

Ecommerce-Price-Scraper: Pustaka Ekstraksi Harga E-Commerce Indonesia Melalui *Web Scraping*

Richard Owen Hoan^{1, a)}, Helmi^{2, b)}, Nur Ali Rajab^{3, c)}, Bhagavatgita Mahardika Kesuma Putra^{4, d)}, Gusti Ahmad Fanshuri Alfarisy^{5, e)}

^{1,2,3,4,5} Program Studi Informatika, Jurusan Matematika dan Teknologi Informasi, Institut Teknologi Kalimantan, Balikpapan.

^{a)} 11211075@student.itk.ac.id

^{b)} 11211043@student.itk.ac.id

^{c)} 11211067@student.itk.ac.id

^{d)} 11211023@student.itk.ac.id

^{e)} gusti.alfarisy@lecturer.itk.ac.id

Abstrak. Penelitian ini berfokus pada pengembangan *library* Python berbasis BeautifulSoup untuk melakukan *web scraping* secara otomatis pada platform *e-commerce* Lazada dan Bukalapak. Tujuan utama penelitian ini adalah untuk memberikan solusi praktis dalam ekstraksi harga produk secara efisien tanpa harus berpindah antar platform. Metodologi yang digunakan melibatkan integrasi BeautifulSoup dan Selenium untuk memastikan akurasi pengambilan data, meskipun struktur HTML situs sering berubah. Sistem yang dikembangkan dapat mengambil informasi seperti nama produk, harga, dan tautan produk berdasarkan kata kunci pencarian pengguna. Hasil pengembangan *library* menunjukkan bahwa *library* ini mampu mengotomatisasi proses pengambilan data dengan tingkat akurasi tinggi, sekaligus memberikan peluang untuk pengembangan aplikasi perbandingan harga yang lebih kompleks. Selain itu, kami menyediakan fungsi kepada pengembang untuk dapat diperluas ke lingkungan *e-commerce* yang lain. Namun, adaptasi terhadap perubahan struktur HTML *e-commerce* tetap menjadi tantangan yang perlu diperhatikan pada pengembangan selanjutnya. Diharapkan solusi ini dapat meningkatkan efisiensi belanja online bagi pengguna.

Kata Kunci: *web scraping*, *BeautifulSoup*, *e-commerce*, perbandingan harga, *library*, *python library*

1. Pendahuluan

Indonesia adalah negara kepulauan yang sangat luas dan salah satu negara dengan penduduk terbanyak di dunia. Jumlah penduduk Indonesia pada tahun 2024 mencapai sekitar 281.603.800 jiwa (Badan Pusat Statistik Provinsi Sulawesi Utara, 2024). Semakin banyak jumlah penduduk semakin tinggi juga kebutuhan pasar yang dibutuhkan baik primer maupun sekunder. Di era teknologi yang sudah semakin maju, mayoritas penduduk melakukan transaksi jual beli secara online, walaupun tidak jarang pasar tradisional juga masih tetap dilestarikan. Namun masyarakat Indonesia lebih mengutamakan kemudahan dan praktis dalam berbagai hal, sehingga belanja secara online semakin sering dilakukan. Adapun *e-commerce* yang sering dijumpai dalam melakukan transaksi jual beli online adalah Lazada, Bukalapak, dan lainnya.

Masyarakat Indonesia dalam melakukan pembelian secara online pada *e-commerce* sering melakukan perpindahan dari satu *e-commerce* ke *e-commerce* lain. Perpindahan antar *e-commerce* bertujuan untuk melihat perbandingan harga yang lebih murah dalam melakukan pembelian barang yang sama. Proses ini tidak hanya memakan waktu, tetapi juga menyulitkan pengguna dalam menentukan platform mana yang menawarkan harga terbaik. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah *library* yang dapat mempermudah proses pengambilan data harga barang dari berbagai platform *e-commerce*, khususnya Lazada dan Bukalapak.

Dalam penelitian ini, kami berfokus pada pengembangan library berbasis Python yang kami rancang untuk melakukan proses pengambilan data atau data scraping secara otomatis dari dua platform e-commerce tersebut. Salah satu komponen utama yang kami gunakan dalam pengembangan library ini adalah BeautifulSoup, sebuah library Python yang digunakan untuk memproses data HTML dan XML. BeautifulSoup memungkinkan peneliti atau praktisi untuk dengan mudah mengekstrak informasi tertentu dari halaman web, seperti nama barang, harga, dan tautan produk, dengan cara yang efisien dan terstruktur.

Pada dasarnya penelitian ini bertujuan untuk merancang dan mengembangkan sebuah library berbasis Python yang berfungsi sebagai dasar untuk melakukan proses web scraping secara otomatis dari dua platform e-commerce, yaitu Lazada dan Bukalapak. Library ini dirancang untuk memberikan solusi yang praktis bagi pengembang aplikasi lain yang membutuhkan data harga barang dari kedua platform tersebut. Dengan memanfaatkan teknologi seperti BeautifulSoup, library ini mampu mengambil informasi yang relevan berdasarkan kata kunci yang dimasukkan oleh pengguna, seperti nama barang, harga, dan tautan produk.

Penelitian ini memiliki pembatasan terkait jenis data yang dikumpulkan. Data yang diperoleh hanya berkisar pada nama produk, harga, dan link produk yang berkaitan dengan kata kunci yang dicari oleh pengguna. Dengan memperhatikan jenis data ini, penelitian ini menjamin bahwa proses pengumpulan data tetap efisien dan tepat tanpa menyertakan informasi tambahan yang tidak diperlukan.

Melalui pengembangan library ini, diharapkan dapat tercipta sebuah alat yang tidak hanya memudahkan pengambilan data secara otomatis, tetapi juga membantu pengembang menciptakan aplikasi yang lebih kompleks, seperti portal perbandingan harga atau rekomendasi produk. Dalam jangka panjang, penelitian ini diharapkan dapat mengurangi kesulitan yang sering dihadapi masyarakat dalam membandingkan harga barang secara manual antar platform, sehingga memberikan pengalaman belanja daring yang lebih efisien, praktis, dan hemat waktu.

2. Metode

Pada bab ini akan membahas beberapa poin penting terkait penelitian dalam membangun pustaka untuk melakukan scraping. Tujuannya adalah untuk mengekstraksi harga pada dua e-commerce yang ada di Indonesia: Bukalapak dan Lazada.

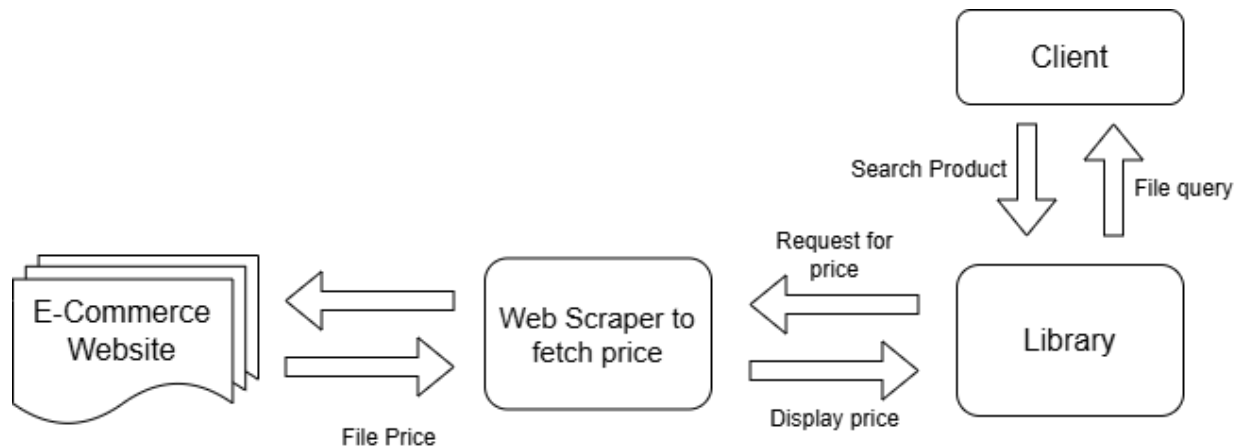
2.1 Gambaran Sistem

Gambaran sistem merupakan alur dari kinerja di balik layar terkait kinerja sistem dalam melakukan scraping. Dalam penelitian ini melibatkan perancangan arsitektur web scraping yang menggunakan pustaka bawaan dari bahasa pemrograman Python, yaitu BeautifulSoup untuk mengakses, mengambil, dan memproses data dari situs online yang merupakan e-commerce di Indonesia, antara lain Lazada dan Bukalapak. Diagram alur sistem termasuk bagaimana data dapat diambil, disimpan, dan diolah, berikut adalah gambaran sistem pada penelitian ini.

Sistem terdiri dari beberapa komponen utama. Pertama, E-Commerce Website menjadi sumber data yang diakses oleh web scraper untuk mengambil informasi produk. Data produk yang diambil ini diteruskan ke Library, yang berisi fungsi dan logika untuk memproses dan menganalisis data tersebut. Hasilnya kemudian dikirimkan ke Client, yang merupakan antarmuka pengguna atau aplikasi tempat data harga ditampilkan.

2.2 Query Processing

Pada tahap ini merupakan tahap awal untuk melakukan scraping dengan memasukkan query produk yang akan dicari. Pustaka menerima masukan berupa kata kunci pencarian sesuai format masing-masing platform dari dua e-commerce: Bukalapak dan Lazada.



GAMBAR 1. Gambaran Sistem

2.3 Data Retrieval

Pada tahap ini, sistem mulai mengakses situs e-commerce yang telah ditargetkan, seperti Lazada dan Bukalapak, untuk mengambil informasi terkait produk berdasarkan query yang diberikan oleh pengguna. Data retrieval menggunakan pustaka BeautifulSoup dilakukan untuk melakukan navigasi pada struktur HTML di situs web dan mengekstrak elemen-elemen tertentu seperti nama produk, harga, dan link (tautan) yang akan mengarah ke web browser pengguna. Langkah ini juga mencakup pemanfaatan teknik HTTP request untuk memastikan koneksi yang stabil serta pengambilan data yang akurat.

2.4 Data Parsing dan Filtering

Proses parsing data bertujuan untuk mengekstrak informasi yang relevan dari halaman web dengan struktur HTML yang kompleks. Pada tahap ini, kode menggunakan pustaka BeautifulSoup untuk memproses sumber halaman yang telah dirender oleh Selenium. Parsing dilakukan dengan mencari elemen-elemen HTML tertentu yang mengandung informasi penting, seperti nama produk, harga, dan tautan ke produk. Misalnya, nama produk diekstrak dari elemen `<p>` dengan kelas tertentu yang sesuai dengan struktur halaman web, sementara harga produk diambil dari elemen serupa dengan atribut kelas yang berbeda. Teknik ini memastikan bahwa hanya informasi yang dibutuhkan diambil dari halaman web, sehingga data yang dihasilkan lebih terstruktur dan mudah digunakan untuk analisis lebih lanjut. Filtering dilakukan untuk memastikan bahwa data yang dikumpulkan valid, relevan, dan siap digunakan. Langkah pertama dalam filtering adalah memverifikasi apakah elemen produk berhasil ditemukan di halaman web. Jika tidak ada elemen produk yang ditemukan, program akan memberikan pesan kesalahan. Proses parsing dan filtering ini berperan penting dalam memastikan kualitas dari data yang didapatkan dari hasil scraping yang dilakukan oleh pustaka.

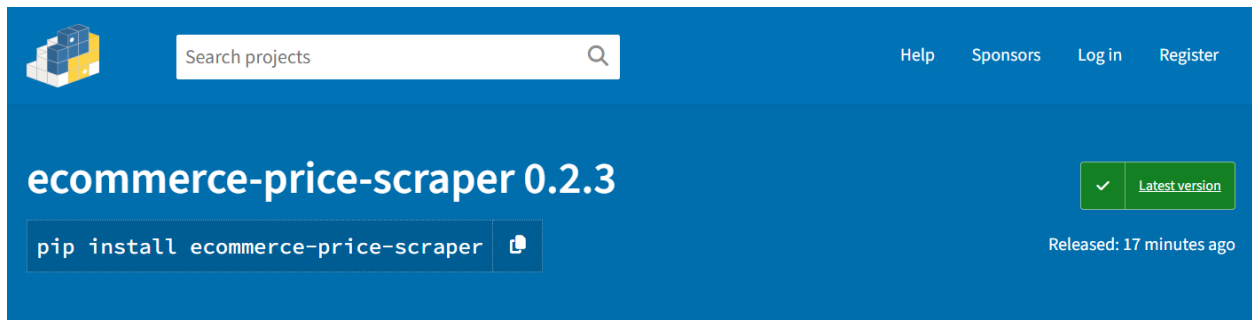
2.5 Output Formatting

Setelah data berhasil diambil, sistem memformat *output* agar mudah dipahami oleh pengguna. Data disajikan dalam bentuk objek *json* yang mencakup nama produk, harga, dan tautan. Format *output* dirancang agar tersusun secara terstruktur dan mudah dipahami pada hasil data yang diperoleh.

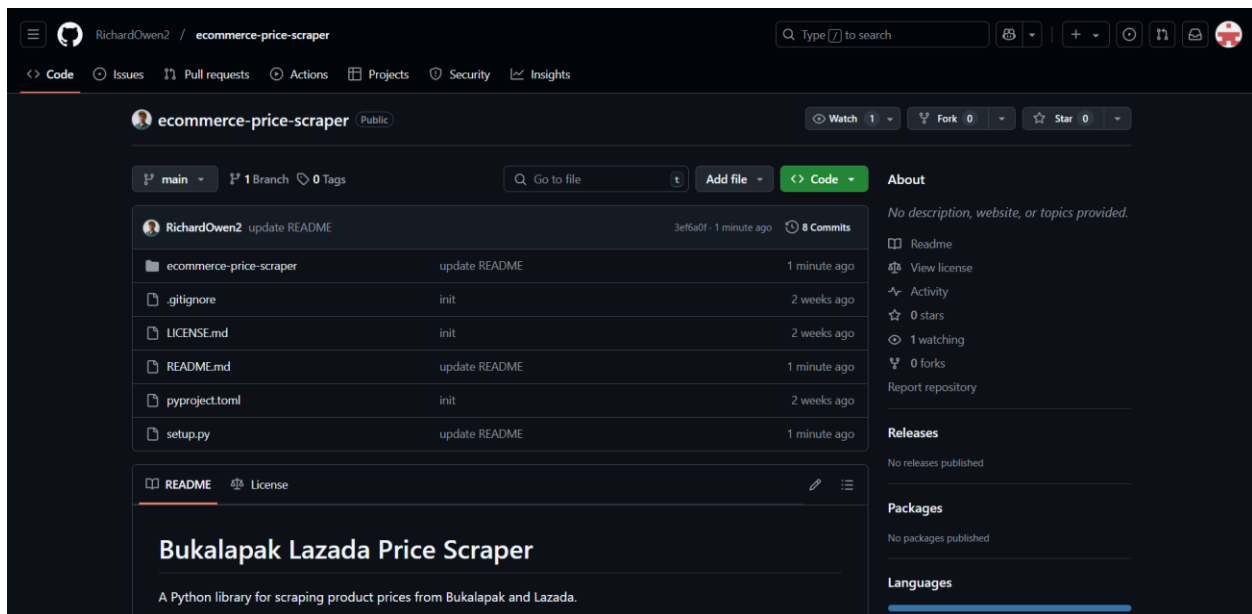
2.6 Potensi Perbandingan Harga

Tahap akhir ini adalah tahap yang dilakukan diluar sistem. Artinya adalah pengguna melakukan peninjauan sendiri dalam membandingkan harga dari hasil *scraping* yang sudah dilakukan, terdapat tiga data yang dapat dilihat, yaitu nama produk, harga, dan tautan. Pada tautan dapat diklik dan akan diarahkan ke *web browser* pengguna baik itu tautan dari Bukalapak atau Lazada. Sehingga pengguna dapat membuat keputusan dengan lebih cepat dan efisien.

3. Hasil dan Pembahasan



Gambar 2. *Library Deployment*



Gambar 3. *Github Library*

Pada pembahasan ini adalah hasil dari pengembangan sistem yang sudah dilakukan dalam membuat dua sistem *scraping* di masing-masing *web e-commerce* di Indonesia, yaitu Bukalapak dan Lazada. Pada seksi ini juga akan menjelaskan penggunaan pustaka yang sudah dibuat.

Adapun *library* (pustaka) yang sudah dibuat telah terpublikasi di pypi.org yang merupakan sumber pustaka terbuka dari python yang dapat diakses secara gratis, serta kode lengkap dari pembuatan pustaka ini juga diunggah pada github yang dapat dilihat dan mengikuti panduan penggunaan pustaka yang sudah dibuat. pustaka dapat dilihat pada **Gambar 2** dan **Gambar 3**.

3.1 Penggunaan Pustaka

Penggunaan pustaka sendiri dapat diakses di dua sumber baik dari pypi.org dan github, berikut adalah tautan yang mengarah ke halaman *web* tersebut. Berikut masing-masing tautan pustaka.

pypi : <https://pypi.org/project/ecommerce-price-scraper>

github : <https://github.com/RichardOwen2/ecommerce-price-scraper>

1) Source Code

Adapun tahapan awal adalah membuka IDE (*Integrated Development Environment*), seperti Vscode atau IDE lainnya yang dapat menjalankan program tersebut.

Tabel 1. Contoh penggunaan pustaka untuk mendapatkan informasi barang dan harga

Contoh <i>Source Code</i>	
1	<code>from ecommerce-price-scraper import search</code>
2	
3	<code>search_query = "iphone"</code>
4	
5	<code># Lazada</code>
6	<code>lazada_results = search('lazada', search_query)</code>
7	<code>print(lazada_results)</code>
8	
9	<code># Bukalapak</code>
10	<code>bukalapak_results = search('bukalapak', search_query)</code>
11	<code>print(bukalapak_results)</code>

2) Install Package

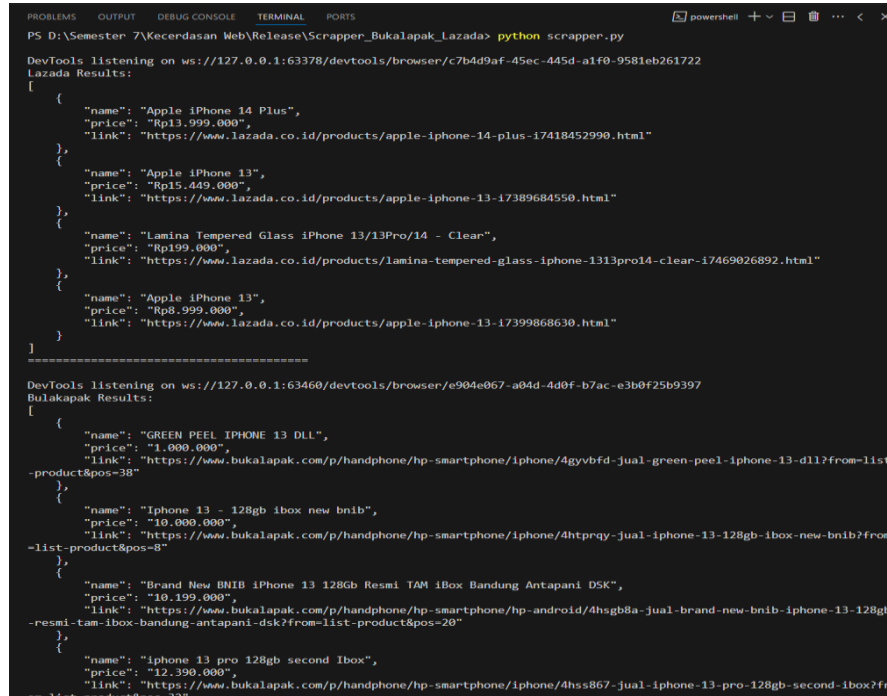
Tahap berikutnya adalah melakukan *install* dengan menjalankan kode pip install ecommerce-price-scraper pada terminal IDE. Jika berhasil maka cek menggunakan kode pip list untuk melihat hasil package yang sudah di-*install*.

Package	Version
-----	-----
attrs	24.2.0
beautifulsoup4	4.12.3
certifi	2024.8.30
cffi	1.17.1
charset-normalizer	3.4.0
ecommerce-price-scraper	0.2.3
h11	0.14.0

Gambar 4. Pip list

3) *Input Query*

Tahap terakhir adalah menjalankan program dengan terlebih dahulu memasukkan *query* barang yang ingin dicari sesuai pada baris 3 pada **Tabel 1**. Jika sudah, maka hasil program yang dijalankan seperti pada **Gambar 5** yang di mana akan menampilkan *list item* berdasarkan pencarian barang yang diinginkan dari situs Lazada dan Bukalapak



```

PS D:\Semester 7\Kecerdasan Web\Release\Scraper_Bukalapak_Lazada> python scrapper.py
DevTools listening on ws://127.0.0.1:63378/devtools/browser/c7b4d9af-45ec-445d-a1f0-9581eb261722
Lazada Results:
[
  {
    "name": "Apple iPhone 14 Plus",
    "price": "Rp13.999.000",
    "link": "https://www.lazada.co.id/products/apple-iphone-14-plus-17418452990.html"
  },
  {
    "name": "Apple iPhone 13",
    "price": "Rp15.449.000",
    "link": "https://www.lazada.co.id/products/apple-iphone-13-17389684550.html"
  },
  {
    "name": "Lamina Tempered Glass iPhone 13/13Pro/14 - Clear",
    "price": "Rp199.000",
    "link": "https://www.lazada.co.id/products/lamina-tempered-glass-iphone-1313pro14-clear-17469026892.html"
  },
  {
    "name": "Apple iPhone 13",
    "price": "Rp8.999.000",
    "link": "https://www.lazada.co.id/products/apple-iphone-13-17399868630.html"
  }
]
=====
DevTools listening on ws://127.0.0.1:63460/devtools/browser/e984e067-a04d-4d0f-b7ac-e3b0f25b9397
Bukalapak Results:
[
  {
    "name": "GREEN PEEL IPHONE 13 DLL",
    "price": "1.000.000",
    "link": "https://www.bukalapak.com/p/handphone/iphone/4gyvbf-d-jual-green-peel-iphone-13-d11?from=list-product&pos=38"
  },
  {
    "name": "Iphone 13 - 128gb ibox new bnib",
    "price": "10.000.000",
    "link": "https://www.bukalapak.com/p/handphone/iphone/4htprqy-jual-iphone-13-128gb-ibox-new-bnib?from=list-product&pos=8"
  },
  {
    "name": "Brand New BNIB iPhone 13 128Gb Resmi TAM iBox Bandung Antapani DSK",
    "price": "10.199.000",
    "link": "https://www.bukalapak.com/p/handphone/iphone/4hs867-jual-brand-new-bnib-iphone-13-128gb-resmi-tam-ibox-bandung-antapani-dsk?from=list-product&pos=20"
  },
  {
    "name": "Iphone 13 pro 128gb second Ibox",
    "price": "12.390.000",
    "link": "https://www.bukalapak.com/p/handphone/iphone/4hs867-jual-iphone-13-pro-128gb-second-ibox?from=list-product&pos=32"
  }
]

```

Gambar 5. Hasil program

4) *Dynamic function*

Kode yang ditunjukkan pada **Tabel 2** merupakan contoh fungsi dinamis untuk scraping data Pencarian *e-commerce* lainnya. Fungsi menerima parameter pencarian produk (*query*), membuka halaman *marketplace* dengan Selenium, dan menunggu elemen harga dimuat sepenuhnya. Setelah itu, halaman diambil dan diproses menggunakan BeautifulSoup untuk menemukan nama, harga, dan tautan produk. Jika data produk ditemukan, informasi disusun dalam bentuk daftar yang berisi nama, harga, dan tautan produk, lalu diurutkan berdasarkan harga. Fungsi ini juga memiliki penanganan kesalahan untuk menangkap error selama proses scraping dan memastikan Selenium ditutup dengan aman.

Tabel 2. *Dynamic Function*

<i>Dynamic Function</i>	
1	<code>def some_marketplace_function(driver, init_soup, query):</code>
2	<code>url = f'https://www.bukalapak.com/products?search[keywords]={search_query}'</code>
3	
4	<code>try:</code>

```

5  # Open the Marketplace url
6  driver.get(url)
7
8  # Wait for the price elements to be loaded
9  WebDriverWait(driver, 30).until(
10     EC.presence_of_all_elements_located((By.CLASS_NAME, 'bl-product-card-new__price'))
11 )
12
13 # Get the page source after JS is rendered
14 page_source = driver.page_source
15
16 # Initialize Beautiful soup
17 soup = init_soup(page_source, 'html.parser')
18
19 # Search Product Logic
20 product_cards = soup.find_all('div', class_='bl-product-card-new__wrapper')
21
22 if not product_cards:
23     return {"error": "No product cards found. Page might not have loaded correctly."}
24
25 for card in product_cards:
26     name_tag = card.find('p', class_='bl-text bl-text--body-14 bl-text--secondary bl-text--ellipsis_2')
27     product_name = name_tag.text.strip() if name_tag else "No product name"
28
29     price_tag = card.find('p', class_='bl-text bl-text--semi-bold bl-text--ellipsis_1 bl-product-card-new__price')
30     product_price = price_tag.text.strip() if price_tag else "No price available"
31
32     link_tag = card.find('a', class_='bl-link')
33     product_link = link_tag["href"] if link_tag else "No link available"
34
35     products.append({
36         'name': product_name,
37         'price': product_price,
38         'link': product_link
39     })
40
41 products.sort(key=lambda x: x['price'])
42
43 return products
44 except Exception as e:
45     return {"error": str(e)}
46 finally:
47     driver.quit()

```

4. KESIMPULAN

Penelitian ini menghasilkan pengembangan *library* Python berbasis *BeautifulSoup* yang dirancang untuk melakukan *web scraping* secara otomatis, dengan fokus pada pengambilan data harga produk dari platform *e-commerce* Lazada dan Bukalapak. Sistem ini dirancang untuk membantu pengguna membandingkan harga secara langsung tanpa perlu berpindah antar platform, sehingga lebih menghemat waktu dan tenaga. Dengan dukungan integrasi *BeautifulSoup* dan *Selenium*, sistem ini mampu menangani halaman *web* yang memuat konten melalui struktur *body* pada HTML pada *web e-commerce*. Selain itu,

library ini memberikan peluang bagi pengembang untuk membangun aplikasi lanjutan, seperti portal perbandingan harga yang lebih kompleks. Namun, hasil penelitian ini juga mengungkapkan bahwa perubahan pada struktur HTML platform *e-commerce* dapat mempengaruhi performa scraping, sehingga disarankan adanya pengembangan lebih lanjut untuk meningkatkan kemampuan adaptasi sistem terhadap perubahan tersebut. Solusi yang ditawarkan diharapkan dapat mempermudah pengguna dalam membandingkan harga produk serta meningkatkan efisiensi belanja online.

DAFTAR PUSTAKA

- [1] Arentze, T. A. and Timmermans, H. J. P. (2003) 'Modelling agglomeration forces in urban dynamics: a multi-agent approach' in *Proceedings of the Computers in Urban Planning and Urban Management Conference*, Sendai, Japan: CUPUM.
- [2] Cross, M. (2004) 'A graphic glimpse of urban planning', *The Guardian*, available from - <http://www.guardian.co.uk/society/2004/jan/28/regeneration.epublic?INTCMP=SRCH>
- [3] Hamnett, S. (2000) 'The late 1990s: Competitive versus sustainable cities' in S. Hamnett and R. Freestone (Eds.) *The Australian Metropolis: A Planning History*, pp: 168-188, Crow's Nest (NSW): Allen and Unwin.
- [4] Harper, T. L., Hibbard, M., Costa, H. and Yeh, A. G. (Eds.) (2010) *Dialogues in Urban and Regional Planning* (Volume 4), London: Routledge.
- [5] Yaakup, A., Bakar, Y. A., Kadir, M. N. A. and Sulaiman S. (2004) 'Computerised development control and approval system for City Hall of Kuala Lumpur', *Geo-Spatial Information Science*, Vol. 7, No. 1: 39-49.