

PENGEMBANGAN APLIKASI WEB SCRAPING UNTUK PENGUMPULAN DATA HARGA PRODUK E-COMMERCE MENGGUNAKAN PYTHON (STUDI KASUS: TOKOPEDIA)

Vicky Alhafiz¹, Mochamad Adhari Adiguna²

^{1,2}Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia, 15310
e-mail: ¹vickyalhafiz@gmail.com, ²dosen01864@unpam.ac.id

Abstract

Development of a web scraping application system for collecting e-commerce product price data using Python designed to automatically retrieve product data from the Tokopedia e-commerce platform in order to help sellers prepare sales strategies and market competition which usually require excessive time and resources. A sales strategy is a guideline for seeking innovation to run a business, and can determine benchmarks for success as a seller. This research aims to design a system that is able to work automatically and efficiently to provide accurate data and help sellers to analyze market conditions. The design of this system was carried out using the waterfall development method and using the Python as a programming language. The results of this research will provide accurate data and a dashboard that helps sellers analyze the market from the Tokopedia platform. Application testing also ensures that it has been carried out for accurate knowledge and that the resulting application is more user friendly and the resulting system meets the existing flexibility criteria. The results of all these tests also can be concluded that the resulting application is quite good although there is still a lot of development needed in terms of processing speed and analysis results displayed on the dashboard.

Keywords: Web Application Scraping; Data Collection; E-commerce; Product Datas; Python

Abstrak

Pengembangan sistem aplikasi web scraping untuk pengumpulan data harga produk e-commerce menggunakan python dirancang untuk mengambil data produk secara otomatis dari platform e-commerce Tokopedia agar dapat membantu penjual untuk membantu menyiapkan strategi penjualan maupun kompetisi pasar yang biasanya memerlukan waktu dan sumber daya yang berlebihan. Strategi penjualan merupakan salah satu pedoman dalam mencari inovasi untuk menjalankan bisnis, serta dapat menentukan tolak ukur keberhasilan sebagai penjual. Penelitian ini bertujuan untuk merancang sebuah sistem yang mampu bekerja otomatis dan efisien untuk memberikan data yang akurat dan membantu penjual untuk menganalisa kondisi pasar. Perancangan sistem ini dilakukan dengan metode pengembangan waterfall dan menggunakan bahasa pemrograman Python. Hasil penelitian ini akan memberikan data yang akurat dan dashboard yang membantu penjual untuk menganalisa pasar dari platform Tokopedia. Pengujian aplikasi juga telah dilakukan demi mengetahui akurasi serta memastikan aplikasi yang dihasilkan lebih user friendly dan sistem yang dihasilkan sudah memenuhi kriteria fleksibilitas yang ada. Hasil dari keseluruhan pengujian ini dapat disimpulkan bahwa aplikasi yang dihasilkan sudah cukup baik walaupun masih banyak butuh pengembangan dari segi kecepatan memproses maupun hasil analisa yang ditampilkan didashboard.

Kata kunci: Aplikasi Web Scraping; Pengumpulan Data; E-Commerce; Data Produk; Python

1. PENDAHULUAN

Dalam era digital saat ini, e-commerce atau Marketplace telah menjadi salah satu sektor utama dalam dunia perjual-belian, Pertumbuhan pesat e-commerce saat ini juga telah menciptakan tantangan baru, seperti kebutuhan untuk mengumpulkan dan menganalisis data khususnya data harga produk secara efisien. Dalam konteks ini, pengembangan aplikasi atau tools dengan menggunakan bahasa pemrograman Python menjadi salah satu kebutuhan untuk mencapai hal itu.

Menyadari hal tersebut, banyak pengembang atau developer banyak berlomba – lomba menciptakan aplikasi scraping guna menghindari proses pengumpulan data dan analisa yang memakan waktu dan sumber daya yang cukup besar. Oleh karena itu, penggunaan teknik web scraping dapat memberikan solusi yang sangat efisien untuk menjalani tugas tersebut. Web scraping sendiri merupakan sebuah teknik yang memungkinkan aplikasi untuk mengakses dan mengekstrak informasi secara terstruktur.

Python dipilih sebagai bahasa pemrograman utama untuk pengembangan aplikasi atau tools ini karena memiliki beragam pustaka dan kerangka kerja yang memampuni untuk melakukan web scraping. Pustaka seperti BeautifulSoup dan Selenium memungkinkan pengembang untuk dengan mudah mengambil informasi yang diperlukan dari kode HTML halaman web e-commerce.

2. METODE

Metodologi penelitian ini akan membimbing langkah-langkah yang akan diambil untuk mengembangkan aplikasi web scraping dan mengumpulkan harga produk e-commerce menggunakan Python. Metodologi ini mencakup tahapan pengembangan, pengumpulan data, dan analisis. Berikut adalah rincian metodologi penelitian yang diterapkan:

2.1. Metode pengumpulan data

Metode pengumpulan data yang digunakan dalam penelitian ini dalam rangka untuk memperoleh informasi sebagai berikut:

1. Observasi
Melakukan studi literatur untuk memahami konsep web scraping, bahasa Python, perpustakaan BeautifulSoup, Selenium, dan e-commerce. Lalu dilanjutkan untuk menganalisis teknik web scraping yang ada dan prinsip-prinsip dasar yang terkait.
2. Studi Pustaka
Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis.

2.2. Metode Pengembangan Sistem

Metode pengembangan perangkat lunak yang digunakan adalah metode waterfall, di mana pengembangan sistem dilakukan secara berurutan dan berjenjang melalui serangkaian tahapan yang terdefinisi dengan jelas. Pendekatan ini dimulai dengan mendefinisikan kebutuhan lengkap untuk sistem secara menyeluruh sebelum memulai pengembangan. Tahapan-tahapan pengembangan kemudian dilakukan secara berurutan, dimulai dari analisis kebutuhan, perancangan, implementasi, pengujian, integrasi, dan akhirnya masuk ke dalam tahap pemeliharaan.[1]

2.2.1 Komponen Pengembangan Sistem

- a. Python
Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar.[2]
Bahasa pemrograman Python sering digunakan dalam pengembangan aplikasi web scraping karena memiliki pustaka-pustaka yang kuat untuk mengakses dan mengolah data dari halaman web. Keunggulan Python dalam pemrosesan data membuatnya cocok untuk pengumpulan data harga produk e-commerce.[3]
- b. Pycharm
PyCharm adalah lingkungan pengembangan terintegrasi (IDE) yang digunakan dalam pemrograman komputer, yang memberikan berbagai macam alat atau tools untuk Python developer. Aplikasi ini dikembangkan oleh sebuah perusahaan asal Ceko bernama JetBrains. PyCharm juga mendukung HTML, CSS, JavaScript, TypeScript, serta database SQL.[4]

c. Website

Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang diseluruh dunia.[5] Website berisikan keseluruhan halaman - halaman web yang terdapat dari sebuah domain yang mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan.[6]

d. Browser

Browser adalah perangkat lunak yang digunakan untuk mengakses dan menampilkan konten web. Browser memungkinkan pengguna untuk menavigasi, membaca, dan berinteraksi dengan halaman web.[7] Browser bekerja dengan cara menginterpretasikan dan mengeksekusi kode HTML, CSS, dan JavaScript untuk menampilkan konten web secara visual kepada pengguna. [8]

e. Tokopedia

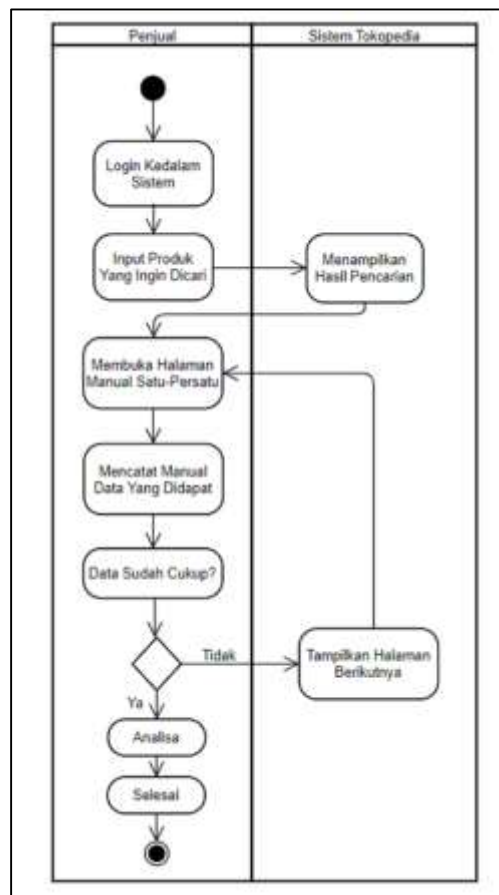
Tokopedia merupakan perusahaan teknologi Indonesia dengan misi pemerataan ekonomi secara digital di Indonesia. Hingga saat ini, Tokopedia termasuk marketplace yang paling banyak dikunjungi oleh masyarakat Indonesia.[9]

f. MySQL

MySQL adalah salah satu jenis database server yang terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses database nya. Selain itu, MySQL bersifat free (tidak perlu membayar untuk menggunakannya) pada berbagai platformnya.[10]

2.2.2 Analisa Sistem Berjalan

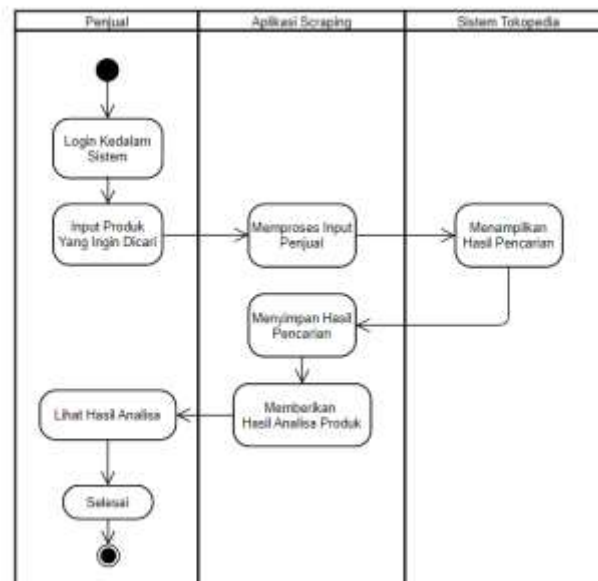
Dalam penelitian ini adalah kerabat peneliti yang berprofesi sebagai penjual dalam platform tersebut. Dapat dilihat disini proses manual yang digunakan pada sistem berjalan yang memakan waktu cukup lama, serta proses seperti ini seharusnya bisa dilakukan simplikasi untuk melakukan pengumpulan data mengingat diperlukannya strategi yang tepat dalam bersaing dalam platform e-commerce dan membutuhkan waktu yang cepat dalam menganalisanya.



Gbr 1. Analisa Sistem Berjalan

2.2.3 Analisa Sistem Usulan

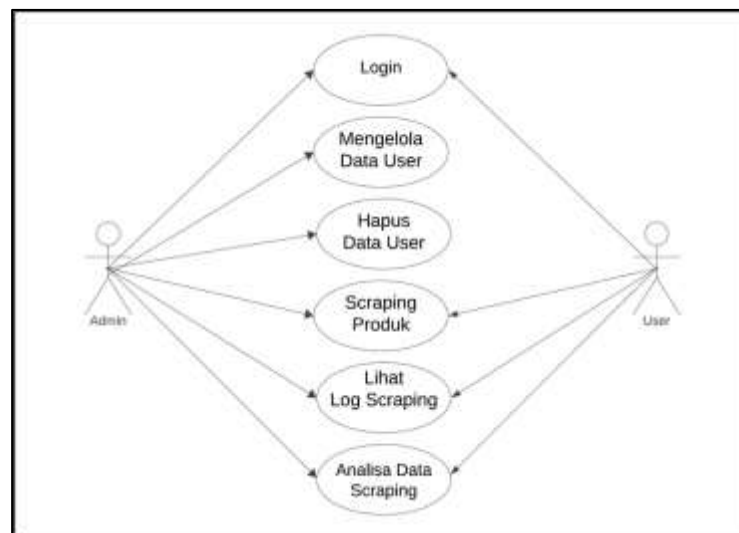
Perancangan Aplikasi yang diusulkan juga dideskripsikan dengan model Activity Diagram dengan rancangan aplikasi sebagai berikut guna meningkatkan efisiensi dan tingkat akurasi analisa data yang dibutuhkan.



Gbr 2. Analisa Sistem Berjalan

2.2.4 Use Case Diagram

Use Case Diagram merupakan salah satu dari beragam jenis diagram dalam UML (Unified Modeling Language) yang memvisualisasikan interaksi antara sistem dan pihak yang terlibat (aktor). Diagram Use Case mampu menjelaskan jenis interaksi antara pengguna sistem dan sistem itu sendiri [11]. UML merupakan sebuah standar bahasa yang umum digunakan dalam industri untuk menggambarkan kebutuhan, melakukan analisis dan desain, serta merancang arsitektur dalam pengembangan perangkat lunak berbasis objek [12].



Gbr 3. Use Case Diagram

Dari diagram *use case* diatas dapat disimpulkan bahwa:

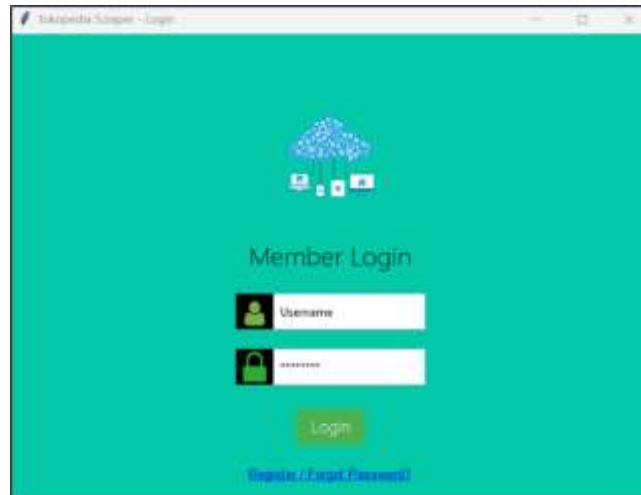
Admin dapat login, mengelola data user, hapus data user, scraping produk, lihat log scraping, analisa data scraping. Sedangkan user hanya dapat scraping produk, lihat log scraping, dan analisa data scraping saja.

3. HASIL

Antarmuka pengguna (*User Interface/UI*) adalah titik kontak antara manusia dan mesin, yang memungkinkan pengguna untuk berinteraksi dengan sistem komputer atau perangkat lainnya.[13] Desain UI yang efektif memastikan bahwa pengguna dapat dengan mudah berinteraksi dengan sistem, memahami informasi yang disajikan, dan menyelesaikan tugas-tugas mereka dengan cepat dan efisien.

3.1. Login

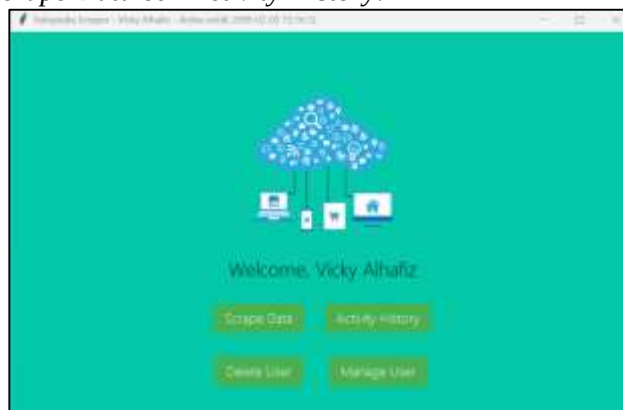
Antarmuka ini merupakan tampilan yang harus dilewati pengguna untuk mengakses aplikasi. Sebelum masuk ke menu utama, seluruh pengguna akan diwajibkan untuk mengisi username dan password pada menu login.



Gbr 4. Antarmuka Login

3.2. Menu Utama

Pada antar muka menu utama berisikan menu-menu yang bisa diakses sesuai dengan akses pengguna, jika pengguna merupakan admin tampilan tombol menunya berisikan empat menu yaitu *Scrape Data*, *Activity History*, *Delete User*, dan *Manage User*. Jika pengguna tidak memiliki akses admin tombol yang muncul hanya dua yaitu *Scrape Data* dan *Activity History*.



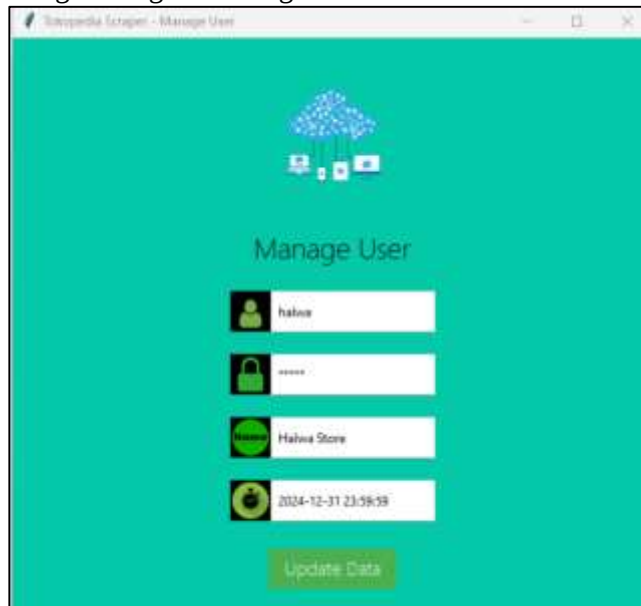
Gbr 5. Antarmuka Login Admin



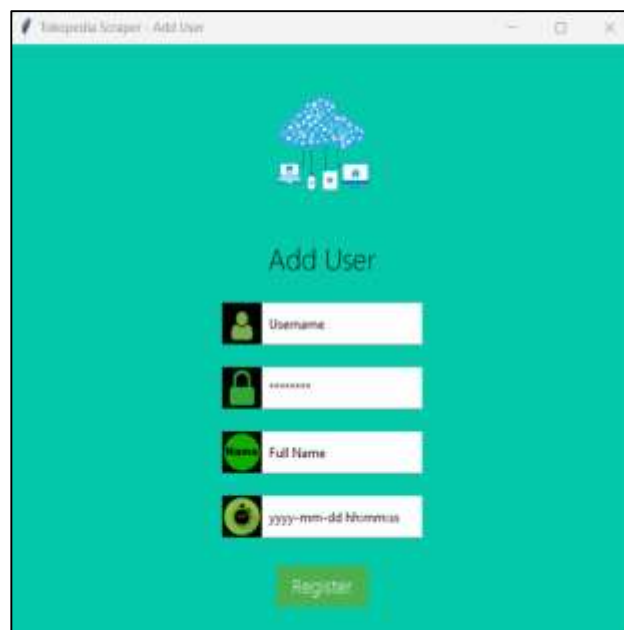
Gbr 6. Antarmuka Login Basic User

3.3. Manage User

Menu manage user terbagi menjadi dua section yaitu add dan manage data user. Jika admin memasukkan username yang sudah terdaftar didalam database tampilan yang muncul akan otomatis menjadi manage data, sebaliknya juga berlaku bagi username yang belum terdaftar dalam database yaitu akan menjadi add user yang berfungsi sebagai form registrasi user.



Gbr 7. Antarmuka Manage Data User



Gbr 8. Antarmuka Add User

3.4. Delete User

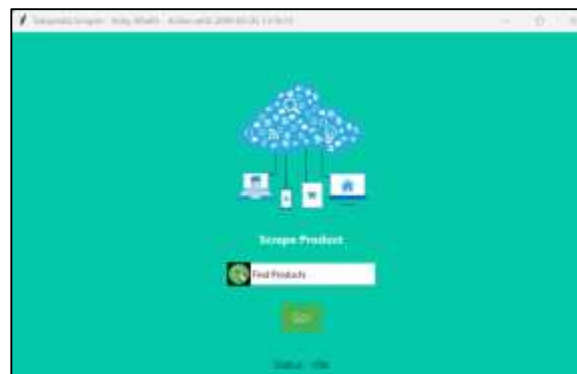
Pada menu delete user, admin dapat memilih user yang ingin dihapus datanya dengan cara double click list pada table yang tersedia dan nanti akan muncul konfirmasi deletenya. Hal ini ditujukan untuk menghindari miss click yang akan menyebabkan penghapusan data user yang tidak dikehendaki oleh admin.



Gbr 9. Antarmuka Delete User

3.5. Scraping Produk

Pada menu scraping produk, pengguna yang sudah login diharuskan mengisi kata kunci sebagai parameter pencarian pada Tokopedia, setelah kata kunci diinput script scraping otomatis akan berjalan di background dan status pada interface scraping produk pun berubah menjadi “Scraping..”. Jika proses scraping sudah selesai, akan muncul notifikasi pop-up yang berisikan bahwa scraping produknya sudah selesai.



Gbr 10. Antarmuka Scraping Produk

3.6. History

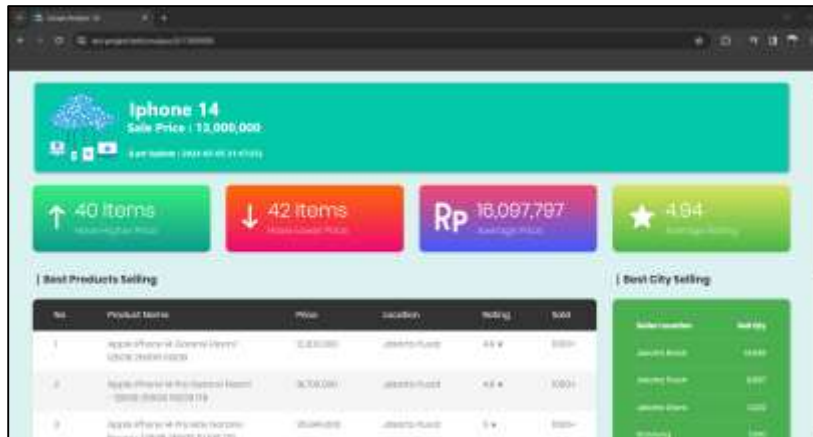
Menu history sendiri berisikan log scraping yang sudah dilakukan oleh pengguna yang bersangkutan.



Gbr 11. Antarmuka History Scraping Produk

3.7. Dashboard Analisa Scraping

Dashboard analisa scraping berisikan detail produk seller yang akan dikomparasi dengan produk-produk lain pada Tokopedia, pada header yang berisikan nama produk, harga jual, waktu scraping yang seller lakukan. Pada mini section berikutnya berisi card-card yang berisi summary tentang produk yang dicari seperti jumlah produk yang lebih mahal, produk yang lebih murah, rata-rata harga, dan rating pada kompetitor. Selanjutnya aplikasi juga menampilkan detail produk yang merupakan best seller yang diurutkan dari produk yang paling banyak terjual. List tersebut bisa di *double click* untuk membuka halaman pada Tokopedia. Lalu dashboard juga menyediakan data group berdasarkan kota untuk mengetahui kota yang memiliki minat terbesar.



Gbr 11. Antarmuka Analisa Scraping

4. PEMBAHASAN

Pengujian sistem adalah proses dimana sistem diuji coba kelayakan dan kualitas dari perangkat lunak yang dibuat. Ada dua jenis pengujian dalam tahap ini yaitu secara *white box* dan *black box*.

4.1. Pengujian Black Box

Black-Box testing (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.[14]

Tabel 1. Hasil Pengujian Black Box

Kelas Uji	Skenario Testing	Output yang Diharapkan	Kesimpulan
Login	Menjalankan aplikasi	Menampilkan Page Login	Valid
	Input username	Bisa diinput	Valid
	Input password	Bisa diinput dan hide input dengan symbol password	Valid
	Menekan tombol login	Validasi kredensial dan munculkan error jika kredensial tidak sesuai atau masa aktif akun sudah memasuki expired	Valid
Manage User	Mengakses menu manage user	Menampilkan Page Manage User	Valid
	Input username	Bisa diinput	Valid
	Logic Form Add / Manage data	Menampilkan menu add user jika username tidak ditemukan, menampilkan menu edit user data jika username ditemukan	Valid
	Isi form register	Form bisa diinput	Valid
	Save form register	Menambahkan record baru pada database	Valid
	Isi form manage data	Form bisa diinput	Valid
	Save form manage data	Mengedit record pada database sesuai dengan perubahan yang diterapkan pada setiap user	Valid

Delete User	Mengakses menu delete user	Menampilkan Page Delete User	Valid
	Menampilkan list user	Menampilkan user yang terdaftar di database	Valid
	Pilih user untuk di delete	List user yang ditampilkan bisa dipilih dengan cara double click list	Valid
	Menampilkan konfirmasi delete	Menampilkan konfirmasi setelah memilih user untuk dihapus	Valid
	Konfirmasi delete	Menghapus record pada database sesuai dengan user yang dipilih dan merefresh list lalu menampilkan kembali list user yang masih aktif	Valid
	Batalkan konfirmasi delete	Tidak ada record yang terhapus dari database	Valid
Scraping Produk	Mengakses menu Scrape Product	Menampilkan Page Scrape Product	Valid
	Input kata kunci pencarian	Bisa diinput	Valid
	Jalankan script scraping di background	Script scraping dijalankan di background	Valid
	Ubah status idle	Status menjadi “Scraping..” saat script berjalan di background	Valid
	Tampilkan notifikasi scrape selesai	Menampilkan notifikasi popup kalau proses scraping telah selesai	Valid
	Ubah status Scraping	Status menjadi “idle” saat script selesai	Valid
	Simpan data didatabase	Insert ke table history untuk log scraping dan insert ke table scrape_data untuk item yang sudah didapatkan dari proses scraping.	Valid
History	Mengakses menu History	Menampilkan Page History	Valid
	Menampilkan list pada table history	List log scraping tampil sesuai dengan user yang bersangkutan	Valid
Analisa Data Scraping	Membuka browser dengan redirect url dari list log scraping	Membuka browser dan membuka halaman web dashboard	Valid
	Tampilkan record pada header	Header terisi sesuai dengan parameter	Valid
	Tampilkan perhitungan pada card	Card terisi summary	Valid
	Tampilkan best product selling	Best product selling terisi, bisa redirect link asli	Valid
	Tampilkan best city selling	Best city selling terisi	Valid

4.2. Pengujian White Box

White box testing (pengujian kotak putih) yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. White box testing sebagai metode pengujian yang melibatkan pengetahuan detail tentang struktur internal sistem yang diuji, termasuk teknik seperti path testing, control flow testing, dan data flow testing.[15]

Tabel 2. Hasil Pengujian White Box

No	Testing	Test Case Valid
1	<pre> logo_label = tk.Label(frame, image=self.img, bg='#03c9a9') user_label = tk.Label(frame, image=self.userimg, bg='#000000') pass_label = tk.Label(frame, image=self.passimg, bg='#000000') login_label = tk.Label(frame, text="Member Login", fg="#000000", font=("Segoe UI Light", 20), bg='#03c9a9') style = ttk.Style() style.configure("TEntry", padding=5, relief="flat", borderwidth=5, font=("Segoe UI Light", 16)) self.username_entry = ttk.Entry(frame, style="TEntry") self.password_entry = ttk.Entry(frame, style="TEntry", show="*") self.username_entry.insert(0, "Username") self.password_entry.insert(0, "Password") self.username_entry.bind("<FocusIn>", self.on_username_focus_in) self.password_entry.bind("<FocusIn>", self.on_password_focus_in) login_button = tk.Button(frame, text="Login", fg="#FFFFFF", bg="#4CAF50", font=("Segoe UI Light", 13), command=self.login, bd=0, relief=tk.FLAT) register_label = tk.Label(frame, text="Register / Forgot Password?", fg="#0000FF", cursor="hand2", font=("Arial", 10, "underline"), bg='#03c9a9') register_label.bind("<Button-1>", lambda e: self.open_browser("https://wa.me/+620895332670775")) frame.columnconfigure(1, weight=1) logo_label.grid(row=0, column=0, columnspan=2, pady=20) login_label.grid(row=1, column=0, columnspan=2, sticky="news", pady=10) user_label.grid(row=2, column=0, pady=0, ipadx=0, ipady=0, sticky="ew") self.username_entry.grid(row=2, column=1, pady=10, ipadx=10, ipady=5, sticky="ew") pass_label.grid(row=3, column=0, pady=0, ipadx=0, ipady=0, sticky="ew") self.password_entry.grid(row=3, column=1, pady=10, ipadx=10, ipady=5, sticky="ew") login_button.grid(row=4, column=0, columnspan=2, pady=15, ipadx=10, ipady=1) register_label.grid(row=5, column=0, columnspan=2, pady=5) frame.pack(side='top', expand=True) </pre>	Menampilkan menu login, bisa mengisi username dan password yang characternya sudah di replace menjadi * agar privasi lebih terjaga, button login yang berisikan action validasi kredensial user, link untuk menghubungi admin via whatsapp
2	<pre> frame = tk.Frame(self.master, bg='#03c9a9', padx=20, pady=20) logo_label = tk.Label(frame, image=self.img, bg='#03c9a9') welcome_label = tk.Label(frame, text="Welcome, " + self.full_name_data, fg="#000000", font=("Segoe UI Light", 19), bg='#03c9a9') divider = tk.Label(frame, text="-----", fg="#03c9a9", font=("Segoe UI Light", 19), bg='#03c9a9') scrape_button = tk.Button(frame, text="Scrape Data", fg="#FFFFFF", bg="#4CAF50", font=("Segoe UI Light", 13), command=self.scrape_action, bd=0, relief=tk.FLAT) history_button = tk.Button(frame, text="Activity History", fg="#FFFFFF", bg="#4CAF50", font=("Segoe UI Light", 13), command=self.history_action, bd=0, relief=tk.FLAT) manage_button = tk.Button(frame, text="Manage User", fg="#FFFFFF", bg="#4CAF50", font=("Segoe UI Light", 13), command=self.manage_user, bd=0, relief=tk.FLAT) delete_button = tk.Button(frame, text="Delete User", fg="#FFFFFF", bg="#4CAF50", font=("Segoe UI Light", 13), command=self.delete_user, bd=0, relief=tk.FLAT) frame.columnconfigure(0, weight=1) # Center the logo </pre>	Menampilkan menu home beserta button yang berisikan perintah untuk mengakses menu sesuai akses user seperti scrape, history, manage user, dan delete user

	<pre> frame.columnconfigure(1, weight=1) # Center the buttons logo_label.grid(row=0, column=0, columnspan=2, pady=0, sticky="n") welcome_label.grid(row=1, column=0, columnspan=2, sticky="news", pady=10) scrape_button.grid(row=2, column=0, pady=15, ipadx=10, ipady=1) history_button.grid(row=2, column=1, pady=15, ipadx=10, ipady=1) if self.username_data == 'vici': manage_button.grid(row=3, column=1, pady=15, ipadx=10, ipady=1) delete_button.grid(row=3, column=0, pady=15, ipadx=10, ipady=1) divider.grid(row=4, column=0, columnspan=2, sticky="news", pady=10) frame.pack(side='top', expand=True) </pre>	
3	<pre> def extract_data(driver): driver.implicitly_wait(20) driver.refresh() scrolling() data_items = wait(driver, 30).until(EC.presence_of_all_elements_located((By.XPATH, '//div[contains(@class, "css-jza1fo")]'))) if len(data_items) != 80: driver.implicitly_wait(10) data_items = wait(driver, 30).until(EC.presence_of_all_elements_located((By.XPATH, '//div[contains(@class, "css-jza1fo")]'))) for item in tqdm(data_items): elements = wait(item, 10).until(EC.presence_of_all_elements_located((By.XPATH, '//div[contains(@class, "css-llwpbs")]'))) for element in elements: name = element.find_element(By.XPATH, './div[contains(@class, "prd_link-product-name")]').text price = element.find_element(By.XPATH, './div[contains(@class, "prd_link-product-price")]').text location = element.find_element(By.XPATH, './span[contains(@class, "prd_link-shop-loc")]').text try: rating = element.find_element(By.XPATH, './span[contains(@class, "prd_rating-average-text")]').text except: rating = None try: sold = element.find_element(By.XPATH, './span[contains(@class, "prd_label-integrity")]').text except: sold = 0 details_link = element.find_element(By.XPATH, './div[contains(@class, "prd_container-card")]//div[contains(@class, "pcv3__container")]//div[starts-with(@class, "css-")]/a').get_property('href') data = { 'name': name, 'price': price, 'location': location, 'rating': rating, 'sold': sold, 'link': details_link, 'h_id': None } product_data.append(data) stop = 1 while stop <= pages: print(f"Scraping page {stop}") extract_data(driver) print(f"Number of items scraped: {len(product_data)}") if progress_callback: progress_callback() try: </pre>	Running script background untuk akses halaman pencarian tokopedia sesuai dengan kata kunci yang diinput user

	<pre> next_page = wait(driver, 60).until(EC.element_to_be_clickable((By.CSS_SELECTOR, '[aria-label="Laman berikutnya"]'))) except: next_page = wait(driver, 60).until(EC.element_to_be_clickable((By.CSS_SELECTOR, '[aria-label="Laman berikutnya"]'))) try: next_page.click() except: break stop += 1 </pre>	
4	<pre> def on_double_click(self, event): item = self.tree.selection()[0] values = self.tree.item(item, 'values') h_id = values[3] self.scrape_item = values[1] self.master.destroy() user_input = simpledialog.askstring("Go!", f"Enter your sale price for {self.scrape_item}:") if user_input is not None: if user_input.isdigit(): self.open_browser("https://vici-project.tech/analyze/" + h_id + "/" + user_input) else: messagebox.showerror(title="Error", message="Please enter a valid number!") else: root = tk.Tk() history_app = HistoryApp(root, self.username_data, self.full_name_data, self.exp_date, self.user_id) root.mainloop() </pre>	Membuka browser dan redirect ke website dashboard sesuai dengan parameter url yang ada

5. KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisa yang telah dibahas pada bab sebelumnya maka dapat diperoleh beberapa kesimpulan, yaitu:

- Aplikasi web scraping Tokopedia merupakan solusi efektif untuk mengumpulkan data dari platform e-commerce terkemuka seperti Tokopedia. Dengan menggunakan teknologi scraping, aplikasi ini dapat secara otomatis mengakses halaman-halaman Tokopedia dan mengekstraksi informasi produk, harga, ulasan, dan lainnya.
- web scraping yang berbasis python ini dapat digunakan sebagai media pengumpulan dan analisa data yang tetap mengutamakan etika dan syarat dan ketentuan yang berlaku pada platform yang ada.

5.2. Saran

Adapun saran yang penulis ajukan terhadap sistem aplikasi ini untuk pengembangan sistem ini selanjutnya adalah sebagai berikut :

- Sistem aplikasi web scraping ini merupakan versi awal dan masih harus banyak dikembangkan dari segi kompilasi program maupun hosting yang memerlukan peningkatan untuk performa pengelolaan data yang lebih cepat dan meminimalisir connection time out saat memproses data.
- Diperlukannya pembaruan secara berkala script scraping yang digunakan dalam aplikasi untuk mengakomodasi perubahan dalam tata letak atau struktur halaman Tokopedia

6. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada semua pihak yang telah mendukung dan berkontribusi dalam penyelesaian penelitian ini:

- Seluruh civitas Yayasan Sasmita Jaya Group.
- Seluruh tenaga pengajar Universitas Pamulang.
- Bapak Mochamad Adhari Adiguna S.ST., M.Kom. selaku Dosen Pembimbing.
- Orang tua dan keluarga yang telah mendukung dan memotivasi dalam kehidupan saya.
- Semua pihak yang telah memberikan dukungan dalam menyelesaikan penelitian ini.

7. DAFTAR PUSTAKA

Journal Article

- [1] Wikipedia contributors. (2024, May 8). *Waterfall model*. Wikipedia, The Free Encyclopedia.
- [2] Akbar Nur Syahrudin, & Tedi Kurniawan. (2018). Input Dan Output Pada Bahasa Pemrograman Python. *Jurnal Dasar Pemrograman Python STMIK*, 1–7.
- [3] Wes McKinney. (2017). *Python for Data Analysis* (2nd ed). O'Reilly Media.
- [4] Wikipedia contributors. (2024, Feb 27). *Pycharm*. Wikipedia, The Free Encyclopedia.
- [5] Rohi Abdulloh. (2018). *7 in 1 Pemrograman Web untuk Pemula*. PT. Elex Media Komputindo.
- [6] Yuhefizar. (2016). *10 Pengertian Website Menurut Para Ahli*.
- [7] Terry Felke-Morris. (2021). *Web Development and Design Foundations with HTML5* (9th ed).
- [8] Jean-Claude Wippler. (2017). *The Essential Guide to HTML5: Using Games to Learn HTML5 and JavaScript*.
- [9] Wikipedia contributors. (2024, Feb 1). *Tokopedia*. Wikipedia, The Free Encyclopedia.
- [10] Wikipedia contributors. (2023, Des 1). *MySQL*. Wikipedia, The Free Encyclopedia.
- [11] U. L. D. Putri, Sintia and A. F. Prasetya, "Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language)," *Jurnal Ilmiah Komputer Terapan dan Informasi*, pp. 14-18, 2022.
- [12] D. W. T. Putra and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohon Pembayaran Restitusi SPPD," *Jurnal TEKNOIF*, pp. 32-39, 2019.
- [13] Alan Cooper, Robert Reimann, & David Cronin. (2014). *About Face: The Essentials of Interaction Design* (4th ed).
- [14] Rosa AS, & Muhammad Salahuddin. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- [15] William E. Lewis. (2015). *Software Testing and Continuous Quality Improvement*.