu, komunitas, hiburan dan kuliner. Hanya artikel yang berada dibawah kategori tersebut yang akan diekstrak. Hal ini untuk mengetahui apakah scrapy juga bisa digunakan sebagai web scrapng terbimbing.

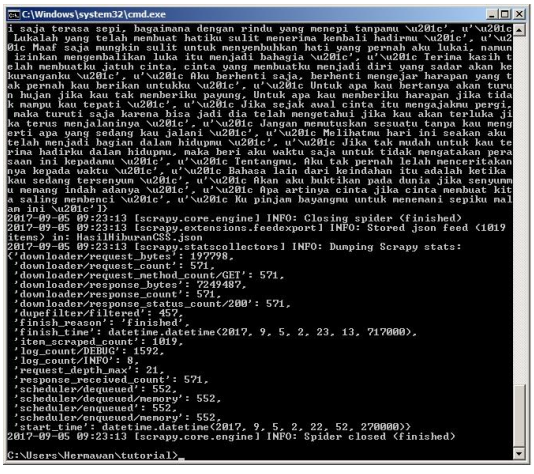
4.2 Variable Penelitian

Dari semua artikel yang berada didalam blogdetik, hanya data tertentu yang akan diekstrak dan disimpan kedalam bebrapa *variable*. *Variable* yang pertama adalah link yang berisi tautan untuk menuju halaman web yang memuat artikel berita. Yang kedua adalah judul, yang berisi judul artikel, kemudian berikutnya adalah deskripsi yang berisi deskripsi singkat dari artikel yang bersangkutan dan yang terakhir

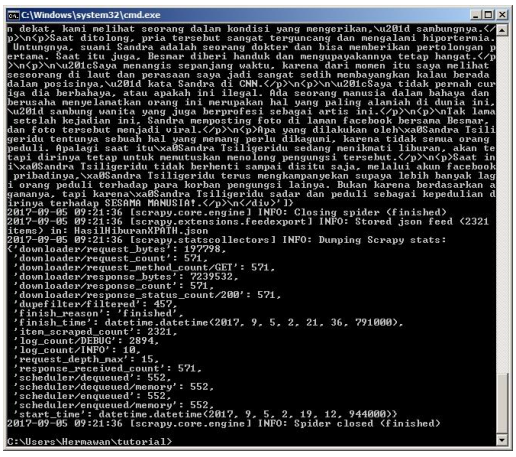
adalah *variable* post yang berisi artikel secara lengkap. Ke-empat *variable* tersebut digunakan oleh dua *spider* yang memuat dua metode yang berbeda,yaitu CSS *Slector* dan Xpath *Selector.*

5 HASIL DAN PEMBAHASAN

Hasil luaran dari proses web scraping dalam penelitian ini disimpan kedalam file dengan ekstensi .JSON, yang kemudian disebut sebagai korpus berita komunitas, korpus berita hiburan dan korpus berita kuliner. Masing-masing korpus berita tersebut berisi hasil web scraping blogdetik menggunakan dua metode yaitu XPATH *Selecto*r dan CSS *Selector* yang disimpan secara terpisah berdasarkan kategori artikelnya. Hasil tampilan dari proses tersebut seperti pada gambar 2 untuk penggunaan CSS *Selector* dan gambar 3 untuk penggunaan Xpath.



Gambar 2. CSS Selector



Gambar 3. Xpath

Pada pengujian sistem terdapat empat fokus pengujian yaitu jumlah item, ukuran file .json, penggunaan memory dan waktu yang dibutuhkan untuk proses *crawling*. Hasil perbandingan proses *crawling* dengan mengunakan css dan Xpath untuk korpus berita hiburan ditunjukan pada Tabel 1.

Tabel 1. Perbandingan korpus hiburan

Korpus	Hiburan	
	CSS	XPATH
Jumlah Item	1019	2321
Ukuran (KB)	1364	7991
Penggunaan Memory	552	552
Waktu	0:00:21	0:02:24

Pada proses *crawling* korpus berita hiburan jumlah item yang didapat dengan xpath lebih banyak daripada penggunaan css dan berdampak pada besarnya file .json dan waktu yang dibutuhkan xpath untuk proses *crawling*. Pada Tabel 2 menunjukan perbandigan fokus pengujian untuk korpus berita komunitas.

Tabel 2. Perbandingan korpus komunitas

Korpus	Hiburan	
	CSS	XPATH
Jumlah Item	925	1581
Ukuran (KB)	1447	5549
Penggunaan Memory	553	553
Waktu	0:03:20	0:00:18

Pada proses *crawling* korpus berita komunitas jumlah item yang didapat dengan xpath lebih banyak daripada penggunaan css dan berdampak pada besarnya file .json akan tetapi waktu yang dibutuhkan xpath untuk proses *crawling* lebih cepat dibandingkan dengan *crawling* menggunakan CSS. Pada saat dilihat hasilnya, beberapa variable post yang digunakan untuk menyimpan artikel berita menggunakan metode XPATH ternyata kosong. Hal ini kemungkinan disebabkan karena adanya penulisan node yang berbeda pada beberapa halaman web blogdetik, sehingga system melakukan *crawling* kedalam link tersebut namun tidak bisa mengambil elemen dibawah *node* yang berbeda. Untuk Tabel 3 menunjukan perbandigan untuk korpus berita kuliner.

Tabel 3. Perbandingan korpus kuliner

Korpus	Hiburan	
	CSS	XPATH
Jumlah Item	316	460
Ukuran (KB)	435	1905
Penggunaan Memory	202	202
Waktu	0:02:01	0:00:08

Pada proses *crawling* untuk korpus berita kuliner, jumlah item dan ukuran file yang didapat menggunakan metode XPATH lebih besar dari pada menggunakan metode CSS. Penyebabnya juga sama dengan kasus sebelumnya, yaitu terdapatnya variable kososng yang ikut tersimpan kedalam file.

Dari ketiga korpus berita yang sudah didapatkan, jumlah item dan ukuran file yang didapatkan menggunakan metode XPATH lebih besar dibandingkan menggunakan metode CSS. Hal ini disebabkan karena pada saat menggunakan metode XPATH semua *node* yang berada dibawah Selector akan dijelajahi (*crawl*) terlepas ada atau tidaknya variable yang ingin disimpan, sehingga ada beberapa item yang tersimpan ke dalam file namun kosong. Selain itu, saat kita menggunakan metode XPATH, semua elemen yang berada dibawah selector akan ikut tersimpan. Hal ini berarti kita tidak hanya menyimpan artikel berita saja, namun juga semua kode HTML yang berada dibawah selector. Sehingga dibutuhkan proses lain untuk membersihkan artikel yang didapatkan dari kode HTML.

Pada proses *web scraping* menggunakan metode CSS, jumlah item dan file yang didapatkan relative lebih kecil. Hal ini disebabkan karena hampir semua node yang ada pada halaman blogdetik menggunakan *style* CSS yang sama, sehingga hanya elemen dibawah selector yang dapat tersimpan. Selain itu, artikel yang dihasilkan juga *relative* lebih bersih dari kode HTML. Hal ini disebabkan

karena *system* bisa menyeleksi elemen yang dibutuhkan dengan lebih spesifik.

6 PENUTUP

Penggunaan metode XPATH Selector untuk web scraping situs berita menghasilkan artikel yang lebih lengkap dibandingkan dengan menggunakan metode CSS *Selector*. Hal ini ditunjukkan dengan jumlah item dan ukuran file yang didapatkan lebih besar dibandingkan metode CSS *Selector*. Namun hal ini juga menyisakan pekerjaan yang lebih banyak, karena butuh proses lain untuk menghilangkan kode HTML yang tidak diinginkan dari artikel yang dihasilkan menggunakan metode XPATH *Selector*.

Untuk penggunaan memory baik metode XPATH *Selector* dan CSS *Selector* tidak memiliki perbedaan yang signifikan bahkan cenderung sama. Hal ini disebabkan karena engine scrapy yang baik dalam penggunaan *resource*-nya, sehingga kinerjanya tidak membebani mesin, baik komputer lokal maupun *server* blogdetik.

Untuk waktu yang dibutuhkan pada proses *crawling* dan *scraping* secara umum metode XPATH *Selector* memiliki waktu proses yang lebih cepat daripada menggunakan metode CSS *Selector*. Pada metode XPATH, selector cukup mengikuti *node* pada halaman web, bukan mencari *style* halaman seperti pada metode CSS, sehingga waktu yang dibutuhkan relatif lebih singkat.

7 DAFTAR PUSTAKA

Akbar, Subhan Agus, Eko Sediyonob, Oky Dwi Nurhayati. 2015. Analisis Sentimen Berbasis Ontologi di Level Kalimat untuk Mengukur Persepsi Produk. Jurnal Informasi Bisnis. 02. 2015. Universitas Diponegoro. Semarang

Kouzis-Loukas, Dimitrios. 2016. *Learning Scrapy*. Packt Publishing: Birmingham-Mumbai.

Hatzi, Vassiliki, dkk. 2014. *Web Page Download Scheduling Policies for Green Web Crawling*. 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM).

Jing Wang, Yuchun Guo. 2012. *Scrapy-based Crawling and User-behavior Characteristics Analysis on Taobao*. 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover.

Wijaya, Akhmad Pandu, Heru Agus Santoso. 2016. *Naive Bayes Classification* pada Klasifikasi Dokumen Untuk Identifikasi Konten *E-Government*. Journal of Applied Intelligent System. 2016;1(1):48-55