Perbandingan Metode *Web Scraping* Dalam Pengambilan Data: Kajian Literatur

Khirana Dwicahyo
Program Studi Informatika
Fakultas Teknologi Industri, Universitas Islam Indonesia
Yogyakarta, Indonesia
19523134@students.uii.ac.id

Chanifah Indah Ratnasari
Program Studi Informatika
Fakultas Teknologi Industri, Universitas Islam Indonesia
Yogyakarta, Indonesia
chanifah.indah@uii.ac.id

Abstract*—Web scraping merupakan suatu teknik untuk melakukan ekstraksi sejumlah data yang terdapat pada *website* tertentu. Data menjadi kebutuhan yang sangat penting khususnya bagi para peneliti dalam mencari suatu fenomena ataupun dalam mencari informasi. *Web scraping* banyak digunakan dalam penelitian seperti pengembangan *web*, analisis sentimen, dan analisis perbandingan harga. Berdasarkan penelitian yang telah menggunakan metode *web scraping*, terdapat kebutuhan data yang berbeda-beda dari penelitian-penelitian tersebut. Perbedaan ini menjadikan metode *web scraping* semakin berkembang dan beragam. Adapun metode tersebut seperi Xpath *Selector*, CSS *Selector*, JSON *Parsing*, HTML *Parsing*, dan metode lainnya. Namun dari berbagai metode *web scraping* yang ada, masing-masing memiliki karakteristik pengambilan data yang berbeda-beda dan tidak dapat dilakukan pada seluruh *website* karena adanya proteksi ataupun jenis dari *website* yang ingin dituju. Maka dari itu, *paper* ini melakukan kajian literatur untuk melihat perbedaan dan memberi kesimpulan dari berbagai metode *web scraping* yang telah digunakan pada penelitian sebelumnya. Kajian ini juga melihat dari sisi performa metode *web scraping* yang telah dilakukan. Hasil dari penelitian ini ditemukan bahwa penelitian mengenai implementasi *web scraping* banyak dilakukan dan penggunaan metode *web scraping* dengan mengekstraksi dokumen HTML juga banyak digunakan pada penelitian sebelumnya.

Keywords*—web scraping, data, ekstraksi data, pengumpulan data.

I. PENDAHULUAN

Web scraping merupakan suatu teknik untuk melakukan ekstraksi sejumlah data yang terdapat pada *website* [1]. Pendapat lain menyebutkan, *web scraping* merupakan teknik untuk mendapatkan suatu informasi dari situs tertentu untuk dapat dilakukan pengambilan data baik secara manual ataupun otomatis [2]. Walau begitu dua definisi tersebut memiliki fokus yang sama, yaitu untuk mengambil data dari suatu *website*. Data menjadi kebutuhan yang penting khususnya bagi para peneliti dalam melakukan analisis terhadap suatu fenomena ataupun dalam mencari suatu informasi [1]. *Web scraping* banyak digunakan dalam penelitian seperti pengembangan web [3], analisis sentimen [4], dan analisis perbandingan harga [5].

Berdasar dari penelitian sebelumnya yang menggunakan metode *web scraping*, terdapat kebutuhan data yang berbeda-beda dalam melakukan ekstraksi dan juga pengumpulan data. Perbedaan ini tentunya menjadikan metode *web scraping* yang digunakan juga semakin berkembang dan menjadi beragam. Adapun metode *web scraping* tersebut seperti CSS *Selector*, Xpath *Selector*, JSON *Parsing*, HTML *Parsing*, dan metode lainnya. Beragam metode tersebut memiliki karakteristik pengambilan data yang cukup berbeda dan tidak dapat dilakukan pada seluruh *website* karena adanya proteksi ataupun jenis *website* yang ingin dilakukan *scraping* [6].

Tujuan dari *paper* ini adalah untuk melakukan kajian, melihat perbedaan dan mengambil kesimpulan dari berbagai metode *web scraping* yang telah digunakan pada penelitian-penelitian sebelumnya. Kajian ini juga akan melihat dari sisi performa metode *web scraping* yang telah dilakukan. Diharapkan dengan adanya kajian ini dapat membantu para peneliti untuk menggunakan metode *web scraping* yang sesuai dengan keperluan penelitian yang akan dilakukan.

II. METODE PENCARIAN LITERATUR

Dalam mencari literatur terdapat kata kunci yang digunakan untuk memilih literatur yang akan dikaji. Kata kunci tersebut adalah: (a) "*web scraping*", (b) "*scraping data*", dan (c) "*implementasi web scraping*". Pencarian dilakukan melalui portal jurnal seperti Google Scholar, Garuda, dan ReseachGate. Literatur juga ditelusuri melalui kesesuaian antara tujuan dan daftar pustaka yang memiliki topik yang sama dalam pembahasan kajian ini. Proses selanjutnya literatur yang akan dipilih perlu memenuhi kriteria yang telah ditetapkan oleh penulis, yaitu:

- Literatur merupakan terbitan kisaran tahun 2017 hingga tahun 2022.
- Literatur membahas mengenai perbedaan metode web *scraping* atau implementasi *web scraping* pada *website*.
- Metode *web scraping* yang digunakan pada literatur masih relevan.
- Literatur menggunakan metode *web scraping* dalam pengambilan sumber data untuk penelitian..

III. METODE ANALISIS

Metode analisis yang digunakan yaitu membuat bingkai analisis dengan menampilkan hasil dari kajian literatur yang telah dilakukan. Dari bingkai analisis, literatur akan dibagi ke dalam beberapa jenis metode yang digunakan, untuk melihat perbedaan dari masing-masing metode *web scraping*. Perbedaan juga akan melihat dari sisi performa metode yang digunakan mulai dari penggunaan CPU, Memori, waktu yang dibutuhkan dalam melakukan *scraping*, dan bagaimana perbedaan data yang dihasilkan.

IV. HASIL DAN PEMBAHASAN

Literatur yang telah dikumpulkan, selanjutnya dianalisis, dan dibuat rangkumannya dalam bentuk tabel untuk memudahkan dalam melihat perbedaan-perbedaan dari literatur-literatur tersebut. Hal tersebut ditunjukkan pada Tabel 1.

TABLE I. KAJIAN LITERATUR

Literatur	Tahun	Metode *Web Scraping*					*Tools*	Hasil Penelitian
		Xpath Selector	**CSS Selector**	**DOM Parsing**	**HTML tag**	**JSON API**		
[1]	2017	✓	✓				Scrapy	Xpath Selector mengahasilkan jumlah data yang lebih besar dibandingkan dengan CSS Selector, dan keduanya memiliki waktu proses yang relatif sama.
[2]	2021			✓			-	Pengambilan data dari tiga situs tersebut berhasil dilakukan menggunakan Parsing DOM.
[6]	2022			✓		✓	Scrapy,Goutte, cURL, Cheerio, Selenium	Penggunaan tools Scrapy, Cheerio, Goutte dan cURL tidak dapat mengeksekusi scraping melalui DOM sehingga pada website yang memuat data menggunakan Javascript tidak dapat dilakukan. Sedangkan pada Scrapy dapat dilakukan karena dapat mengekseksui Javascrpit pada saat scraping. Dari segi waktu scraping melalui JSON API sangat efisien dibanding melalui DOM dan juga dari pengambilan data, JSON API dapat mengambil data lebih banyak dibanding melalui DOM hanya saja dalam pengambilan data melalui JSON API ada beberapa data yang tidak terdapat dan hanya ada pada DOM seperti data “deskripsi produk”.
[7]	2022	✓			✓		Selenium	Informasi yang diinginkan dari penelitian tersebut berhasil didapatkan melalui penggunaan Xpath dan Tag HTML, dan implementasi web scraping berfungsi dengan baik pada halaman web.
[8]	2019				✓		-	Berdasarkan implementasi yang telah dilakukan menggunakan metode web scraping, dilakukan pengujian unit, integrasi, dan validasi menghasilkan nilai 100% valid dan sistem yang dikembangkan dapat berkerja pada jenis browser yang berbeda.
[9]	2020			✓			-	Data artikel ilmiah yang ada pada Google Shcholar berhasil diambil menggunakan HTML DOM, dan terdapat kurang lebih 2.523 artikel berhasil diambil.
[10]	2020			✓			-	Penggunaan metode DOM Parsing dapat dilakukan dalam mengambil data produk marketplace.
[11]	2017			✓			-	Sistem mampu menghasilkan dokumen korpus dengan menggunakan metode scraping HTML DOM. Dari implementasi metode HTML DOM telah berhasil mengambil sebanyak 38.712 pasang korpus pararel, dan juga penggunaan HTML DOM bergantung dengan hardware dan kecepatan internet yang digunakan.
[12]	2021			✓			-	Implementasi metode Web Scraping yang dilakukan dapat dilakukan dalam mendapatkan perkiraan kata kunci dan allintittle dari mesin pencari Google dengan memberikan akurasi 100%. Yang mengindikasikan bahwa teknik yang dilakukan dalam mendapatkan kata kunci yang diinginkan memiliki nilai akurasi yang luar biasa.
[13]	2020				✓		-	Penggunaan HTML tag dalam melakukan scraping berhasil memperoleh 5.211 judul penelitian kesehatan yang ada pada Jurnal SINTA.
[14]	2022			✓			Selenium	Penelitian yang dilakukan dalam pengambilan informasi situs berita dapat dilakukan menggunakan metode web scraping dan hasil datanya sesuai dengan yang diinginkan dan berhasil disimpan dalam bentuk file excel csv.
[15]	2022			✓			Selenium	Penerapan yang dilakukan dalam menggunakan metode web scraping berhasil

								dilakukan dengan baik dalam melakukan scraping data deskripsi produk.

A. *Xpath Selector dan CSS Selector*

Pada hasil penelitian [1], dijelaskan cukup mendetail perbedaan mengenai penggunaan Xpath dan CSS *Selector*. Dalam penelitian tersebut pengambilan digunakan dengan bantuan *tools* Scrapy untuk melakukan pengambilan data dari dua metode yang berbeda. Karakteristik pengambilan data dari kedua metode ini adalah dengan memilih salah satu simpul (*node*) dalam suatu dokumen HTML. Namun hasil proses pengambilan data dari kedua metode tersebut memiliki penggunaan memori yang relatif sama, juga waktu yang dibutuhkan dalam proses pengambilan data Xpath lebih cepat dibanding dengan CSS dan ukuran *file* data yang dimiliki dalam metode Xpath lebih besar. Walaupun begitu kedua metode dapat dilakukan tanpa adanya kendala.

B. *HTML tag*

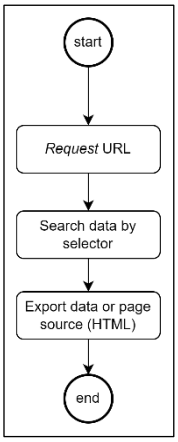
Hasil literatur yang ditemukan juga banyak menggunakan pengambilan data melalui *tag* atau elemen dan atribut pada dokumen HTML. Termasuk pada penelitian [7], [8], dan [13] yang karakteristik pengambilan datanya melalui pengambilan dengan elemen dan atribut yang ada pada dokumen HTML, lalu hasil dikumpulkan dan ditampilkan pada web yang telah dikembangkan. Pada penelitian [9] juga melakukan hal yang serupa, namun data yang diambil pada dokumen HTML tersebut adalah elemen *table row*, sehingga proses pengambilannya cukup rumit. Prinsipnya tetap sama, dengan mengambil elemen dan atribut yang dibutuhkan lalu mengonversikan ke dalam teks, hanya saja struktur kode HTML pada *website* yang dituju berbeda-beda sehingga diperlukan *logic* dalam pengambilan datanya. Hasil dari metode ini berhasil melakukan ekstraksi pada elemen pada dokumen HTML yang dimuat oleh suatu *website* dan menampilkan hasil dengan baik pada web yang dikembangkan tanpa adanya kendala.

C. *DOM Parsing*

DOM umumnya merupakan model objek yang membantu dalam mendefinisikan dokumen HTML untuk bisa dilakukan modifikasi pada bahasa pemrograman tertentu. Seperti pada penelitian [6], DOM *parsing* dilakukan untuk melakukan ekstraksi data pada dokumen HTML. Penelitian ini menggunakan berbagai *tools* dalam melakukannya. Seperti Selenium, Scrapy, Cheerio, Goutte dan cURL.

1) Selenium

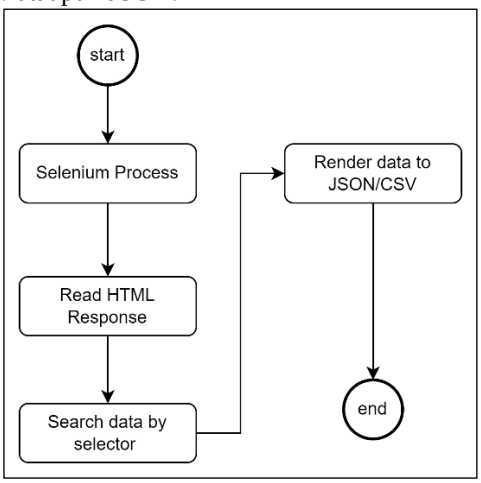
Selenium merupakan *tools* yang memungkinkan untuk menjalankan *browser* dalam memuat data pada suatu *website* secara otomatis dengan menggunakan *web driver*. Proses dalam selenium meliputi *request* URL, mencari data sesuai dengan *selector* pada dokumen HTML, dan melakukan *export* hasil data kedalan JSON,CSV, atau *page source*.



Gambar 1 alur *scraping* menggunakan Selenium dengan metode DOM *parsing*

2) Scrapy

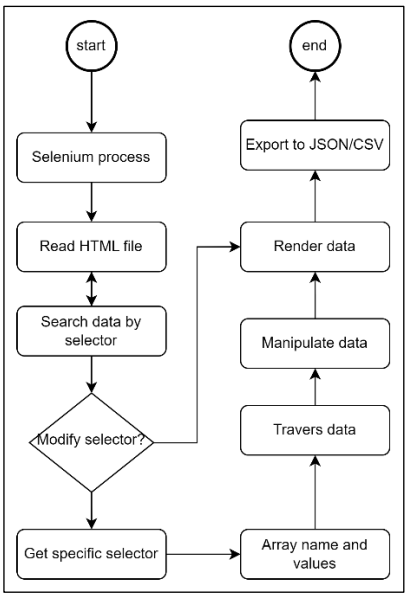
Penggunaan scrapy dalam metode DOM *parsing* hampir sama dengan menggunakan selenium, yaitu dengan melakukan *request* URL, cari data yang ingin diambi melalui Xpath *Selector*, kemudian data di *render* lalu di *export* dalam format csv ataupun JSON.



Gambar 2 Alur *scraping* scrapy dengan metode DOM *parsing*

3) Cheerio

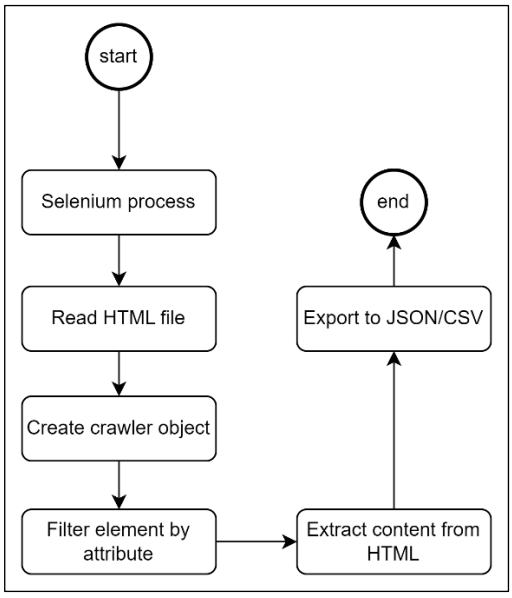
Penggunaan cheerio dalam metode DOM *parsing* juga memiliki karakteristik pengambilan data yang hampir sama pada *tools* sebelumnya. Namun yang membedakan adalah dalam cheerio dapat dilakukan modifikasi dalam memilih *selector* data yang ingin di ambil untuk lebih spesifik, lalu setelah itu dilakukan *encode* kedalam *array*. Dalam cheerio terdapat opsi untuk melakukan manipulasi data sebelum dilakukan export.



Gambar 3 alur *scraping* Cheerio dengan metode DOM *parsing*

4) Goutte

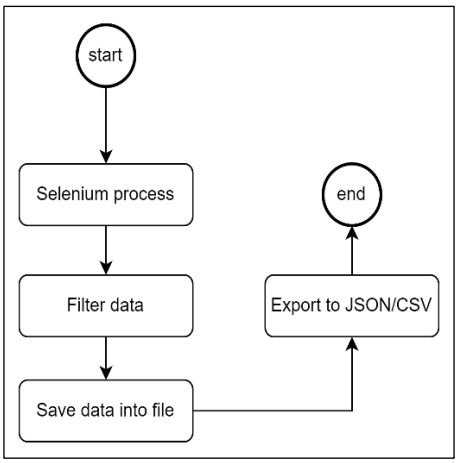
Dalam prosesnya Goutte proses *scraping* juga dilakukan dengan melakukan *request* URL, yang membedakan pada Goutte diperlukan proses *filtering* dengan memilih CSS *Selector* untuk mengambil data yang diinginkan. Lalu data yang dikumpulkan di *export* pada file JSON ataupun CSV.



Gambar 4 alur *scraping* Goutte dengan metoe DOM *parsing*

5) cURL

Dalam *tools* ini penggunaan cURL dalam metode DOM *Parsing* dilakukan dengan melakukan *request* URL seperti biasanya sama seperti *tools* sebelumnya, lalu dilakukan *filtering* data, *filtering* dapat dilakukan menggunakan Xpath ataupun CSS *Selector* dari ekstrasi dokumen HTML yang telah dilakukan sebelum akhirnya data di export.



Gambar 5 alur *scraping* cURL dengan metode DOM *parsing*

Melalui kelima *tools* tersebut dilakukan pengambilan data pada beberapa *website* seperti Tokopedia, OVO, dan Airpaz. Hasil yang diperoleh dari kelima *tools* tersebut bahwa kecepatan *tools* melalui metode DOM *parsing* dalam melakukan *scraping* memiliki nilai yang relatif sama, dan pengambilan data melalui situs Airpaz memiliki nilai penggunaan CPU yang sangat tinggi. Penelitian [6] ini menggunakan tiga parameter untuk melihat perbedaan dari masing-masing *tools* yang digunakan yaitu penggunaan CPU, memori, waktu yang digunakan, dan kecepatan per item dalam detik.

TABLE II. PERBANDINGAN DENGAN METODE DOM PARSING

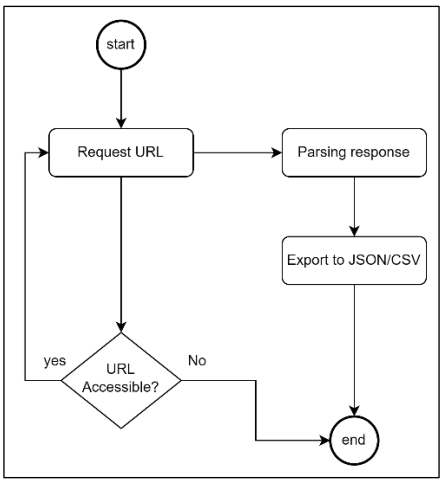
		Scrapy	Cheerio	Goutte	cURL	Selenium
Tokopedia	CPU (%)	18.78	16.17	15.51	15.43	18.2
	Memory (%)	3.15	4.15	3.93	3.87	3.63
	Time (s)	2167	2196	2266	2301	1910
	Speed (item/s)	0.06	0.05	0.05	0.05	0.06
OVO	CPU (%)	11.69	13.18	13.51	11.94	12.1
	Memory (%)	2.73	2.3	2.22	2.14	3.06
	Time (s)	672	653	648	663	756
	Speed (item/s)	0.3	0.3	0.29	0.28	0.16
Aripaz	CPU (%)	42.71	42.76	42.4	42.77	46.55
	Memory (%)	2.38	2.58	2.53	2.46	3.77
	Time (s)	57	54	56	55	66
	Speed (item/s)	0.56	0.59	0.55	0.56	0.44

D. JSON API

Melanjutkan dari penelitian [6], penggunaan metode JSON API juga diterapkan pada *tools* seperti Scrapy, Goutte, dan cURL.

1) Scrapy

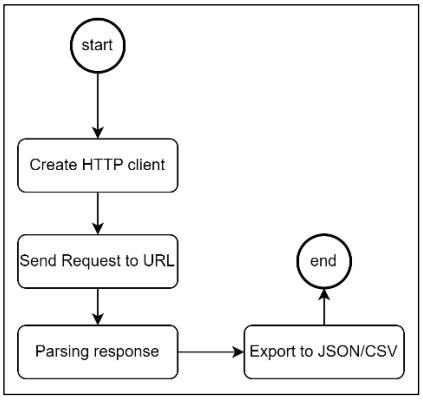
Dalam metode ini Scrapy akan melakukan *request* API dan menghasilkan *response* berupa JSON. Setelah itu dari setiap URL yang dilakukan *request* akan dilakukan *parsing* untuk mengambil data yang diinginkan. Kemudian setelah dilakukan *parsing* hasil akan di *export* ke dalam format JSON ataupun CSV.



Gambar 6 alur *scraping* menggunakan Scrapy dengan metode API

2) Goutte

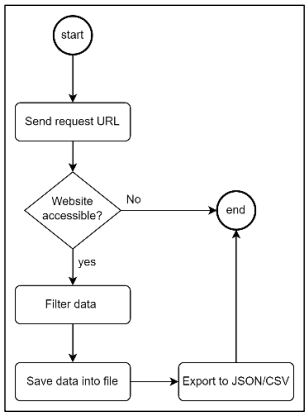
Berbeda dengan Scrapy, pada tahap awal *tools* Goutte akan membuat HTTP *Client* untuk melakukan *request* pada setiap URL. Kemudian akan dilakukan *parsing* dan di *export* sesuai format yang diinginkan.



Gambar 7 alur *scraping* menggunakan Goutte dengan metode API

3) cURL

Sama dengan *tools* lainnya cURL juga melakukan *request* pada setiap URL dan menghasilkan *response* JSON. Tetapi dalam cURL tidak dilakukan *parsing*, namun melakukan *filtering* dengan menggunakan Regular expression dengan cara *user* melakukan pencarian secara manual melalui pola dokumen dari hasil *response*. Ketika proses *filetring* selelesai data yang telah diambil akan di *export* baik JSON ataupun CSV.



Gambar 8 alur *scraping* menggunakan Selenium dengan metode API

TABLE III. PERBANDINGAN DENGAN METODE API

		Scrapy	Goutte	cURL
Tokopedia	CPU (%)	13.56	9.34	10.44
	Memory (%)	0.35	0.19	0.12
	Time (s)	12.74	19.99	34.89
	Speed (item/s)	14.01	6.31	4.08
OVO	CPU (%)	8.46	5.92	4.69
	Memory (%)	0.8	0.19	0.22
	Time (s)	65.62	127.69	193.81
	Speed (item/s)	3.77	1.77	1.12
Airpaz	CPU (%)	0.76	3.80	0.12
	Memory (%)	0.23	-0.03	-0.03
	Time (s)	32.04	35.64	50.12
	Speed (item/s)	1.08	5.36	0.87

Hasil dari penelitian ini menunjukkan bahwa *scraping* melalui API lebih efisien dalam segi waktu jika dibandingkan pada metode ekstraksi pada *file* HTML. Hal ini karena *scraping* melalui ekstraksi pada *file* HTML memerlukan waktu untuk memuat data pada setiap halamannya. Hasil data yang diperoleh juga berbeda, *scraping* dengan metode ekstraksi *file* HTML dapat mengambil data yang tidak dapat diambil dengan API. Ini karena ekstraksi *file* HTML dapat bebas memilih *selector* untuk diambil sehingga cakupan data yang didapat lebih luas. Di sisi lain, *website* Airpaz yang menjadi salah satu objek penelitian terdapat kendala dalam melakukan *scraping*. *Website* Aripaz memiliki server yang kurang stabil sehingga dalam proses pengambilan datanya cukup terkendala dalam melakukan *request* untuk halaman berikunya dan pastinya penggunaan memori dalam melakukan *scraping* cukup tinggi dibanding dengan *website* lainnya. Hal ini mungkin terjadi dikarenakan pada *website* Airpaz perubahan data terjadi cukup cepat dalam hitungan jam. Namun, untuk *website* lainnya tidak memiliki hasil yang signifikan baik menggunakan metode ekstraksi *file* HTML ataupun melalui API.

Melalui hasil analisis yang telah dilakukan beberapa penelitian cukup banyak melakukan pengembangan web dan pengembangan sistem dalam melakukan *crawling* data menggunakan metode *web scraping*. Jenis penelitian ini banyak mengambil data melalui metode ekstraksi dokumen HTML dari *website* yang akan dituju sesuai dengan *input* nama produk yang ingin dicari dari *website* atau sistem yang telah dikembangkan. Rata-rata pengambilan data pada jenis penelitian ini banyak merujuk pada jenis *website e-commerce*, *marketplace*, dan portal berita. Hasil masing-masing penelitian yang melakukan pengembangan web ataupun pengembangan sistem tidak memiliki perbedaan yang cukup signifikan, begitu juga data yang diambil sesuai dengan harapan.

V. KESIMPULAN

Berdasarkan literatur-literatur yang telah dikumpulkan dan dilakukan analisis. Penelitian mengenai implementasi *web scraping* sangat populer dan mayoritas peneliti memiliki kebutuhan yang relatif sama dalam menggunakan *web scraping* sebagai metode untuk mengumpulkan data, meskipun mayoritas menggunakan metode *web scraping* melalui ekstraksi dokumen HTML.

Web scraping sebagai metode dalam pengumpulan data cukup banyak digunakan pada penelitian saat ini, khususnya penelitian yang membutuhkan otomatisasi dalam mengumpulkan data dari suatu *website* tertentu. Selain itu juga ditemukan bahwa metode API lebih efisien dalam segi waktu dalam melakukan *scraping* dibandingkan melakukan ekstraksi pada dokumen HTML. Hal ini karena metode API tidak perlu untuk memilih *selector* data yang ingin diambil. Namun hal ini berpengaruh pada data yang dihasilkan, bahwa metode ekstraksi pada dokumen HTML dapat melakukan modifikasi pada *selector* sehingga data yang diperoleh lebih spesifik dibandingkan dengan metode API.

DAFTAR PUSTAKA

[1] T. Rizaldi dan H. A. Putranto, "Perbandingan Metode Web Scraping Menggunakan CSS Selector dan Xpath Selector," *Teknika*, vol. 6, no. 1, hlm. 43–46, Nov 2017, doi: 10.34148/teknika.v6i1.56.

[2] F. Djiwadikusumah dan G. H. Irawan, "WEB SCRAPING SITUS E-COMMERCE MENGGUNAKAN TEKNIK PARSING DOM," 2021.

[3] F. Sembiring, D. Yudistyral, dan D. P. Sari, "Penerapan Teknik Scraping Python pada Website Marketplace Indonesia," vol. 2, no. 1.

[4] B. A. H. Hakim, A. S. Mujahidah, dan A. S. Rusydiana, "SENTIMENT ANALYSIS ON HALAL CERTIFICATION," *Harmoni*, vol. 21, no. 1, hlm. 78–93, Jun 2022, doi: 10.32488/harmoni.v21i1.609.

[5] A. Fauziah, D. S. Kusumo, dan I. L. Sardi, "Analisis Penggunaan Pada Web Content Generator Perbandingan Harga Barang di 5 E-Commerce Indonesia Menggunakan Metode Scraping".

[6] M. Levi, H. N. Palit, dan S. Rostianingsih, "Perbandingan Performa Tools Web Scraping pada Website dengan Data Statis dan Dinamis".

[7] A. S. Yondra, D. Triyanto, dan S. Bahri, "IMPLEMENTASI WEB SCRAPING UNTUK MENGUMPULKAN INFORMASI PRODUK DARI SITUS E-COMMERCE DAN MARKETPLACE DENGAN TEKNIK PEMROSESAN PARALEL," vol. 10, no. 01, 2022.

[8] F. R. Wibowo, D. S. Rusdianto, dan A. Arwan, "Pengembangan Sistem Pengumpulan Promo E-Commerce Berbasis Website Dengan Menerapkan Teknik Web Scraping Dalam Proses Pengambilan Data Promo".

[9] A. Rahmatulloh dan R. Gunawan, "Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar," *Indones. J. Inf. Syst.*, vol. 2, no. 2, hlm. 95–104, Feb 2020, doi: 10.24002/ijis.v2i2.3029.

[10] D. D. A. Yani, H. S. Pratiwi, dan H. Muhardi, "Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace," *J. Sist. Dan Teknol. Inf. JUSTIN*, vol. 7, no. 4, hlm. 257, Okt 2019, doi: 10.26418/justin.v7i4.30930.

[11] V. Mitra dan H. Sujaini, "Rancang Bangun Aplikasi Web Scraping untuk Korpus Paralel Indonesia - Inggris dengan Metode HTML DOM," vol. 5, no. 1, 2017.

[12] A. W. Murdiyanto dan A. Priadana, "Analysis of Web Scraping Techniques to Get Keywords Suggestion and Allintitle Automatically from Google Search Engines," *Compiler*, vol. 10, no. 2, hlm. 71, Nov 2021, doi: 10.28989/compiler.v10i2.1064.

[13] Y. Sahria, "Implementasi Teknik Web Scraping pada Jurnal SINTA Untuk Analisis Topik Penelitian Kesehatan Indonesia," 2020.

[14] M. R. Fikri, R. T. Handayanto, dan D. Irwan, "Web Scraping Situs Berita Menggunakan Bahasa Pemograman Python," *J. Stud. Res. Comput. Sci.*, vol. 3, no. 1, hlm. 123–136, Mei 2022, doi: 10.31599/jsrcs.v3i1.1514.

[15] S. Kusumo, "PENERAPAN WEB SCRAPING DESKRIPSI PRODUK MENGGUNAKAN SELENIUM PYTHON DAN FRAMEWORK LARAVEL," *JATISI J. Tek. Inform. Dan Sist. Inf.*, vol. 9, no. 4, hlm. 3426–3435, Des 2022, doi: 10.35957/jatisi.v9i4.2727.