

BAB II

KAJIAN PUSTAKA

2.1 Tinjauan Teori

Tinjauan teori menyajikan landasan konseptual dan teoritis yang digunakan sebagai dasar dalam pelaksanaan penelitian ini. Pembahasan pada subbab ini bertujuan untuk menjelaskan konsep, definisi, serta prinsip-prinsip utama yang berkaitan dengan topik penelitian secara sistematis dan terstruktur. Teori-teori yang dikaji mencakup web scraping, *Document Object Model* (DOM), Selenium beserta komponen utamanya, penggunaan selektor dalam pengambilan elemen web, serta statistika deskriptif sebagai pendekatan untuk analisis data. Dengan adanya tinjauan teori ini, diharapkan pembaca memperoleh pemahaman mengenai kerangka keilmuan yang mendasari metode, teknik, dan pendekatan yang digunakan, sehingga penelitian yang dilakukan memiliki dasar akademik yang kuat dan dapat dipertanggungjawabkan secara ilmiah.

2.1.1 Web Scraping

Secara teoretis, web scraping didefinisikan sebagai proses pengumpulan data dengan cara apa pun selain melalui interaksi program dengan API—atau tentu saja selain melalui akses manusia menggunakan peramban web. Cara yang paling umum digunakan adalah dengan membuat program otomatis yang mengirimkan permintaan ke server web, meminta data (biasanya dalam bentuk HTML dan berkas lain penyusun halaman web), lalu mengurai data tersebut untuk mengambil informasi yang dibutuhkan (Mitchel, 2024).

Web scraping membuka ruang yang lebih luas untuk memanfaatkan data yang ada di website dibandingkan hanya mengakses website melalui browser. Data ratusan halaman website yang dapat ditampilkan melalui browser dapat diakses dan disimpan ke dalam basis data terstruktur sehingga dapat digunakan untuk keperluan analisis lanjutan.

2.1.2 Document Object Model (DOM)

Document Object Model (DOM) merupakan antarmuka pemrograman (*programming interface*) untuk dokumen web. DOM merepresentasikan struktur dokumen HTML yang mewakili halaman web ke dalam memori dalam bentuk pohon logika (*logical tree*) yang memungkinkan dokumen tersebut diakses dan dimanipulasi oleh bahasa pemrograman seperti JavaScript (Mozilla Developer Network, 2024).

Pohon logika terdiri dari node dan objek. Setiap cabang pohon berakhir pada sebuah node, dan setiap node berisi objek. Melalui metode-metode yang disediakan oleh DOM, pengembang dapat mengakses pohon ini secara terprogram untuk mengubah struktur, gaya, atau konten dari dokumen tersebut. Selain itu, node-node dapat dilengkapi dengan event handler yang akan dieksekusi ketika suatu peristiwa (event) tertentu dipicu. Setiap dokumen memiliki representasi DOM-nya sendiri, seperti pada contoh dokumen HTML berikut:

```
<html lang="en">
  <head>
    <title>My Document</title>
  </head>
  <body>
    <h1>Header</h1>
    <p>Paragraph</p>
  </body>
</html>
```

Struktur DOM digambarkan secara hierarkis. Akar (*root*) dari pohon ini adalah objek dari dokumen. Dokumen ini bercabang menjadi *node* `<html>`, yang kemudian memiliki dua *node* anak (*child nodes*), yaitu `<head>` dan `<body>`. *Node* `<head>` memiliki anak `<title>`, yang mengandung *text node* "My Document". Demikian pula, `<body>` memiliki anak `<h1>` dan `<p>`, yang masing-masing juga mengandung *text node* "Header" dan "Paragraph". Struktur DOM inilah yang memungkinkan navigasi dan manipulasi elemen. Sebuah *node* dapat memiliki hubungan orang tua-anak (*parent-child*), saudara (*siblings*), dan sebagainya. Ketika peramban web (*web browser*) mengurai (*parse*) dokumen HTML, peramban membangun DOM *tree* ini dalam memori dan kemudian menggunakannya untuk menampilkan dokumen serta menyediakan landasan bagi skrip JavaScript untuk berinteraksi dengannya.

DOM merupakan fondasi utama dalam pengembangan web front-end yang interaktif. Pemahaman yang mendalam tentang DOM memungkinkan pengembang untuk:

1. Mengubah konten secara dinamis seperti memperbarui teks, gambar, atau struktur HTML halaman tanpa perlu memuat ulang.
2. Merespons interaksi pengguna seperti menangani *event* klik, ketik, atau geser untuk menciptakan pengalaman pengguna yang responsif.
3. Memanipulasi CSS seperti mengubah penampilan elemen secara real-time.
4. Melakukan pembaruan sebagian (*partial update*) pada halaman website.

2.1.3 Selenium

Selenium merupakan software yang awalnya dikembangkan sebagai alat pengujian otomatis pada website, tetapi kini juga banyak dimanfaatkan untuk *web scraping*, terutama untuk proses *scraping* yang memerlukan representasi akurat dari tampilan situs web sebagaimana muncul di browser. Cara kerjanya adalah dengan mengotomatisasi browser untuk membuka halaman, mengambil data yang dibutuhkan, dan melakukan tangkapan layar di dalam website (Mitchell, 2024).

Oleh karena itu, Selenium dipilih dalam penelitian ini karena website Glints.com memiliki konten yang ditampilkan secara dinamis melalui browser, sehingga proses pengambilan data memerlukan pemuatan halaman dan interaksi dengan elemen website secara langsung. Penggunaan Selenium memungkinkan proses *scraping* dilakukan dengan mengotomatisasi browser, sehingga data yang

diperoleh dapat merepresentasikan informasi sebagaimana ditampilkan kepada pengguna pada halaman Glints.com. Selain itu, Selenium dapat digunakan untuk menunggu proses pemuatan konten dan mengakses elemen halaman yang tidak langsung tersedia pada struktur HTML awal, sehingga lebih sesuai untuk mendukung proses pengumpulan data lowongan pekerjaan pada website Glints.com dalam penelitian ini.

Selenium sendiri tidak memiliki browser bawaan, sehingga harus dijalankan bersama browser pihak ketiga. Misalnya, jika digunakan dengan Chrome, maka akan muncul jendela Chrome yang secara otomatis membuka situs dan menjalankan instruksi sesuai kode program.

2.1.3.1 Komponen Utama Selenium

Secara arsitektural, Selenium terdiri atas tiga komponen utama:

1. Client Libraries, sebagai pustaka dalam berbagai bahasa pemrograman (Python, Java, C#, dan lain-lain) yang mengirim perintah ke WebDriver.
2. Browser Drivers, sebagai komponen perantara yang menerjemahkan instruksi Selenium menjadi operasi spesifik browser, seperti men-klik, mengetik, atau mengambil teks.
3. Real Browser Instance, sebagai browser yang berjalan sungguhan, bukan simulasi, sehingga mampu mengeksekusi JavaScript dan memuat konten dinamis.

2.1.3.2 WebDriver

WebDriver merupakan API dan protokol yang menyediakan antarmuka netral terhadap bahasa pemrograman untuk mengendalikan perilaku browser. Setiap browser memiliki implementasi WebDriver khusus yang disebut driver. Driver ini berfungsi sebagai penghubung yang menangani komunikasi antara Selenium dan browser, sekaligus meneruskan perintah dari Selenium ke browser terkait (Selenium Developments Partners, 2024).

Halaman web modern menggunakan JavaScript untuk memuat data secara dinamis. Oleh sebab itu, web scraper membutuhkan alat yang mampu mengeksekusi JavaScript, memuat seluruh struktur DOM, dan melakukan navigasi antar halaman secara stateful. Hal ini menjadi alasan penggunaan Selenium WebDriver dalam penelitian ini. WebDriver bekerja dengan mengirimkan instruksi ke browser melalui protokol WebDriver ke sebuah *driver* yang terkait dengan jenis browser tertentu (misal: ChromeDriver untuk Chrome), kemudian menerima respons berupa kondisi DOM yang telah diperbarui.

Karakteristik penting WebDriver:

1. Browser sebagai black-box

WebDriver tidak meniru perilaku browser, tetapi mengendalikan browser aktual sehingga seluruh JavaScript, AJAX, dan rendering halaman terjadi secara nyata.

2. Controller dan driver

Selenium berfungsi sebagai *controller*, sedangkan ChromeDriver bertindak sebagai *browser driver* yang menghubungkan instruksi Python dengan mesin browser.

3. Eksekusi real-time

WebDriver menjalankan script secara sinkron dengan perubahan halaman, sehingga sangat efektif untuk situs yang menggunakan mekanisme pemuatan dinamis.

Karakteristik WebDriver ini memungkinkan program scraper untuk menelusuri halaman lowongan pekerjaan, menunggu elemen muncul, dan mengekstrak informasi yang telah ter-render sempurna.

2.1.4 Selektor

Dalam melakukan web scraping menggunakan Selenium WebDriver, pemilihan elemen HTML (node dari struktur DOM) yang tepat merupakan langkah kritis untuk mengekstrak data yang diinginkan. Sehubungan dengan itu, selenium menggunakan metode `find_element()` dan `find_elements()` yang dapat menavigasi dan memilih elemen-elemen spesifik yang tepat dalam dokumen HTML melalui berbagai strategi seleksi (Mitchell, 2024).

`Find_element()` mengembalikan elemen pertama yang sesuai dengan kriteria selektor. Sedangkan `find_elements()` mengembalikan daftar semua elemen yang sesuai dengan kriteria selektor. Kedua metode ini menerima dua parameter

utama, yakni objek By yang menentukan strategi seleksi, dan string selektor yang berisi kriteria pencarian. Program scraper yang digunakan pada penelitian ini mengimplementasikan dua string selektor, yaitu XPATH dan CSS_SELECTOR.

2.1.4.1 Implementasi By.XPATH

Strategi seleksi By.XPATH digunakan untuk menangani elemen-elemen yang memerlukan navigasi kompleks dan pencarian berdasarkan konten teks. Berikut beberapa contoh implementasinya:

1. Seleksi Tombol Login Berdasarkan Teks

Kode di bawah ini menggunakan metode `find_element()` secara implisit melalui kondisi `element_to_be_clickable`. Selektor By.XPATH mencari elemen apa pun yang mengandung teks "Masuk dengan Email", sedangkan metode `contains()` memungkinkan pencarian parsial untuk mengakomodasi variasi teks.

```
email_btn = self.wait.until(
    EC.element_to_be_clickable((By.XPATH,
    "[contains(text(), 'Masuk dengan Email')]"))
)
email_btn.click()
```

2. Seleksi Field Password

Pada kode di bawah ini selektor By.XPATH digunakan untuk mencari elemen input yang memiliki atribut `type='password'` or `name='password'`. Penggunaan kombinasi kondisi bertujuan untuk memastikan elemen password dapat ditemukan meskipun terdapat variasi atribut.

```
password_field = self.driver.find_element(
    By.XPATH, "//input[@type='password' or @name='password']")
password_field.clear()
password_field.send_keys(self.password)
```

3. Koleksi Semua Job Cards:

Pada kode di bawah ini selektor By.XPATH digunakan untuk mencari semua elemen div yang class-nya mengandung 'CompactOpportunityCard'. Penggunaan metode contains() memungkinkan penangkapan elemen meskipun terdapat *multiple classes*.

```
job_cards = self.driver.find_elements(By.XPATH,
    "//div[contains(@class, 'CompactOpportunityCard')]")
```

2.1.4.2 Implementasi By.CSS_SELECTOR

Strategi seleksi By.CSS_SELECTOR diterapkan untuk elemen-elemen dengan pola class yang konsisten dan memerlukan seleksi yang lebih efisien. Berikut beberapa contoh implementasinya:

1. Ekstraksi Judul Pekerjaan

Pada kode di bawah ini selektor CSS dengan kombinasi *multiple patterns* (h2 a, [class='JobCardTitle']) digunakan sebagai *fallback mechanism* untuk menangkap judul lowongan.

```

title_elem = job_card.find_element(By.CSS_SELECTOR, "h2 a,
[class='JobCardTitle']")

data["Judul Pekerjaan"] = title_elem.text.strip()

```

2. Ekstraksi Informasi Gaji

Pada kode di bawah ini selector `By.CSS_SELECTOR` dengan atribut substring selector (`[class='SalaryWrapper']`) menargetkan elemen span yang class-nya mengandung 'SalaryWrapper'.

```

salary_elem = job_card.find_element(By.CSS_SELECTOR,
                                     "span[class='SalaryWrapper']")

data["Gaji"] = salary_elem.text.strip()

```

3. Koleksi Metadata Tags

Pada kode di bawah ini selektor `By.CSS_SELECTOR` digunakan untuk mengumpulkan semua div dengan class yang mengandung pattern 'TagStyle__TagContentWrapper', memungkinkan ekstraksi multiple tags sekaligus.

```

tags = job_card.find_elements(By.CSS_SELECTOR,
                              "div[class='TagStyle__TagContentWrapper']")

tags_text = [t.text.strip() for t in tags if t.text.strip()]

```

2.1.5 Pandas

Pandas merupakan pustaka *open-source* pada bahasa pemrograman Python yang digunakan untuk keperluan analisis data. Library ini memungkinkan Python

untuk mengelola data berbentuk tabel, menyerupai spreadsheet, sehingga proses pemuatan data, pemrosesan data, dan penggabungan data dapat dilakukan secara cepat dan efisien. Untuk mendukung kemampuan tersebut, Pandas memperkenalkan dua struktur data utama, yaitu Series dan DataFrame. DataFrame merepresentasikan keseluruhan data berbentuk tabel dua dimensi, sedangkan Series merupakan struktur satu dimensi yang dapat dipahami sebagai satu kolom dari DataFrame (Chen, 2022).

Menurut Chen (2022), ketika suatu analisis perlu diterapkan pada banyak kumpulan data, pendekatan berbasis pemrograman seperti Python dan library seperti Pandas memungkinkan proses tersebut dijalankan secara otomatis dan berulang. Sejalan dengan pernyataan tersebut, pada penelitian ini Pandas digunakan pada seluruh tahapan pemrosesan data, mulai dari pemuatan data hasil web scraping, pemrosesan data, dan penggabungan data.

2.1.6 Statistika Deskriptif

Statistika deskriptif merupakan teknik analisis yang digunakan untuk menggambarkan dan merangkum data suatu variabel secara bermakna. Statistika deskriptif tidak digunakan untuk menarik kesimpulan di luar data yang tersedia, tetapi dapat membantu dalam memahami dan menginterpretasikan data yang dianalisis (Starbuck, 2023). Ruang lingkup statistika deskriptif mencakup konsep-konsep dasar statistika seperti distribusi frekuensi, ukuran pemusatan data (*central tendency*), ukuran penyebaran data (*dispersion*), kemencengan distribusi (*skewness*), keruncingan distribusi (*kurtosis*), dan penyajian data dalam bentuk

grafik, seperti diagram batang, diagram garis, histogram, poligon frekuensi (*polygon*), ogive, angka indeks, dan deret waktu (Nalim & Turmudi, 2012).

2.1.6.1 Ukuran Pemusatan Data

Ukuran pemusatan data umumnya meliputi rata-rata, median, modus, kuartil, desil, dan persentil. Pada penelitian ini ukuran pemusatan data yang digunakan adalah rata-rata. Menurut Huwaida (2019), rata-rata (*average*) adalah nilai yang mewakili sehimpunan atau sekelompok data. Nilai rata-rata umumnya mempunyai kecenderungan terletak di tengah-tengah. Oleh karena itu rata-rata sering disebut ukuran kecenderungan memusat. Misal kita mempunyai variabel X , sebagai hasil pengamatan atau observasi sebanyak N kali, yaitu $X_1, X_2, X_3, \dots, X_N$, maka nilai rata-rata dapat dihitung menggunakan rumus di bawah ini:

$$\mu = \frac{1}{N} (X_1 + X_2 + X_3 + \dots + X_N)$$

Keterangan:

μ = Nilai rata-rata dihitung berdasarkan populasi.

2.1.6.2 Distribusi Frekuensi

Distribusi Frekuensi dapat diartikan sebagai rangkaian data angka yang menggambarkan pembagian frekuensi dari variabel yang sedang dijadikan objek penelitian. Distribusi frekuensi yang pembagian kelas-kelasnya didasarkan pada golongan data yang bersifat kuantitatif disebut sebagai distribusi frekuensi numerikal, sementara distribusi frekuensi yang pembagian kelas-kelasnya didasarkan pada golongan data yang bersifat kualitatif disebut sebagai distribusi frekuensi kategorikal (Nurhayati, 2024).

Frekuensi dalam distribusi data dibedakan menjadi frekuensi absolut dan frekuensi relatif. Frekuensi absolut merupakan jumlah kemunculan data pada suatu kelas tertentu. Rumus frekuensi absolut f_i dapat dinyatakan sebagai berikut:

$$f_i = \text{jumlah data pada kelas ke-}i$$

Keterangan:

$$f_i = \text{frekuensi absolut kelas ke-}i$$

Sedangkan frekuensi relatif merupakan perbandingan antara frekuensi absolut f_i kelas tertentu dengan jumlah seluruh data n yang dinyatakan dalam bentuk persentase. Frekuensi relatif digunakan untuk menunjukkan proporsi atau persentase kemunculan suatu kelas terhadap keseluruhan data penelitian.

Rumus frekuensi relatif dapat dinyatakan sebagai berikut:

$$f r_i = \frac{f_i}{n}$$

Keterangan:

$$f r_i = \text{frekuensi relatif kelas ke-}i$$

$$f_i = \text{frekuensi absolut kelas ke-}i$$

$$n = \text{jumlah seluruh data}$$

Apabila dinyatakan dalam bentuk persentase, maka rumus frekuensi relatif menjadi:

$$fr_i(\%) = \frac{f_i}{n} \times 100\%$$

Keterangan:

$fr_i(\%)$ = frekuensi relatif kelas ke- i dalam persentase

2.1.6.3 Ukuran Penyebaran Data

Ukuran penyebaran data merupakan ukuran yang menunjukkan besar kecilnya perbedaan data dari rata-rata. Ukuran penyebaran mampu melihat seberapa jauh penyimpangan nilai data dari nilai pusatnya atau ukuran yang mampu menunjukkan seberapa banyak nilai data yang berbeda dengan nilai pusatnya Nurhayati (2024).

Pada penelitian ini, ukuran penyebaran data yang digunakan adalah *interquartile Range* (IQR) karena efektif dalam mendeteksi pencilan (*outlier*) sebagaimana dinyatakan oleh Oluleye (2023). Outlier yang ditemukan akan dihapus untuk memastikan kewajaran distribusi data pada variabel numerik sebelum dilakukan analisis statistik deskriptif. Hal ini bertujuan agar nilai yang terlampaui jauh dari pusat data tidak mempengaruhi nilai rata-rata rentang gaji yang akan dianalisis. Nilai IQR diperoleh dari selisih antara kuartil ketiga (Q_3) yang menunjukkan batas 75% data terendah dan kuartil pertama (Q_1) yang menunjukkan batas 25% data terendah. IQR tidak terpengaruh pencilan (*outlier*) karena hanya berfokus pada 50% data di bagian tengah. Oleh karena itu, metode IQR banyak digunakan dalam proses identifikasi dan deteksi *outlier* pada suatu dataset (Oluleye, 2023).

Rumus IQR adalah sebagai berikut:

$$Q1 = \frac{1(n + 1)}{4}$$

$$Q3 = \frac{3(n + 1)}{4}$$

$$IQR = Q3 - Q1$$

Keterangan:

$Q1$ = Posisi kuartil pertama

$Q3$ = Posisi kuartil ketiga

n = Jumlah data

Setelah nilai IQR diperoleh, langkah selanjutnya adalah menentukan batas bawah (*lower bound*) dan batas atas (*upper bound*) untuk mendeteksi data pencilan menggunakan rumus berikut:

$$\text{Batas Bawah} = Q1 - (1,5 \times IQR)$$

$$\text{Batas Atas} = Q3 + (1,5 \times IQR)$$

Data dengan nilai lebih kecil dari batas bawah dikategorikan sebagai *outlier bawah*. Data dengan nilai lebih besar dari batas atas dikategorikan sebagai *outlier atas*. Konstanta 1,5 merupakan standar umum dalam metode IQR untuk menentukan batas nilai pencilan.

2.1.6.4 Langkah-Langkah Pemecahan Masalah Secara Statistik

Menurut Huwaida (2019), pemecahan masalah menggunakan pendekatan statistik deskriptif dapat dilakukan dengan langkah-langkah sebagai berikut:

1. Identifikasi masalah atau peluang

Tujuan tahap ini adalah memahami dan merumuskan masalah secara jelas. Informasi kuantitatif digunakan untuk mengetahui karakteristik, tingkat keparahan, serta dampak masalah, misalnya terhadap sumber daya seperti tenaga kerja, bahan, dana, dan waktu.

2. Pengumpulan data yang tersedia

Data yang digunakan harus akurat, relevan, lengkap, dan tepat waktu. Data dapat berasal dari:

- a. Data internal, yaitu data dari dalam organisasi (akuntansi, produksi, pemasaran, dll).
- b. Data eksternal, yaitu data dari luar organisasi (asosiasi, pemerintah, publikasi statistik, konsumen, pemasok).

3. Pengumpulan data baru (data orisinal)

Jika data yang tersedia belum mencukupi, maka perlu dilakukan pengumpulan data sendiri. Keuntungannya adalah data lebih sesuai dengan kebutuhan penelitian.

4. Pengklasifikasian dan peringkasan data

Data yang terkumpul dikelompokkan berdasarkan karakteristik tertentu dan dapat diberi kode (coding). Setelah itu, data diringkaskan dalam bentuk tabel, grafik, atau ukuran statistik deskriptif agar mudah dipahami dan dianalisis.

5. Penyajian data

Data disajikan dalam bentuk tabel, grafik, dan ukuran kuantitatif untuk membantu memahami masalah, mengidentifikasi hubungan antarvariabel, serta memudahkan penyampaian informasi kepada pihak terkait.

6. Analisis data

Data yang telah disajikan dianalisis menggunakan metode statistik. Hasil analisis diinterpretasikan untuk mencari alternatif solusi terbaik, dengan mempertimbangkan tujuan dan keterbatasan yang ada.

7. Penarikan kesimpulan dan pengambilan keputusan

Hasil analisis diinterpretasikan secara statistik untuk menentukan tindakan atau keputusan yang paling tepat dalam menyelesaikan masalah.

2.2 Penelitian Terdahulu

Penelitian-penelitian terdahulu yang berhubungan dengan web *scraping* dan relevan dengan penelitian ini telah dikumpulkan untuk dijadikan sumber kajian agar mendorong penulis untuk menampilkan kebaruan dan menjaga keaslian dalam penelitian ini. Penelitian-penelitian yang telah dikumpulkan itu mencakup berbagai topik seperti yang dapat dilihat pada tabel berikut ini:

Tabel II - 1. Penelitian Terdahulu

No	Judul penelitian, Nama Penulis, Tahun Publikasi	Permasalahan	Metode	Hasil Penelitian
1.	Indonesian Named	Peneliti	Web scraping	Pembuatan

	Entity Recognition untuk Domain Zakat Menggunakan Conditional Random Fields. Nur Febriana Widiyanti dkk (2023).	memerlukan data teks terkait domain zakat yang akan diekstraksi menggunakan metode Conditional Random Fields.	menggunakan bahasa pemrograman Python dan <i>library Scrapy</i> untuk parsing <i>DOM HTML</i> .	model Indonesian Named Entity Recognition menggunakan metode Conditional Random Fields
2.	Sistem Rekomendasi Pekerjaan	Peneliti memerlukan data lowongan	Web scraping dengan memanfaatkan	Pengembang an sistem rekomendasi

Tabel II - 2. Penelitian Terdahulu (lanjutan 1)

2.	Menggunakan Pendekatan Content-based Filtering. Alya Fitria (2024).	pekerjaan yang akan digunakan untuk membuat sistem rekomendasi berbasis konten.	chrome extension Web Scraper sebagai tools untuk melakukan scraping.	berbasis Content-based Filtering.
3.	Pemanfaatan Teknik Web Scraping Untuk Estimasi Tarif d Sarana Transportasi Bus Jakarta-Malang Pada Traveloka. Dewi Anjani, Dewi Novianti, dan Yogi Bachtiar (2025).	Peneliti memerlukan detail informasi sarana transportasi bus rute Jakarta ke Malang agar dapat memberikan rekomendasi harga bus yang ideal.	Webscraping menggunakan aplikasi web scraping Octoparse.	Tabel tarif dan jadwal perjalanan yang ditawarkan semua PO. Bus jurusan Jakarta Malang.
4	Web Scraping Data Ulasan Pelanggan untuk Kemajuan Bisnis E-Commerce pada	Peneliti memerlukan data ulasan pengguna aplikasi tokopedia untuk melakukan	Webscaping dengan menggunakan Bahasa pemrograman	Analisis sentiment ulasan pelanggan Tokopedia, yang menun-

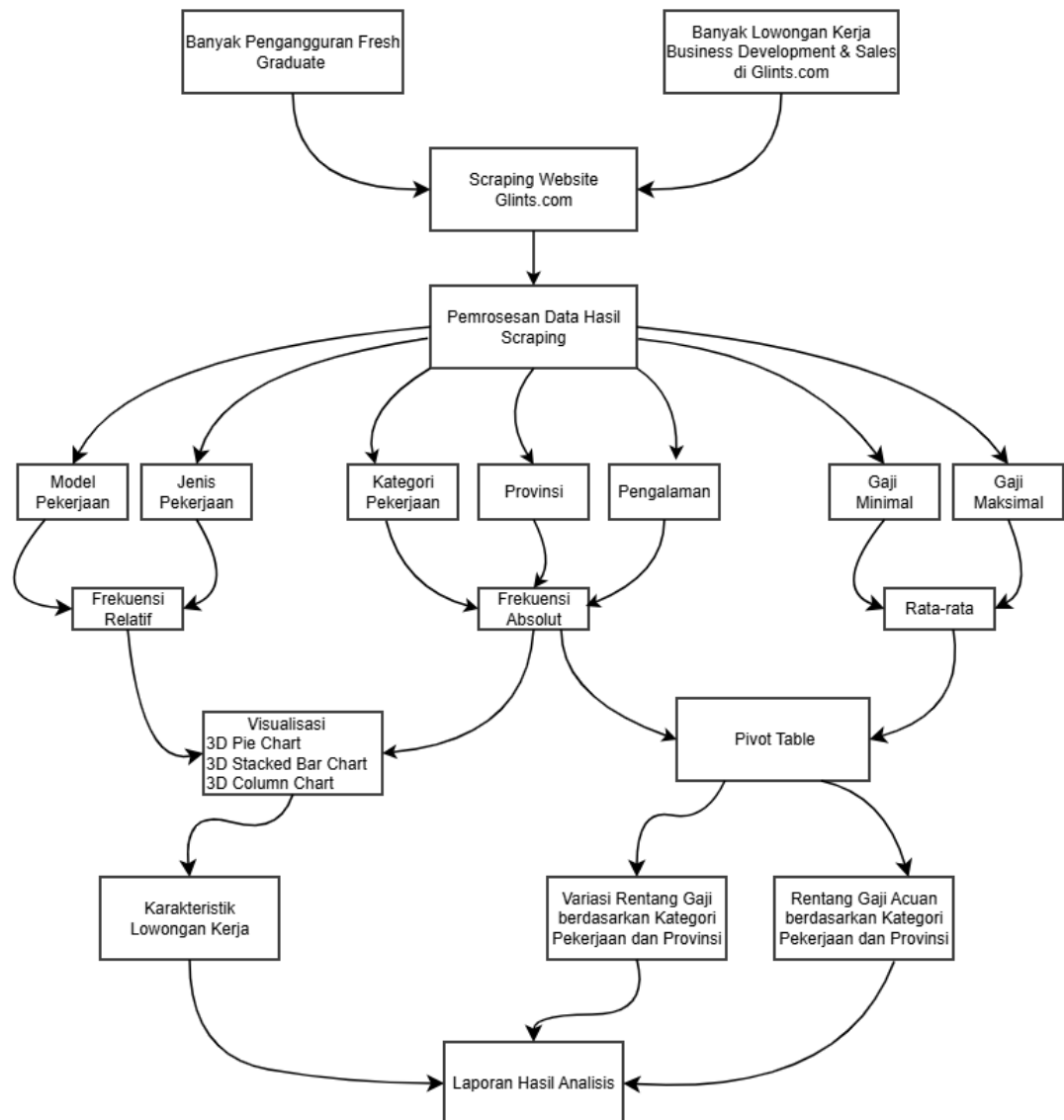
Tabel II - 3. Penelitian Terdahulu (lanjutan 2)

4.	Official Store dan Non-Official Store dengan Pendekatan Natural Language Processing. Muhamad Rafif Al Hafizh dkk (2025).	analisis sentimen dengan pendekatan Natural Language Processing.	Python dan <i>library BeautifulSoup</i> untuk parsing DOM HTML.	jukkan sentimen positif pelanggan brand tertentu lebih tinggi terhadap <i>official store</i> dibanding <i>unofficial store</i> dengan selisih sekitar 11%.
5.	Eksplorasi Pola Penjualan Produk Di E-Commerce Menggunakan Web Scraping. Deigo Anugrah Pratama, Kevin Rafael Manuela Feltya, dan Ibnu Rizal Mutaqin Muyasar (2025).	Peneliti memerlukan cara cepat dan intuitif untuk mengamati tren penjualan produk tertentu di Shopee.	Web scraping menggunakan pemrograman JavaScript yang dijalankan pada developer console browser untuk mengekstrak ulasan pelanggan.	Pengembangan dashboard pola penjualan produk Xiaomi Redmi Note 13 Pro+ 12/256 GB 5G berdasarkan data ulasan pelanggan.

2.3 Kerangka Berpikir

Menurut Syahputri, Fallenia, dan Syafitri (2023), kerangka berpikir merupakan dasar pemikiran dalam penelitian yang disusun melalui perpaduan antara teori, fakta, hasil observasi, dan kajian kepustakaan yang relevan dengan permasalahan penelitian. Kerangka berpikir memuat teori, dalil, atau konsep yang menjadi dasar penelitian serta menjelaskan variabel atau unsur penelitian secara mendalam dan relevan dengan permasalahan yang dikaji. Kerangka berpikir juga berfungsi untuk menggambarkan keterkaitan antarunsur penelitian sehingga dapat memberikan arah yang sistematis dalam pelaksanaan penelitian. Penyusunan kerangka berpikir perlu dilakukan secara logis dan sistematis dengan menggunakan teori serta referensi yang relevan dengan permasalahan penelitian. Selain disajikan dalam bentuk narasi, kerangka berpikir dapat digambarkan dalam bentuk bagan atau diagram untuk menunjukkan alur pemikiran peneliti dan keterkaitan antarvariabel atau unsur yang diteliti. Berdasarkan landasan teori, kajian kepustakaan, dan permasalahan yang telah diuraikan, kerangka berpikir dalam penelitian ini disusun sebagai dasar untuk menggambarkan alur penelitian serta keterkaitan antarunsur yang digunakan dalam menjawab permasalahan penelitian. Kerangka berpikir merupakan buatan peneliti sendiri (bukan dari peneliti lain), yakni cara peneliti berargumentasi dalam merumuskan hipotesis. Argumentasi itu harus analitis, sistematis, dan menggunakan teori yang relevan. Selanjutnya, menurut Sugiyono (2019), kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan beragam aspek yang sudah diidentifikasi.

Gambar II-1 menunjukkan secara visual kerangka berpikir yang disusun dalam penelitian ini:



Gambar II - 1. Kerangka Pemikiran

1. Titik tolak penelitian dimulai dari adanya fenomena pengangguran *fresh graduate* di Indonesia. Fenomena ini menjadi pijakan awal yang mendorong

perlunya dilakukan penelitian untuk memberikan gambaran pasar kerja yang objektif dalam rangka menemukan alternatif solusi permasalahan ini.

2. Pengumpulan data dilakukan dengan metode dokumentasi menggunakan teknik *web scraping* pada website Glints.com. Proses ini bertujuan untuk mengumpulkan data lowongan pekerjaan di bidang *business development* dan *sales* secara otomatis dan terstruktur.
3. Pemrosesan data dilakukan untuk menghasilkan variabel-variabel yang siap diolah menjadi bentuk visual atau tabel untuk dianalisis menggunakan konsep-konsep statistika deskriptif.
4. Pengolahan data ke dalam bentuk visual disesuaikan dengan konsep statistika deskriptif yang diterapkan pada kelompok variabel. Dalam penelitian ini variabel kategorik akan dianalisis berdasarkan distribusi frekuensinya, sedangkan variabel numerik akan dianalisis berdasarkan ukuran pemusatan datanya.
5. Visualisasi data dilakukan untuk menyajikan hasil analisis dalam bentuk yang lebih mudah dipahami. Jenis visualisasi yang digunakan meliputi: 3D Pie Chart (untuk jenis pekerjaan dan model pekerjaan), 3D Stacked Bar Chart (untuk kategori pekerjaan dan pengalaman), 3D Column Chart (untuk distribusi provinsi), serta Pivot Table (untuk rata-rata gaji berdasarkan kategori pekerjaan dan provinsi).

6. Hasil analisis yang terdiri dari tiga komponen utama: (a) Karakteristik Lowongan Kerja, yaitu gambaran umum tentang proporsi jenis pekerjaan dan model pekerjaan; distribusi lowongan berdasarkan pengalaman, provinsi, dan kategori pekerjaan; (b) variasi rentang gaji berdasarkan kategori pekerjaan dan provinsi, yaitu gambaran tentang perbedaan besaran gaji antar kategori pekerjaan dan antar wilayah; serta (c) rentang gaji acuan berdasarkan kategori pekerjaan dan provinsi, yaitu kisaran gaji minimal dan maksimal yang dapat dijadikan pedoman oleh *fresh graduate*.
7. Penyajian hasil dari seluruh rangkaian proses ini adalah Laporan Hasil Analisis yang berisi gambaran empiris tentang pasar kerja bidang *business development* dan *sales* untuk *fresh graduate* di Indonesia.

2.4 Keaslian Penelitian

Keaslian penelitian ini terletak pada perbedaan yang jelas dari sisi objek penelitiannya jika dibandingkan dengan penelitian-penelitian terdahulu. Penelitian ini fokus pada besaran rentang gaji pada lowongan pekerjaan bidang *business development* dan *sales* yang terbuka bagi *fresh graduate*. Penelitian Alya Fitria (2024) menggunakan data lowongan pekerjaan, tetapi tujuannya diarahkan pada pengembangan sistem rekomendasi berbasis *content-based filtering*, bukan pada analisis struktur gaji pada lowongan pekerjaannya. Sementara itu, penelitian-penelitian lain seperti Nur Febriana Widiyanti dkk. (2023), Dewi Anjani dkk. (2025), serta Muhamad Rafif Al Hafizh dkk. (2025) memanfaatkan data hasil scraping dari domain objek penelitian yang berbeda, seperti data zakat, data transportasi, dan ulasan pelanggan.

Keaslian penelitian ini juga ditunjukkan melalui pemilihan sumber data dan teknik pengumpulan data yang digunakan. Penelitian ini menggunakan data lowongan pekerjaan aktual yang dikumpulkan secara langsung dari platform Glints.com melalui teknik *web scraping* berbasis bahasa pemrograman Python dengan library Selenium. Pemilihan Selenium memungkinkan proses automasi login dan scrolling pada website dinamis, sehingga data yang diperoleh mencakup informasi penting seperti gaji minimal, gaji maksimal, kategori pekerjaan, jenis pekerjaan, model kerja, dan lokasi kerja. Pendekatan ini membedakan penelitian ini dari penelitian terdahulu yang atau hanya mengekstraksi data seperti teks ulasan atau teks lainnya.

Dari sisi pendekatan analisis, penelitian ini menitikberatkan pada analisis deskriptif untuk memetakan lanskap pasar kerja secara empiris. Berbeda dengan penelitian Deigo Anugrah Pratama dkk. (2025) yang menganalisis tren penjualan berdasarkan jumlah ulasan harian sebagai indikator, penelitian ini mengolah data lowongan kerja secara langsung berdasarkan hasil parsing dan ekstraksi elemen DOM HTML yang merepresentasikan informasi pekerjaan dan gaji secara eksplisit. Analisis dilakukan melalui penyajian visualisasi deskriptif, seperti *tree map*, *stacked bar chart*, dan *pie chart*, untuk menggambarkan distribusi kategori pekerjaan, variasi rentang gaji, distribusi kota, serta proporsi jenis dan model kerja.

Kebaruan lainnya terletak pada tujuan dan hasil penelitian yang secara spesifik ditujukan bagi *fresh graduate*. Penelitian ini tidak hanya bertujuan untuk menyajikan data, tetapi juga menghasilkan laporan analisis lanskap pasar kerja bidang *business development* dan *sales* pada periode 17 April hingga 8 Mei 2026 di platform Glints.com yang memberikan gambaran objektif dan realistis mengenai karakteristik lowongan pekerjaan dan variasi gaji, sehingga dapat dimanfaatkan sebagai referensi dalam menentukan ekspektasi gaji.