BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Program kerja magang sebagai *data engineer* di PT Keindahan Sejahtera Hutama memiliki tugas melakukan pengumpulan dan pengelolaan data perusahaan yang bersumber dari eksternal. Kedudukan *data engineer* memiliki tugas dan wewenang di bawah kedua manager divisi dan *AI developer* (Reynard Mathhew Yaputra). *Data engineer* berkolaborasi dengan *AI developer* dalam membuat dan menjalankan program pengumpulan data yang akan digunakan oleh sistem pendukung keputusan di dalam server aplikasi UNNIS. Tugas lain yang dikerjakan oleh *data engineer* ialah melakukan pembersihan data-data produk kecantikan yang telah dimiliki perusahaaan, serta memberikan ide dan saran terhadap arsitektur data tersebut sehingga dapat tercapai dan terukur.

Kegiatan koordinasi pengerjaan tugas dilakukan melalui rapat seluruh tim IT maupun dengan seluruh karyawan KSH lain bila perlu dan memungkinkan (*on-demand*). Rapat dilakukan secara bertemu tatap muka di kantor dan juga secara *online* sesuai materi yang perlu didiskusikan dan pihak-pihak yang terlibat. Seluruh karyawan juga dapat berkoordinasi berkelompok melalui platform komunikasi bisnis Slack dan Whatsapp. Agenda perusahaan dan catatan hasil rapat yang sedang dan akan dilakukan perusahaan disampaikan melalui aplikasi Notion. Pengiriman surat elektronik internal perusahaan, sekaligus sebagai sistem ERP, KSH menggunakan ECount Webmail. Karyawan dan *intern* memiliki hak untuk mengajukan pertanyaan dan berkomunikasi dengan supervisor, serta dengan anggota tim lainnya maupun karyawan dari departemen Brand Strategist terkait kebutuhan atau masalah yang muncul selama kegiatan kerja magang. Contoh kebutuhan tersebut dapat berupa permintaan izin tidak hadir untuk urusan akademik, informasi mengenai sistem data kepada *Back-end developer*, atau permintaan informasi lain mengenai tugas yang dibutuhkan untuk pembuatan konten promosi perusahaan kepada departemen Brand Strategist.

3.2 Tugas dan Uraian Kerja Magang

Tugas utama yang diberikan KSH kepada *data engineer intern* ialah pekerjaan yang terkait dengan pengumpulan data produk untuk sistem rekomendasi produk aplikasi UNNIS dan melakukan pekerjaan lain yang berkaitan dengan pengelolaan data yang dimiliki KSH seperti *data cleansing* maupun *standardization*. Tabel 3.1 memaparkan pekerjaan-pekerjaan yang dilakukan oleh *Data Engineer* selama enam bulan masa kontrak kerja magang ini.

Tabel 3.1 Waktu Kegiatan Kerja Magang Hingga Mei 2024

Pekerjaan	**Periode Pelaksanaan**
Proyek Utama	
Pengumpulan Gambar Produk Olive Young.	5 Februari – 23 Februari 2024
Pengumpulan Gambar Produk Sephora ID.	26 Februari – 8 Maret 2024
Pengumpulan Gambar Produk Sephora USA.	11 Maret – 26 April 2024
Tugas Sampingan	
Otomatisasi data entri dan validasi data nomor BPOM.	5 Februari – 30 Agustus 2024
Scraping konten video Youtube dan TikTok yang mereview produk kosmetik dan skincare Korea Selatan berbahasa Indonesia.	1 Mei – 17 Mei 2024
Pemindahan data review produk dari konsumen Unnis Pick.	20 Mei – 31 Juli 2024
Pengumpulan data kulit wajah Indonesia dari mall.	1 Juli 2024 – 31 Juli 2024
Pengelolaan dan pengecekan kualitas database aplikasi UNNIS.	Hingga 31 Juli 2024

Pada pekerjaan magang ini dapat dibuat tabel *timeline* magang *Data Engineer* di KSH seperti pada Tabel 3.1 Waktu Kegiatan Kerja Magang Hingga Mei 2024. Proyek utama yang akan dikerjakan mencakup pengumpulan gambar produk dari berbagai platform e-commerce ternama. Tugas ini meliputi

pengumpulan gambar produk dari Olive Young, Sephora ID, dan Sephora USA. Proses ini memerlukan ketelitian dan keterampilan dalam mengelola data visual agar dapat menyediakan konten berkualitas tinggi dan sesuai standar yang dibutuhkan oleh perusahaan. Pengumpulan gambar produk ini bertujuan untuk memperkaya database perusahaan dengan informasi visual yang akurat dan up-to-date, yang nantinya akan digunakan dalam berbagai keperluan pemasaran dan penjualan.

Selain proyek utama, terdapat beberapa tugas sampingan yang tak kalah penting untuk mendukung operasional perusahaan. Salah satunya adalah otomatisasi data entri dan validasi data nomor BPOM, yang bertujuan untuk meningkatkan efisiensi dan akurasi dalam pengelolaan data produk *skincare* kosmetik. Selain itu, pekerjaan ini juga melibatkan scraping konten video dari YouTube dan TikTok yang berisi review produk kosmetik dan *skincare* Korea Selatan dalam bahasa Indonesia. Tugas lainnya termasuk pemindahan data *review* produk dari konsumen Unnis Pick agar memberikan informasi nyata mengenai kesan produk yang dijual setelah digunakan. Kemudian pengumpulan data kulit wajah Indonesia dari mall dilakukan sebagai kerjasama KSH bersama Institut Teknologi Korea. Pekerjaan sampingan lain berupa pengelolaan dan pengecekan kualitas database aplikasi UNNIS juga dilakukan secara konsisten. Semua tugas ini dirancang untuk memastikan bahwa data yang digunakan oleh perusahaan selalu berkualitas tinggi dan relevan dengan kebutuhan pasar.

Sebelum memulai proyek-proyek tersebut, terlebih dahulu diberikan metadata berupa info produk-produk kecantikan yang diperdagangkan oleh Olive Young dan Sephora beserta tautan dari setiap produk yang tercatat seperti contoh daftar informasi produk yang tertera di dalam Tabel 3.2. Data yang telah dikumpulkan perusahaan didapatkan sebelum tanggal 1 Februari 2024 atau sebelum periode kerja magang ini.

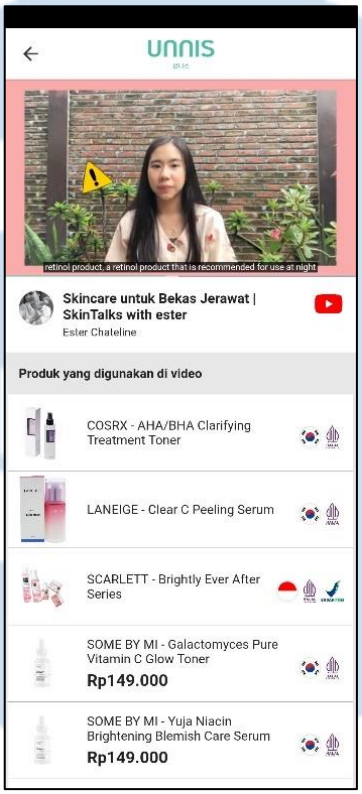
Tabel 3.2 Contoh Data Produk Sephora.co.id yang Diberikan Perusahaan

brand_name	product_name	rating	claim	ingredients	instruction	desc	variant	price

aig**r	Cara Mia Eau De Perfume Spray	0			Spray the perfume ***	Ini adalah: ***	1**ml	Rp 1.***.000
al***s-of-s**n	CE** B*k***iol F**ming Oil	*	Product Claims: ***	"Minyak Biji R**a ***	"GUNAKAN SETIAP ***	Ini adalah: ***		Rp 2.***.000
Anas***-bev**ly-hi***	Full & Feathered Kit (Limited Edition)	5	***	***	***	***	***	***
Anas***-bev**ly-hi***	Full & Feathered Kit (Limited Edition)	5	***	***	***	***	***	***
Anas***-bev**ly-hi***	Full & Feathered Kit (Limited Edition)	5	***	***	***	***	***	***
Anas***-bev**ly-hi***	Full & Feathered Kit (Limited Edition)	5	***	***	***	***	***	***
Anas***-bev**ly-hi***	Full & Feathered Kit (Limited Edition)	5	***	***	***	***	***	***
Anas***-bev**ly-hi***	Cosmos Eye Shadow Palette	4.1	***	***	***	***	***	***
Anas***-bev**ly-hi***	Brow Care Kit (Limited Edition)	0	***	***	***	***	***	***
Anas***-bev**ly-hi***	Brow Care Kit (Limited Edition)	0	***	***	***	***	***	***
Anas***-bev**ly-hi***	Brow Care Kit (Limited Edition)	0	***	***	***	***	***	***
Anas***-bev**ly-hi***	Brow Care Kit (Limited Edition)	0	***	***	***	***	***	***
Anas***-bev**ly-hi***	Tinted Lip Gloss	4.4	***	***	***	***	***	***
Anas***-bev**ly-hi***	Tinted Lip Gloss	4.4	***	***	***	***	***	***
Anas***-bev**ly-hi***	Tinted Lip Gloss	4.4	***	***	***	***	***	***

Data utama yang perlu dikumpulkan *data engineer* ialah gambar-gambar produk dari dataset tautan produk melalui website resmi mereka. Hal ini dilakukan karena KSH tidak memiliki akses ke terhadap API website eksternal yang menjadi sumber rekomendasi produk. Sehingga diperlukan metode lain untuk mengumpulkan gambar produk-produk eksternal. Pengumpulan data penunjang

yang juga menjadi tugas *data engineer* ialah nomor BPOM dari produk-produk yang telah terkumpul.



Gambar 3.1 Tampilan Halaman Menu Rekomendasi Video di Aplikasi UNNIS

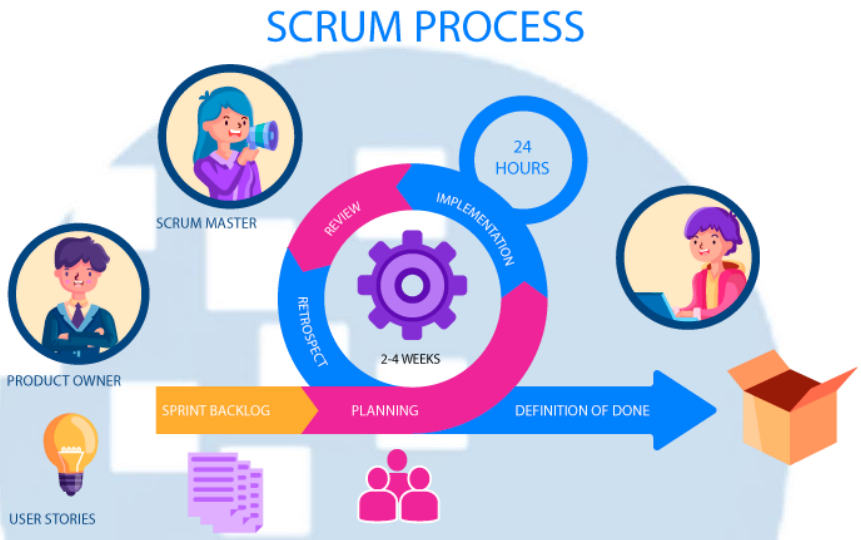
Gambar 3.1 menampilkan bagaimana gambar produk-produk rekomendasi dicantumkan di bawah sebuah video seseorang yang sedang melakukan *review skincare*. Di samping nama dan merek produk, juga dicantumkan bendera negara asal produksi *skincare*, beserta logo BPOM maupun Halal Indonesia bila produk sudah terdaftar di situs resmi Badan POM dan Badan Penyelenggara Jaminan Produk Halal Kementerian Agama RI.

3.2.1 Metode dan alur kerja

Seluruh prosedur penugasan proyek yang di lakukan di divisi IT PT Keindahan Sejahtera Hutama dapat dikategorikan menjadi salah satu metode manajemen proyek perusahaan *agile*, yaitu *Scrum*. Perusahaan memberikan fleksibilitas bagi karyawan membagi proyek ke dalam siklus-skilus pendek (*Sprint*). Karyawan divisi IT KSH diminta untuk responsif terhadap adanya

perubahan prioritas kegiatan bisnis yang mendadak. Misalnya rencana penambahan fitur Chat di dalam aplikasi UNNIS perlu ditunda akibat adanya penugasan prioritas membuat *Research & Development Notes* oleh jajaran eksekutif. Siklus pendek yang dimaksud ialah KSH memberlakukan satu periode pemilihan tugas dan penyelesaiannya selama dua minggu bagi setiap posisi karyawan. Selama satu siklus tugas, satu hari dialokasikan untuk merancang *sprint planning*, delapan hari untuk menyelesaikan tugas *sprint*, dan satu hari untuk meninjau capaian target *sprint* serta refleksi hasil yang didapat.

Divisi IT KSH menerapkan metodologi *Scrum* [11] yang cukup serupa dengan Gambar 3.2. CEO Mrs. Yuna menjadi *Product Owner* yang mana setiap proyek aplikasi UNNIS akan dikerjaskan demi keberhasilan visi perusahan dan target ROI. IT manager berperan sebagai *Scrum Master*, bertanggung jawab atas perkembangan pengerjaan *Scrum*, memberikan pelatihan, dan *mentoring* karyawan saat diperlukan. *Sprint* dirumuskan bersama dan ditaruh ke dalam sebuah papan *notes*. Setiap karyawan berkewajiban menyampaikan hasil imlementasi *Sprint* mereka selama satu hari kerja kepada *Scrum Master*. Hasil yang tercapai di akhir periode Scrum akan dievaluasi dan hal-hal yang menyebabkan keterlambatan atau perubahan target ditinjau agar IT manager dapat memberikan pertimbangan maupun hal-hal yang dapat ditingkatkan bagi seluruh tugas sebelum dinyatakan selesai.



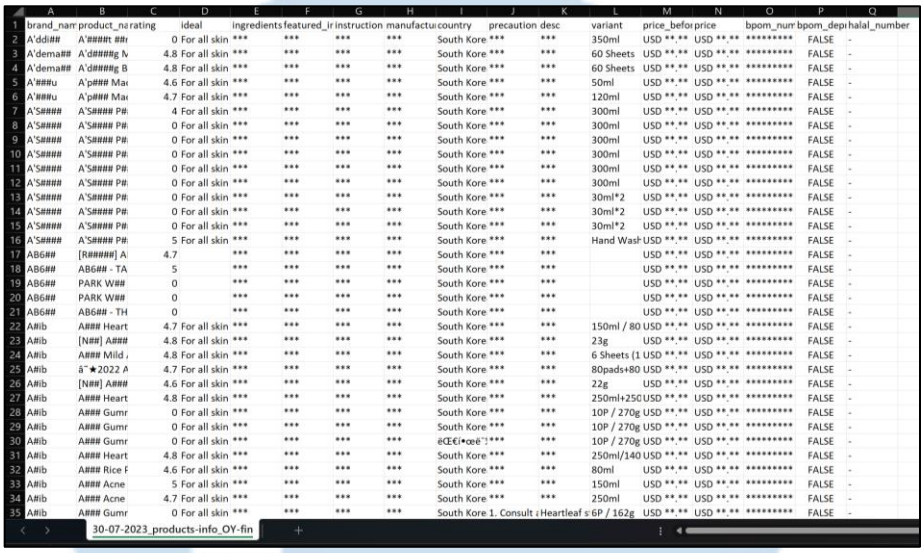
Gambar 3.2 Diagram Proses Scrum pada Agile Methodology

Sumber: [13]

3.2.2 Proyek 1: Pengumpulan Gambar Produk Olive Young

Minggu pertama kerja magang ini dimulai dengan pengarahan tugas kewajiban *data engineer*, adaptasi terhadap lingkungan serta budaya kantor KSH, serta riset mengenai otomasi *website data scraping* menggunakan Python. Didapatkan bahwa Python memiliki pustaka bernama Selenium yang banyak digunakan untuk pembuatan program menjalankan perintah kepada website html secara automatis dan spesifik. Selenium memiliki modul spesial bernama WebDriver yang berguna melakukan pengujian otomatis pada aplikasi website di beragam mesin jelajah internet. Mesin jelajah internet Chrome juga memiliki server mandiri web drivernya sendiri untuk mendukung developer bekreasi dengan kemampuan khusus Chromium. Di sini ChromeDriver digunakan untuk Windows dengan versi 114.0.5735.90 karena mampu mendukung versi Chrome 114 yang digunakan perangkat. Proyek pertama dikerjakan mulai dari 5 Februari sampai dengan 23 Februari 2024 dengan diberikannya dataset Informasi Produk Olive Young dan daftar tautan produk yang telah dimiliki AI developer, tampak pada Gambar 3.3 dan Gambar 3.4. Sebanyak 12.015 baris ragam informasi variasi produk Olive Young dari 7.878

tautan produk yang berbeda. Sebuah tautan produk dapat berisi lebih dari dua variasi yang berbeda, ditunjukkan pada kolom bernama Variant.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	brand_nam	product_na	rating	ideal	ingredients	featured_ir	instruction	manufactu	country	precaution	desc	variant	price_befo	price	bpom_num	bpom_dep	halal_number
2	A'ddi##	A'#### ##	0	For all skin	***	***	***	***	South Kore	***	***	350ml	USD **.**	USD **.**	**********	FALSE	-
3	A'dema##	A'd#### N	4.8	For all skin	***	***	***	***	South Kore	***	***	60 Sheets	USD **.**	USD **.**	**********	FALSE	-
4	A'dema##	A'd#### B	4.8	For all skin	***	***	***	***	South Kore	***	***	60 Sheets	USD **.**	USD **.**	**********	FALSE	-
5	A'###u	A'p### Ma	4.6	For all skin	***	***	***	***	South Kore	***	***	50ml	USD **.**	USD **.**	**********	FALSE	-
6	A'###u	A'p### Ma	4.7	For all skin	***	***	***	***	South Kore	***	***	120ml	USD **.**	USD **.**	**********	FALSE	-
7	A'S####	A'S#### P#	4	For all skin	***	***	***	***	South Kore	***	***	300ml	USD **.**	USD **.**	**********	FALSE	-
8	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	300ml	USD **.**	USD **.**	**********	FALSE	-
9	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	300ml	USD **.**	USD **.**	**********	FALSE	-
10	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	300ml	USD **.**	USD **.**	**********	FALSE	-
11	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	300ml	USD **.**	USD **.**	**********	FALSE	-
12	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	300ml	USD **.**	USD **.**	**********	FALSE	-
13	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	30ml*2	USD **.**	USD **.**	**********	FALSE	-
14	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	30ml*2	USD **.**	USD **.**	**********	FALSE	-
15	A'S####	A'S#### P#	0	For all skin	***	***	***	***	South Kore	***	***	30ml*2	USD **.**	USD **.**	**********	FALSE	-
16	A'S####	A'S#### P#	5	For all skin	***	***	***	***	South Kore	***	***	Hand Wash	USD **.**	USD **.**	**********	FALSE	-
17	AB6##	[R######] A	4.7		***	***	***	***	South Kore	***	***		USD **.**	USD **.**	**********	FALSE	-
18	AB6##	AB6## - TA	5		***	***	***	***	South Kore	***	***		USD **.**	USD **.**	**********	FALSE	-
19	AB6##	PARK W##	0		***	***	***	***	South Kore	***	***		USD **.**	USD **.**	**********	FALSE	-
20	AB6##	PARK W##	0		***	***	***	***	South Kore	***	***		USD **.**	USD **.**	**********	FALSE	-
21	AB6##	AB6## - TH	0		***	***	***	***	South Kore	***	***		USD **.**	USD **.**	**********	FALSE	-
22	A#ib	A### Heart	4.7	For all skin	***	***	***	***	South Kore	***	***	150ml / 80	USD **.**	USD **.**	**********	FALSE	-
23	A#ib	[N##] A###	4.8	For all skin	***	***	***	***	South Kore	***	***	23g	USD **.**	USD **.**	**********	FALSE	-
24	A#ib	A### Mild	4.8	For all skin	***	***	***	***	South Kore	***	***	6 Sheets (1	USD **.**	USD **.**	**********	FALSE	-
25	A#ib	á˜★2022 A	4.7	For all skin	***	***	***	***	South Kore	***	***	80pads+80	USD **.**	USD **.**	**********	FALSE	-
26	A#ib	[N##] A###	4.6	For all skin	***	***	***	***	South Kore	***	***	22g	USD **.**	USD **.**	**********	FALSE	-
27	A#ib	A### Heart	4.8	For all skin	***	***	***	***	South Kore	***	***	250ml+250	USD **.**	USD **.**	**********	FALSE	-
28	A#ib	A### Gumr	0	For all skin	***	***	***	***	South Kore	***	***	10P / 270g	USD **.**	USD **.**	**********	FALSE	-
29	A#ib	A### Gumr	0	For all skin	***	***	***	***	South Kore	***	***	10P / 270g	USD **.**	USD **.**	**********	FALSE	-
30	A#ib	A### Gumr	0	For all skin	***	***	***	***	ëŒ€í•œë¯¼	***	***	10P / 270g	USD **.**	USD **.**	**********	FALSE	-
31	A#ib	A### Heart	4.8	For all skin	***	***	***	***	South Kore	***	***	250ml/140	USD **.**	USD **.**	**********	FALSE	-
32	A#ib	A### Rice F	4.6	For all skin	***	***	***	***	South Kore	***	***	80ml	USD **.**	USD **.**	**********	FALSE	-
33	A#ib	A### Acne	5	For all skin	***	***	***	***	South Kore	***	***	150ml	USD **.**	USD **.**	**********	FALSE	-
34	A#ib	A### Acne	4.7	For all skin	***	***	***	***	South Kore	***	***	250ml	USD **.**	USD **.**	**********	FALSE	-
35	A#ib	A### Gumr	0	For all skin	***	***	***	***	South Kore	1. Consult	Heartleaf s	6P / 162g	USD **.**	USD **.**	**********	FALSE	-

30-07-2023_products-info_OY-fin

Gambar 3.3 Dataset Informasi Produk Olive Young

Keterangan variabel-variabel data ialah sebagai berikut:

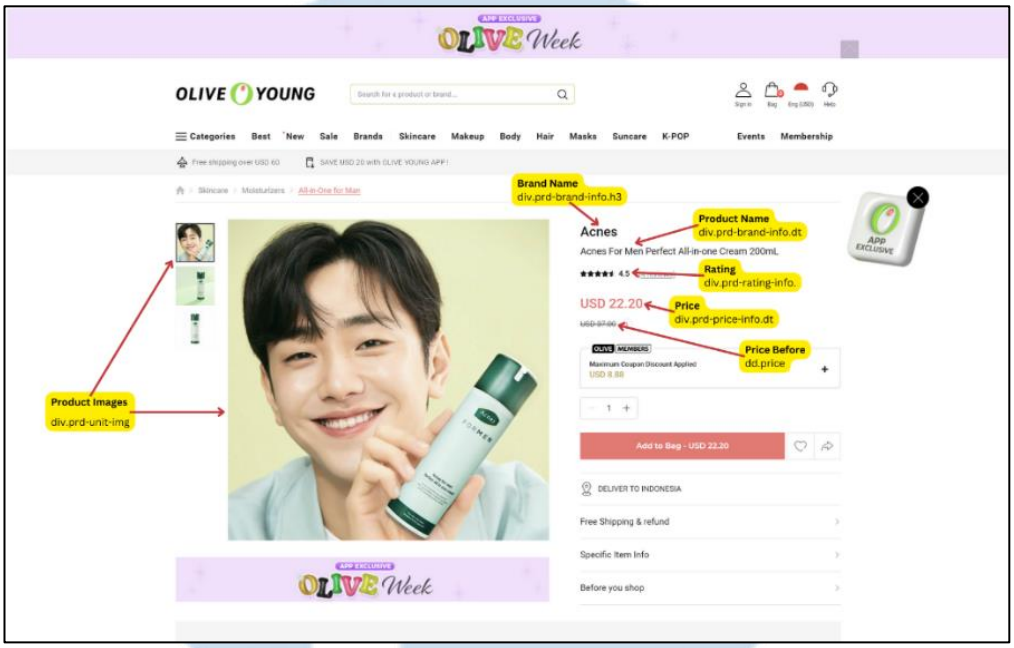
brand_name : Nama merek produk.
product_name : Nama produk.
rating : Skor kepuasan pembeli.
ideal : Spesifikasi peruntukkan jenis kulit/rambut/tubuh.
ingredients : Bahan pembuatan.
Instruction : Tata cara penggunaan.
manufacturer : Nama pabrik produsen.
country : Negara asal pabrik produsen.
precautions : Peringatan sebelum menggunakan produk.
desc : Deskripsi lengkap dari produk.
Variant : Variasi dari produk.
Price_before : Harga produk sebelum *discount*.
Price : Harga jual produk.

	A	B	C	D	E	F	G
1	brand	product_links					
2	A'ddict	https://global.oliveyoung.com/product/detail?prdtNo=GA220715456					
3	A'demang	https://global.oliveyoung.com/product/detail?prdtNo=GA220916197					
4	A'demang	https://global.oliveyoung.com/product/detail?prdtNo=GA220916196					
5	A'pieu	https://global.oliveyoung.com/product/detail?prdtNo=GA220715427					
6	A'pieu	https://global.oliveyoung.com/product/detail?prdtNo=GA220715428					
7	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815724					
8	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815725					
9	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815726					
10	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815727					
11	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815728					
12	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815729					
13	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815730					
14	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815731					
15	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220815732					
16	A'SCENT	https://global.oliveyoung.com/product/detail?prdtNo=GA220916161					
17	AB6IX	https://global.oliveyoung.com/product/detail?prdtNo=GA220715626					
18	AB6IX	https://global.oliveyoung.com/product/detail?prdtNo=GA221016603					
19	AB6IX	https://global.oliveyoung.com/product/detail?prdtNo=GA230217775					
20	AB6IX	https://global.oliveyoung.com/product/detail?prdtNo=GA230217776					
21	AB6IX	https://global.oliveyoung.com/product/detail?prdtNo=GA230518645					
22	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA220815976					
23	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719483					
24	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA220715420					
25	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA220715423					
26	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230418140					
27	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719552					
28	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719580					
29	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719582					
30	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719584					
31	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA220313955					
32	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA210510558					
33	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230618917					
34	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA210004558					
35	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719579					
36	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA221217234					
37	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA221217235					
38	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230217725					
39	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA210001378					
40	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230518780					
41	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA210004559					
42	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA220715422					
43	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA210001054					
44	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719590					
45	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719581					
46	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719583					
47	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA221116806					
48	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA220815936					
49	Abib	https://global.oliveyoung.com/product/detail?prdtNo=GA230719587					

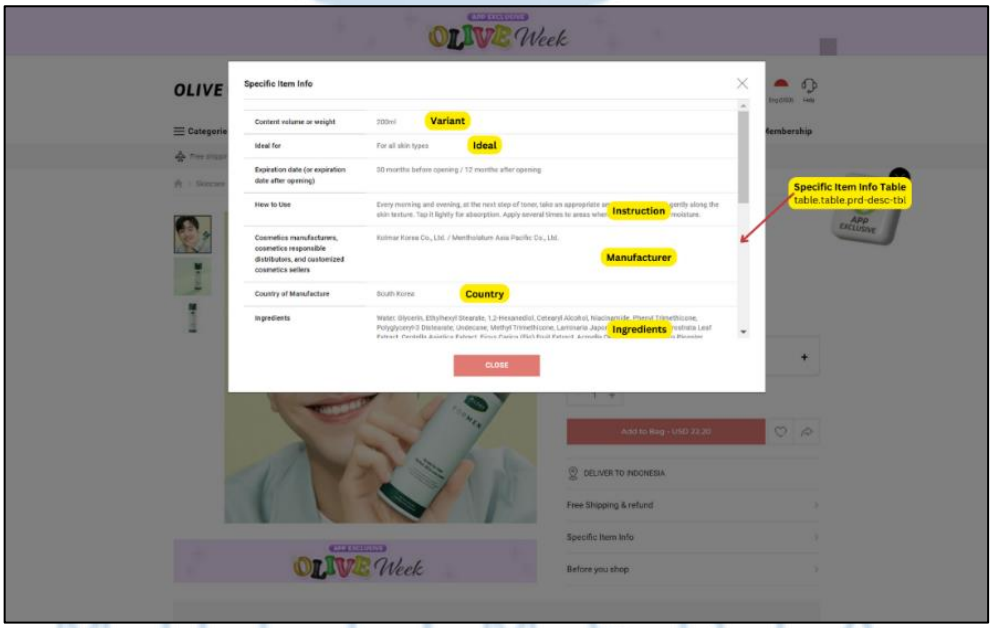
Gambar 3.4 Dataset Tautan Produk Olive Young

Tahap pertama yang dilakukan sebelum menuliskan program ialah menganalisis struktur dari website Olive Young dengan cara memeriksa *Elements* HTML dari website tersebut. *Elements* yang diperiksa meliputi nama-nama *div class* yang mencantumkan informasi produk dan posisi-posisi gambar ditampilkan menggunakan tautan yang konsisten. Berikut struktur dari website

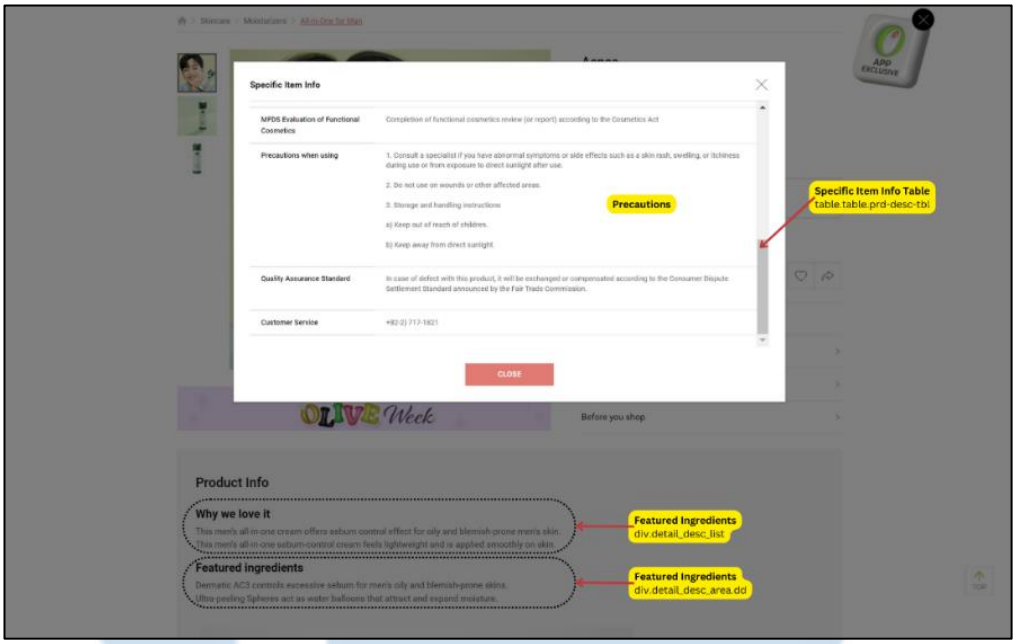
Olive Young berupa nama element dari informasi produk yang ada. Seluruh informasi ini digunakan saat melakukan *scraping* metadata dari detail produk.



Gambar 3.5 Tampilan Halaman Produk di Website Olive Young



Gambar 3.6 Tampilan Menu Specific Item Info (1) di Website Olive Young

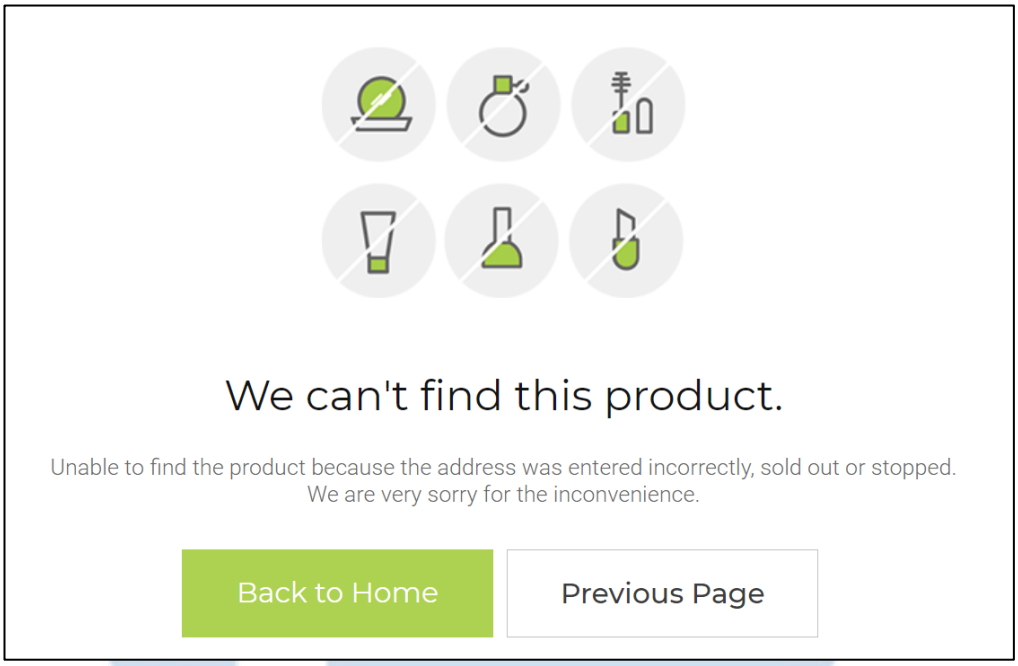


Gambar 3.7 Tampilan Menu Specific Item Info (2) di Website Olive Young

Dari hasil analisa struktur website Olive Young pada Gambar 3.5, Gambar 3.6, & Gambar 3.7, gambar-gambar produk berada di dalam element div.prd-unit-img dan merupakan tautan seperti dibawah ini.

https://image.globaloliveyoungshop.com/prdtImg/***/****.jpg?RS=1500x1500&AR=0

Setiap gambar produk disimpan ke sub-folder prdtImg dan ditampilkan menjadi ukuran 1500x1500 dimensi pixel. Sehingga https://image.globaloliveyoungshop.com/prdtImg/ dipilih sebagai target dari CSS *selector* program *scraping* dan dijadikan variabel bernama image_elements. Artinya *library* Selenium akan mencari seluruh keberadaan image_elements untuk diperintahkan lebih lanjut. Produk-produk penawaran Olive Young (iklan) memiliki struktur yang sama, namun ukuran yang berbedan. Sehingga kode untuk menyaring gambar yang diunduh hanya berukuran 1500x1500 dilakukan. Penemuan lain ialah ketika sebuah tautan produk dibuka, maka ada kemungkinan produk sudah dihapus atau diubah oleh Olive Young dan halaman *website* akan menampilkan kalimat “We can't find this product.” yang terlihat pada Gambar 3.8.



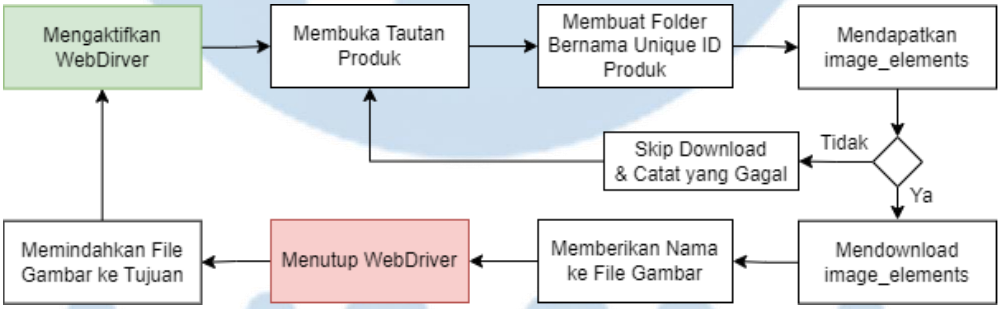
Gambar 3.8 Tampilan Ketika Produk Tidak Ditemukan

Tahap kedua proyek ini adalah merancang bagaimana program ini bekerja dan metode penyimpanan gambar ke dalam *storage device* perusahaan. Syarat utama program ialah harus memenuhi beberapa alasan yang berhubungan dengan saat program dijalankan seperti pada Tabel 3.3.

Tabel 3.3 Persyaratan Program Scraping Gambar

Syarat Program	**Alasan**
Dapat berulang secara otomatis.	Tautan produk sangatlah banyak apabila dikerjakan secara manual.
Lokasi penyimpanan file harus mampu menunjukkan kepemilikan gambar.	Gambar akan dipanggil bersama dengan informasi yang dimiliki.
Mampu memperlihatkan proses yang sedang berjalan.	Kemajuan proses *scraping* gambar dapat terlacak.
Dapat berhentikan untuk sementara.	Program hanya dijalankan saat operator bekerja di kantor.

Berdasarkan persyaratan-persyaratan program yang telah dirumuskan, alur kerja dari program diputuskan menjadi seperti tampak pada Gambar 3.9. Cara kerja dari program *scraping* ini ialah webdriver dari selenium akan melakukan iterasi membuka data tautan produk Olive Young dan mengunduh gambar dengan spesifikasi yang diinginkan. Penamaan folder penyimpanan setiap produk diambil dari tautan produk itu sendiri. Tautan https://global.oliveyoung.com/product/detail?prdtNo=GA2205***** akan menjadi contoh, maka tautan disimpan ke dalam folder bernama prdtNo_GA2205*****. File-file gambar ditentukan agar bernama image_1.jpg, image_2.jpg, hingga seterusnya. Adapun setiap tautan produk yang gagal diunduh akan dicatat ke dalam sebuah dokumen TXT kemudian hanya meninggalkan folder kosong.



Gambar 3.9 Diagram Alur Siklus Program Image Scraping Olive Young

Dengan modal element HTML yang menjadi target telah diketahui, maka penulisan program dapat dimulai menggunakan aplikasi Jupyter Notebook. Jupyter Notebook dipilih karena memiliki UI/UX yang ramah bagi developer dengan *multiple* bahasa pemrograman termasuk Python. Pustaka-pustaka dan modul Python dipanggil dengan susunan sebagai berikut.

1. PIL Image : Membuka dan menyimpan gambar.
2. Os : Manajemen direktori dokumen dan manipulasi Path.
3. Time : Pengaturan waktu operasi program.
4. Selenium : Otomasi perintah interaksi aplikasi website.
5. WebDriver : Mengontrol browser website.

6. Request : Mengunduh gambar dari website.
7. Urllib.parse : Memecah URL dan *parsing query string into dictionary*.

Terdapat dua jenis gambar produk yang dimuat ke dalam *script website*. Pertama ialah *thumbnail* berukuran 70x70 pixel, dan 1500x1500 pixel ketika pengguna menekan *thumbnail* gambarnya. Maka perlu adanya pendefinisian filter agar program yang dibangun mengambil gambar dengan dimensi terbesar (1500x1500 pixel). Kode untuk melakukan penyaringan gambar menjadi yang hanya ingin diunduh disajikan pada Gambar 3.10.

```
def download_image(image_url, save_path):
    response = requests.get(image_url)
    image = Image.open(BytesIO(response.content))
    width, height = image.size
    if width == 1500 and height == 1500:
        with open(save_path, 'wb') as img_file:
            img_file.write(response.content)
        print(f'Downloaded: {save_path}')
    else:
        print(f'Skipped: {image_url} - Image size not 1500x1500 pixels')
```

Gambar 3.10 Potongan Kode untuk Pendefinisan Filter Gambar

Setiap tautan produk disimpan ke dalam subfolder berdasarkan id "pdrtNo" mereka mengunakan function dari library urlsparse dan parse_qs seperti pada Gambar 3.11. Teks URL dipisahkan secara *default*, yakni setiap munculnya karakter *non-string* lalu beberapa bagian tautan diambil menjadi parameter. Fungsi parse_qs digunakan untuk *parsing query string* dari URL menjadi sebuah *dictionary* yang berisi pasangan kunci nilai dari parameter-parameter *query*. Jika mengambil contoh sebelumnya, pdrtNo akan menjadi parameter dan daftar *string* yang telah diparsing diambil nilai pertamanya. maka kode akan mengembalikan *dictionary* yang berisi pdrt_no = 'GA*********'.

```
def get_prdt_number(url):
    parsed_url = urlparse(url)
    query_params = parse_qs(parsed_url.query)
    prdt_no = query_params.get('prdtNo', [''])[0]
    return prdt_no
```

Gambar 3.11 Potongan Kode untuk Memecah URL Olive Young

Langkah selanjutnya ialah menyelesaikan penulisan alur program *scraping* gambar. Fungsi find_elements dari Selenium akan dilakukan menggunakan WebDriver untuk mencari CSS Selector yang memiliki nilai src image_elements yang telah dijelaskan di atas. Program kemudian dapat dibiarkan berjalan secara otomatis dan dapat diberhentikan sesuai kebutuhan. Berdasarkan penemuan, kode program akan terhenti apabila kekuatan sinyal internet *device* perusahaan mengalami kendala. Tangkapan layar kode di sajikan di Gambar 3.12 dan output program pada Gambar 3.13 di bawah ini.

```
def scrape_images(url, save_directory):
    # Set up Selenium WebDriver
    driver = webdriver.Chrome()
    driver.get(url)

    # Let the page load (you might need to adjust the sleep duration)
    time.sleep(15)

    # Check if the product is not found
    if "We can't find this product." in driver.page_source:
        prdt_no = get_prdt_number(url)
        prdt_directory = os.path.join(save_directory, f'prdt_{prdt_no}')
        os.makedirs(prdt_directory, exist_ok=True)
        print(f"Product not found. Created directory for prdtNo: {prdt_no}")
        driver.quit()
        return

    # Find image elements
    image_elements = driver.find_elements("css selector", 'img[src^="https://image.globaloliveyoungshop.com/prdtImg/"]')

    # Get prdtNo from the URL
    prdt_no = get_prdt_number(url)

    # Create the directory to save images
    os.makedirs(save_directory, exist_ok=True)
    prdt_directory = os.path.join(save_directory, f'prdt_{prdt_no}')
    os.makedirs(prdt_directory, exist_ok=True)

    # Keep track of downloaded images
    downloaded_images = set()

    # Download each image
    for idx, img_element in enumerate(image_elements):
        image_url = img_element.get_attribute('src')
        # Check if the image has been downloaded before
        if image_url in downloaded_images:
            print(f'Skipped: {image_url} - Image already downloaded')
            continue

        save_path = os.path.join(prdt_directory, f'image_{idx + 1}.jpg')
        download_image(image_url, save_path)
        print(f'Downloaded: {save_path}')
        # Add the downloaded image URL to the set
        downloaded_images.add(image_url)

    # Close the browser
    driver.quit()

def scrape_images_from_file(file_path, save_directory):
    with open(file_path, 'r') as url_file:
        urls = url_file.read().splitlines()

    for url in urls:
        scrape_images(url, save_directory)

if __name__ == "__main__":
    url_file_path = r"D:\Kerja\recommendation_skin_analyst\28-07-2023_ProductScraping\data\OliveYoung\Active_OY.txt"
    save_directory = r"D:\Kerja\recommendation_skin_analyst\28-07-2023_ProductScraping\data\OliveYoung\product-images"

    scrape_images_from_file(url_file_path, save_directory)
```

Gambar 3.12 Lanjutan Kode Scraping Gambar Produk Olive Young

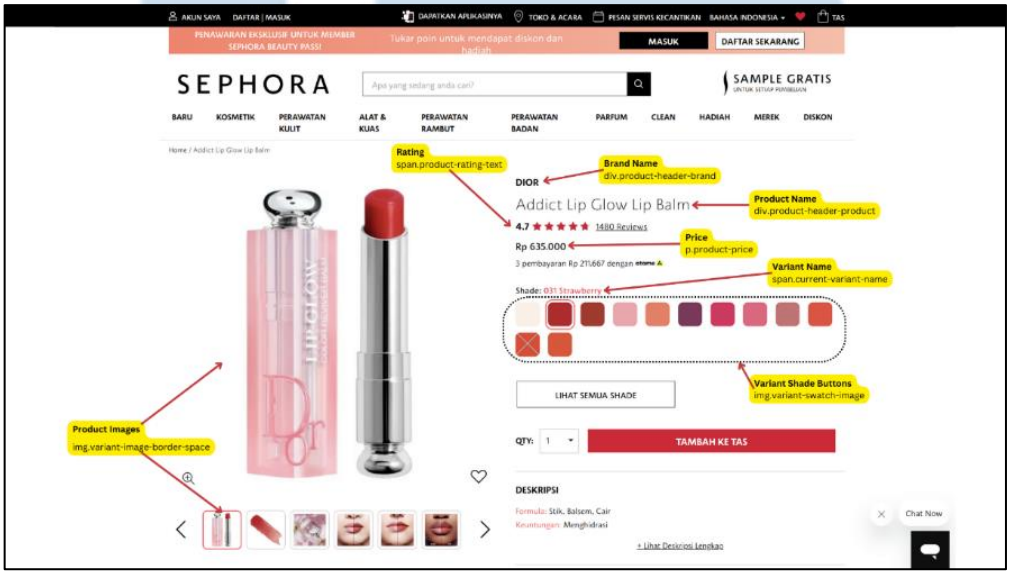
Gambar 3.13 Tampilan Output Program Scraping Gambar Olive Young

Program *scraping* awalnya dirancang untuk struktur alamat penyimpanan https://image.globaloliveyoungshop.com/prdtImg/xxxxxx. Namun setelah proses validasi produk yang gagal terunduh, ditemukan bahwa adanya perbedaan alamat penyimpanan yakni di dalam folder "mig". Kasus ini diselesaikan dengan merevisi variabel image_elements dan menjalankan ulang program pada *link* produk yang ada tapi masih gagal terunduh. Hasil akhir proyek *Scraping* Gambar Produk Olive Young ialah dari 7.877 tautan produk, sebanyak 6.417 berhasil diunduh dan 1460 tautan lainnya sudah tidak aktif.

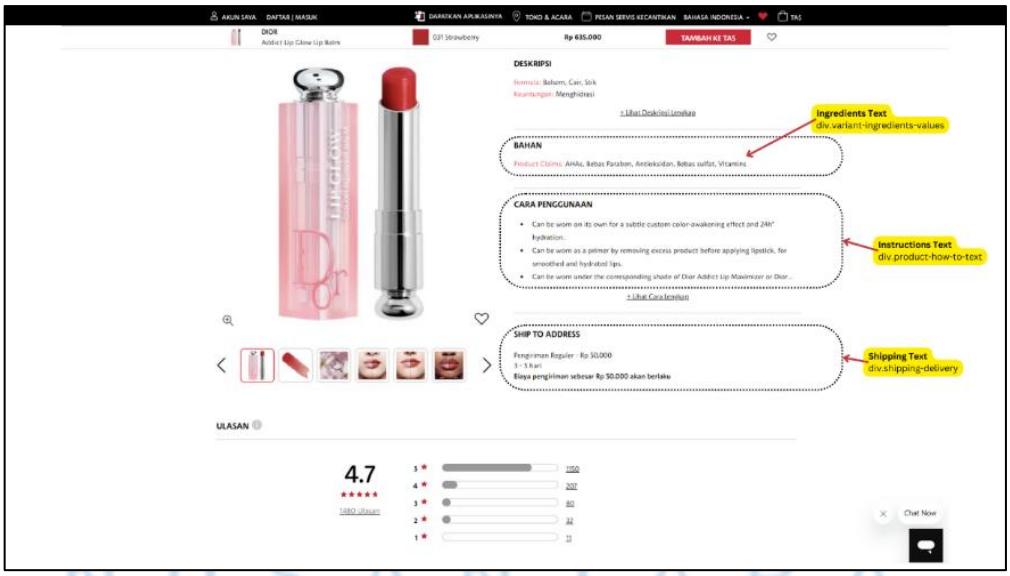
3.2.3 Proyek 2: Pengumpulan Gambar Produk Sephora Indonesia

Proyek kedua kerja magang *data engineer* dikerjakan sejak tanggal 26 Februari sampai dengan 8 Maret 2024. Penugasan yang diberikan untuk situs web https://www.sephora.co.id masih serupa dengan Proyek 1, namun ragam variasi dari produk turut diperintahkan untuk diunduh. Maka dilakukan analisis terhadap lebih banyak sampel tautan produk Sephora Indonesia (ID) untuk mengetahui apa saja identitas yang membedakan antar variasi produk dan metode tautan gambar dimuat oleh *script* HTML. Pengelompokan jenis tautan produk dilakukan, yaitu menjadi *basic size* (ukurannya satu), *variant size* (ukurannya banyak), dan *variant shade* (banyak jenis warna/*shade*) yang ditampakkan pada Gambar 3.14 hingga Gambar 3.17. Ditemukan bahwa cara pengguna merubah tampilan *variant shade* ialah bukan dengan *div button*, melainkan dengan menekan gambar-gambar yang memiliki element HTML

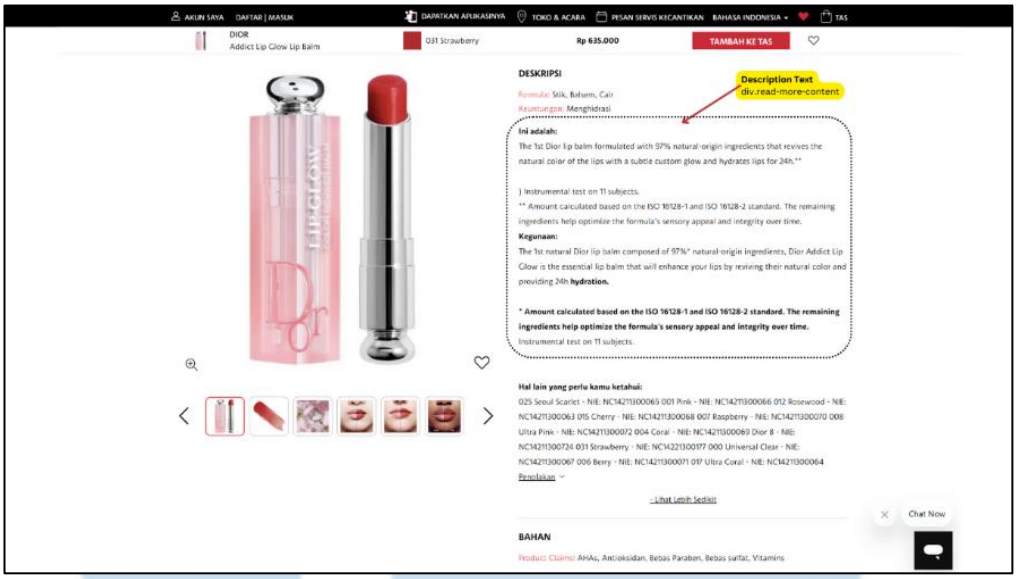
img.variant-swatch-image. Sedangkan produk dengan *variant size* menggunakan div.product-swatch sebagai tombol memilih *variant* dan berupa teks ukuran dari kemasan produk. Detail lain dari informasi produk juga memiliki element HTML-nya masing-masing dan digunakan saat proses pengumpulan metadata detail produk. Misalnya Nama Produk, Merek Produk, hingga spesifikasi lain.



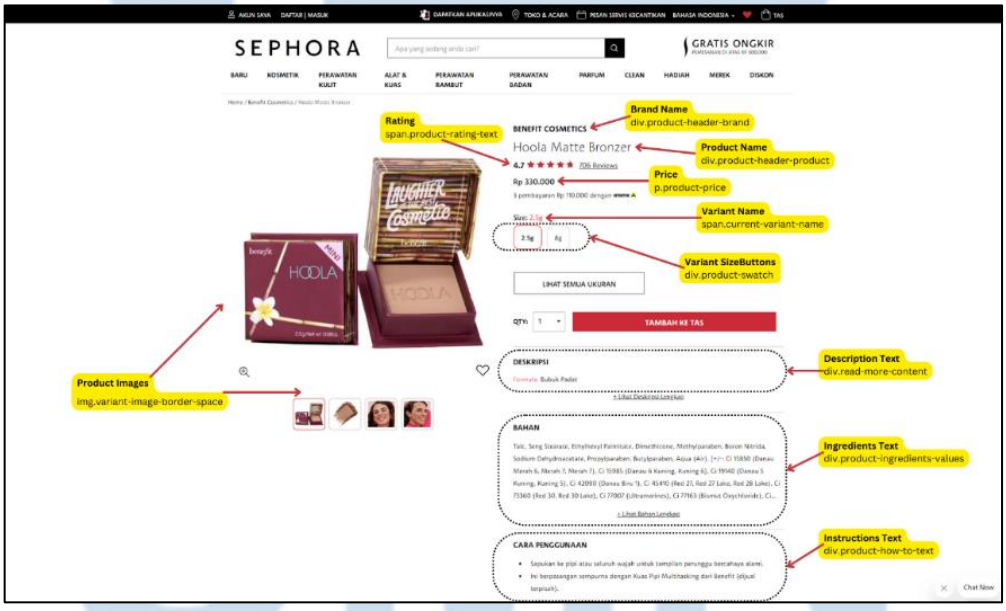
Gambar 3.14 Tampilan Halaman Produk Sephora ID dengan Variant Shade



Gambar 3.15 Tampilan Detail Informasi Produk di Sephora ID (1)



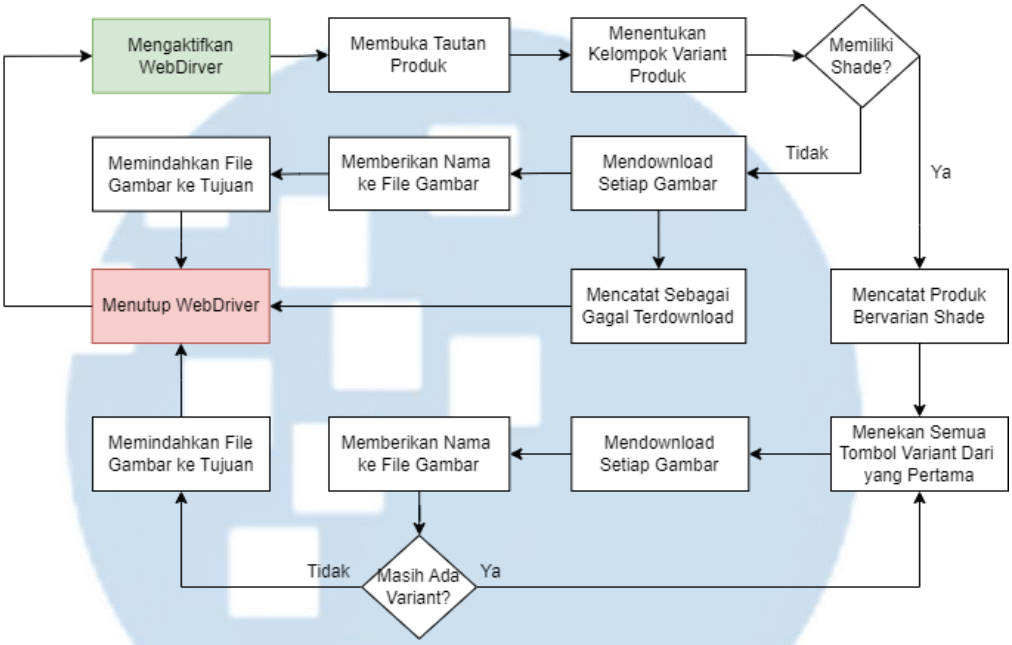
Gambar 3.16 Tampilan Detail Informasi Produk di Sephora ID (2)



Gambar 3.17 Tampilan Halaman Produk Sephora ID dengan Variant Size

Cara kerja dari program *scraping* Sephora ID ialah webdriver dari selenium akan melakukan iterasi membuka data tautan produk dan mengecek variant produk sebelum mengunduh gambar dengan spesifikasi yang diinginkan. Penamaan folder penyimpanan setiap produk diambil dari tautan produk itu sendiri berupa nama produk. Jika produk memiliki variant *shade* maka webdriver akan menekan tombol dan mulai mengunduh gambar hingga

seluruh variant produk terunduh (Gambar 3.18).



Gambar 3.18 Diagram Alur Siklus Program Image Scraping Gambar Sephora ID

Pustaka-pustaka dan modul Python dipanggil untuk program *image scraping* Sephora ID dengan susunan sebagai berikut.

1. PIL Image : Membuka dan menyimpan gambar.
2. OS : Manajemen direktori dokumen dan manipulasi Path.
3. Time : Pengaturan waktu operasi program.
4. IO : Manipulasi di dalam aliran data memori *device*.
5. Request : Mengunduh gambar dari website.
6. JSON : Manipulasi variabel string menjadi objek Python JSON.
7. Regex : Pencocokkan pola dalam string.
8. Pandas : Baca file CSV dan menulisnya ke dalam *data frame*.
9. WebDriver : Mengontrol browser website.
10. BY : Mendefinisikan kriteria pemilihan element dengan Xpath.
11. WebDriverWait : Menjeda proses WebDriver.
12. Expected_Conditions : Memeriksa element di Document Object Model.
13. ActionChains : Melakukan otomasi perintah klik element website.

Setelah alur dari logika program telah disetujui oleh *AI developer*, maka

penulisan kode program dapat dilakukan. Bagian pertama kode program (Gambar 3.19) melakukan penamaan folder penyimpanan gambar dari setiap tautan produk, misal https://www.sephora.co.id/products/allies-of-skin-multi-acids-and-retinoid-brightening-sleeping-facial/v/default. Kode ini memakai fungsi *split* untuk menghilangkan karakter *whitespace* dari awal dan akhir string link. Ini termasuk karakter newline (\n), spasi, dan tab. Hal ini dilakukan untuk memastikan bahwa link yang digunakan tidak memiliki karakter tambahan yang tidak diinginkan. Kemudian tautan dipisah-pisah menjadi kumpulan blok string berdasarkan posisinya dari karakter *slash* (/) lalu string yang berada pada 3 blok terakhir akan menjadi nama folder.

```
# Loop through each link
for link in links:
    # Bersihkan link dari karakter newline
    link = link.strip()

    # Ambil bagian terakhir dari URL sebagai nama subfolder
    subfolder_name = link.split("/")[-3]
    subfolder_directory = os.path.join(base_directory, subfolder_name)

    # Buat folder untuk setiap subfolder
    os.makedirs(subfolder_directory, exist_ok=True)
```

Gambar 3.19 Potongan Kode untuk Penamaan Folder Produk

```
    # Buka link website
    driver.get(link)

    # Tunggu sampai gambar dimuat
    time.sleep(10)

    # Cari semua gambar yang memiliki class yg diminta
    images = driver.find_elements(By.CSS_SELECTOR, "img.variant-image-border-space")

    # Hitung jumlah gambar
    count = 1

    # Cari elemen menggunakan XPath
    xpath_pattern = "//li[contains(@class, 'product-variant-swatch') and contains(@class, 'product-vari
    elements = driver.find_elements(By.XPATH, xpath_pattern)

    # Jika elemen ditemukan, catat link ke dalam file
    if elements:
        with open(found_links_file, "a") as f:
            f.write(link + "\n")
```

Gambar 3.20 Kode Program Image Scraping Sephora ID - Find Elements

```
    # Download setiap gambar
    for image in images:
        # Dapatkan url gambar
        image_url = image.get_attribute("src")

        # Dapatkan dua kata terakhir dari atribut alt
        alt_text = image.get_attribute("alt")

        # Buat nama file
        filename = f"image_{count}_{alt_text}.png"

        # Jika url gambar tidak ada
        if not image_url:
            failed_links.append(link)
            continue

        # Download gambar dan simpan ke folder
        try:
            response = requests.get(image_url, stream=True)
            image_data = response.content

            # Baca data gambar sebagai gambar PIL untuk memeriksa dimensi
            img = Image.open(io.BytesIO(image_data))

            # Periksa dimensi gambar
            if img.width == 1000 and img.height == 1000:
                with open(os.path.join(subfolder_directory, filename), "wb") as f:
                    f.write(image_data)
                success_links.append(link)
            else:
                failed_links.append(link)
        except Exception as e:
            print(f"Gagal mengunduh gambar dari link: {link}, dengan kesalahan: {e}")
            failed_links.append(link)

        # Hitung gambar selanjutnya
        count += 1

# Tutup browser
driver.quit()
```

Gambar 3.21 Kode Program Image Scraping Sephora ID – Download Default

Program *image scraping* dirancang secara default untuk mengunduh produk dengan *variant size* dan bekerja melalui beberapa tahap sebagai ditampilkan pada Gambar 3.20 dan Gambar 3.21. Pertama, program menggunakan Selenium WebDriver untuk membuka tautan yang telah ditentukan. Ini memungkinkan akses ke halaman web yang akan di-scrape. Selanjutnya, program mencari element dengan class dengan fungsi XPath dari Selenium berupa "img.variant-image-border-space" pada script situs web, yang merupakan lokasi gambar produk yang ditargetkan untuk di-*scraping*. Apabila program menemukan element yang memenuhi kriteria "//li[contains(@class, 'product-variant-swatch') and contains(@class, 'product-variant-shade') and contains(@class, 'product-variant-swatch-')", berarti menandakan bahwa produk memiliki *variant shade*. Dalam situasi ini, program akan mencatat tautan tersebut untuk diproses lebih lanjut saat program khusus *variant shade*

dijalankan.

Halaman produk Sephora ID memiliki dua ukuran dimensi gambar, dan gambar dengan dimensi 1000x1000 pixel dipilih. Nama file gambar diambil dari atribut "alt" yang dimiliki oleh class gambar produk. Cara penamaan ini dapat memastikan bahwa nama file tersebut bermakna dan dapat diidentifikasi dengan informasi varian produk yang telah dimiliki.

```
# Jika menemukan elemen, lanjutkan dengan proses klik dan unduh gambar
for element in elements:
    try:
        # Klik elemen
        action.click(element).perform()

        # Tunggu hingga gambar muncul dalam DOM
        WebDriverWait(driver, 10).until(EC.presence_of_all_elements_located((By.CSS_SELECTOR, "img.variant-image-border-space"))

        # Cari semua gambar yang memiliki class yang diminta
        images = driver.find_elements(By.CSS_SELECTOR, "img.variant-image-border-space")

        # Hitung jumlah gambar
        count = 1

        # Download setiap gambar
        for image in images:
            # Dapatkan URL gambar
            image_url = image.get_attribute("src")

            # Dapatkan dua kata terakhir dari atribut alt
            alt_text = image.get_attribute("alt")

            # Buat nama file
            filename = f"image_{count}_.png"

            # Jika URL gambar tidak ada
            if not image_url:
                failed_links.append(image_url)
                continue

            # Download gambar dan simpan ke folder
```

Gambar 3.22 Tambahan Kode untuk Pengunduhan Variant Shade Sephora ID

Untuk produk dengan *variant shade*, WebDriver dirancang untuk melakukan iterasi aksi klik menggunakan fungsi action.click dari Selenium seperti pada Gambar 3.22. WebDriver akan mengklik setiap varian dan mengunduh gambar sesuai dengan varian yang sedang dipilih WebDriver. Dengan demikian, program ini memastikan bahwa semua gambar varian produk terunduh dan tersimpan dengan baik sesuai dengan kategorinya.

Dari 2.647 tautan produk yang diproses, ditemukan 17 tautan yang gagal diunduh karena atribut "alt" pada gambar memiliki karakter *slash* (/) yang tidak dapat digunakan sebagai nama file. Untuk mengatasi masalah ini, dilakukan pengunduhan manual dan penghapusan karakter *slash* dari nama file. Selain itu, ditemukan bahwa sebanyak 391 tautan produk Sephora ID sudah tidak aktif.

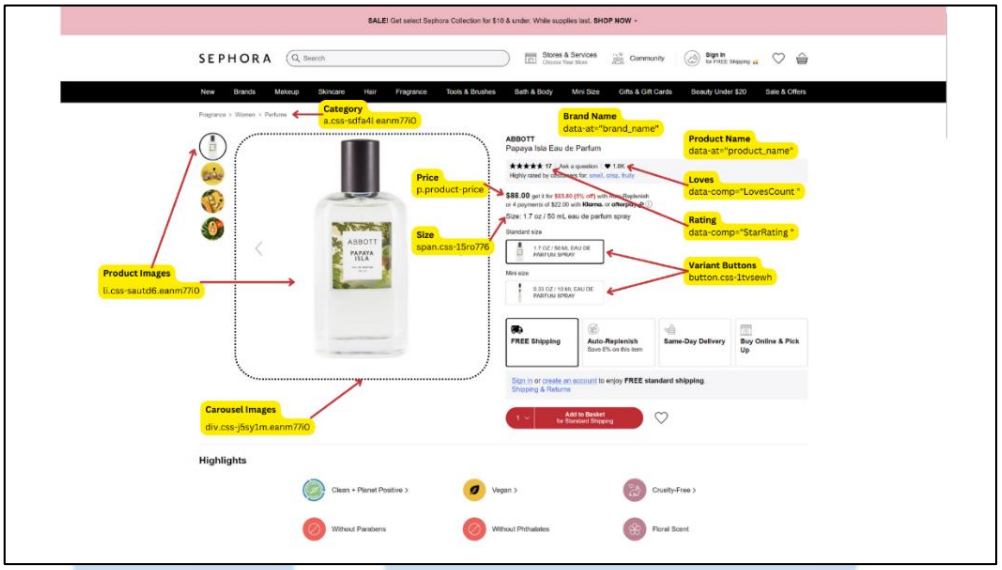
Metode otomatisasi ini tidak hanya mempercepat proses pengunduhan

gambar tetapi juga memastikan bahwa setiap varian produk terdokumentasi dengan baik. Namun, beberapa kendala teknis seperti karakter tidak valid dalam nama file dan tautan yang tidak aktif memerlukan intervensi manual untuk memastikan kelengkapan dan akurasi data yang diunduh. Dengan menyelesaikan masalah ini, integritas dan kualitas data dapat tetap terjaga, memungkinkan analisis lebih lanjut atau penggunaan data dengan lebih efektif.

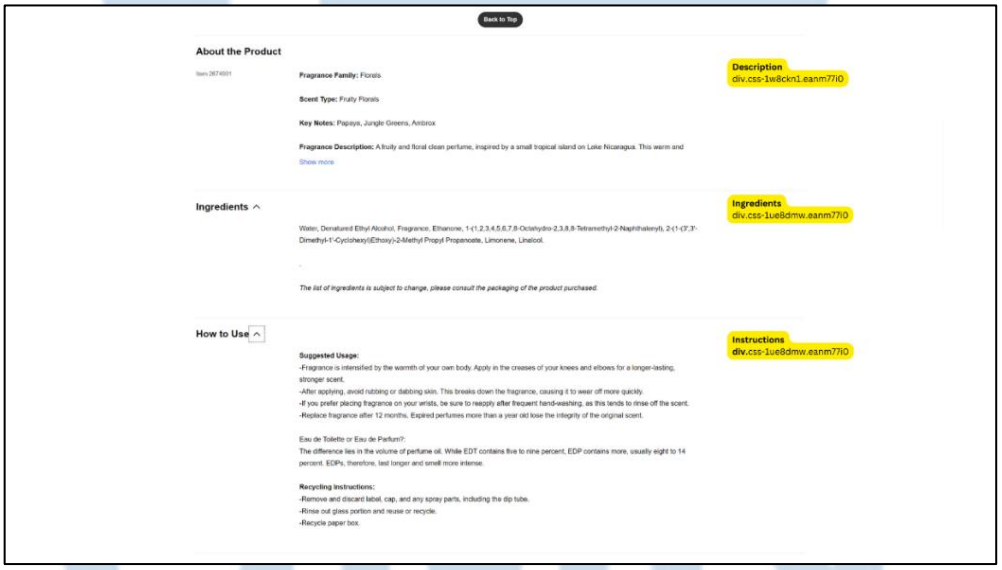
3.2.4 Proyek 3 Pengumpulan Gambar Produk Sephora USA

Proyek *data image scraping* dilanjutkan dengan mengumpulkan gambar-gambar produk dari website https://www.sephora.com. Website tersebut ialah web resmi cabang utama dari perusahaan ritel industri kecantikan Sephora, berdomain di Amerika Serikat dan Kanada. Proses *scraping* berlangsung cukup lama, yaitu sejak tanggal 11 Maret hingga 26 April 2024. Lamanya proses *scraping* ini disebabkan kompleksnya struktur dari halaman website Sephora itu sendiri. Gambar 3.23 hingga Gambar 3.26 merupakan struktur dari script HTML website. Sephora USA juga memiliki dua versi website, *mobile & desktop*, yang sempat menuai kebingungan.

Meskipun menghadapi berbagai tantangan teknis, proyek ini bertujuan untuk memastikan bahwa semua gambar produk dapat dikumpulkan dengan lengkap dan akurat. Kompleksitas struktur HTML dan perbedaan antara versi mobile dan desktop mengharuskan penggunaan teknik *scraping* yang lebih fleksibel. Dengan pendekatan yang tepat, seluruh data gambar dapat diunduh, memungkinkan analisis lebih untuk berbagai keperluan, termasuk penelitian pasar, pengembangan produk, dan pemasaran. Namun, KSH akan menggunakan data produk Sephora USA yang diproduksi dari negara Korea Selatan, selaras dengan lini bisnis perusahaan. Proyek ini juga menunjukkan krusialnya pemahaman mendalam tentang struktur web dan kemampuan teknis untuk menyesuaikan metode scraping dengan kondisi spesifik situs web yang dituju. Keberhasilan proyek ini akan memberikan data yang berharga dan mendalam tentang produk-produk kecantikan yang ditawarkan oleh Sephora di Amerika Serikat dan Kanada.

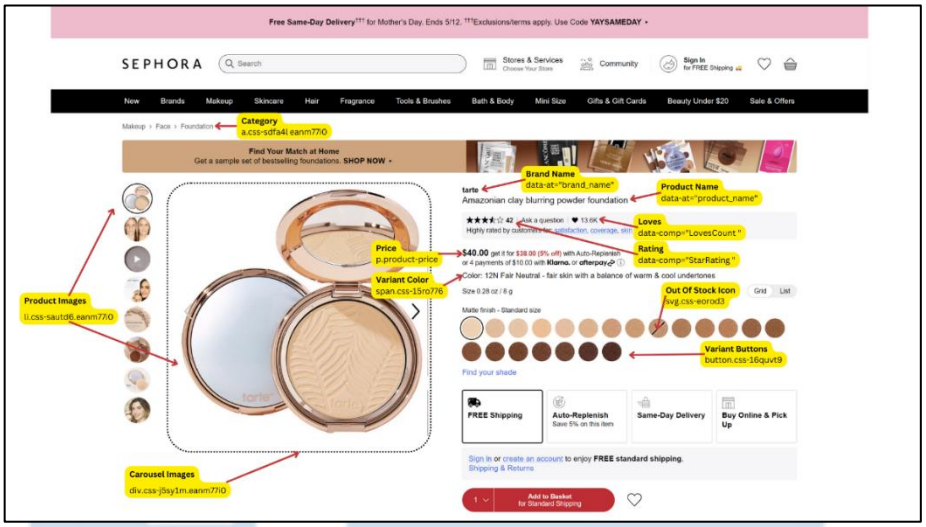


Gambar 3.23 Tampilan Halaman Produk Sephora USA dengan Variant Size



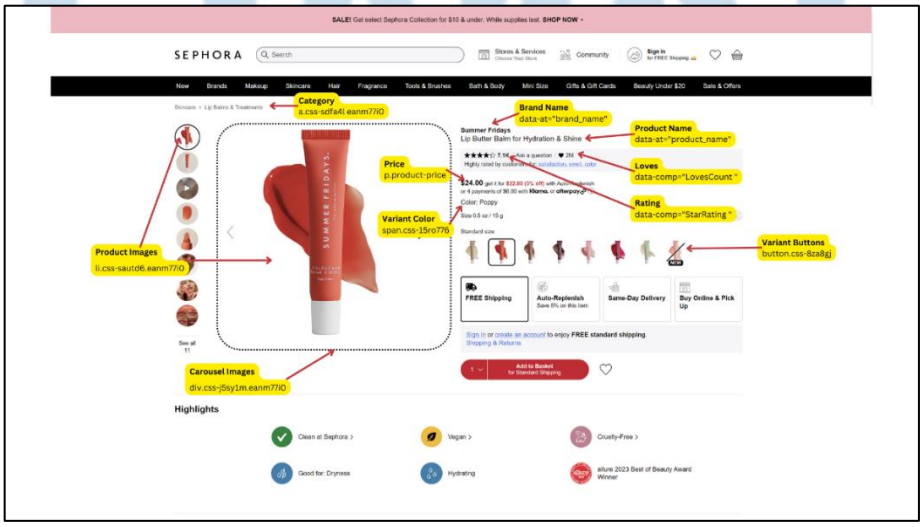
Gambar 3.24 Tampilan Bagian About the Product Halaman Produk Sephora USA

Terdapat 3 jenis penamaan tombol varian, pertama ialah *variant size* (Gambar 3.23) menggunakan button.css-1tvsewh dengan ciri-ciri berbentuk persegi panjang dan menampilkan gambar produk disertai keterangan nama variannya di sebelah kanan. Produk dapat memiliki satu atau lebih v*ariant size* dengan struktur script website yang sama.



Gambar 3.25 Tampilan Halaman Produk Sephora USA dengan Variant Color Versi 1

Jenis penampilan tombol varian kedua ialah *variant color* versi 1 (Gambar 3.25) menggunakan button.css-16quvt9 dengan ciri-ciri berbentuk bulat dan menampilkan warna dari variannya. Produk dengan *variant color* juga dapat memiliki satu atau lebih ukuran, tetapi masih memiliki struktur script HTML yang sama dengan *variant color*. Tombol varian akan ditimpa dengan element svg.css-eorod3 untuk menandakan bahwa stok varian produk itu habis. Jenis penampilan tombol varian terakhir ialah *variant color* versi 2 (Gambar 3.26). Tombol varian versi ini menggunakan element button.css-8za8gj dengan ciri-ciri berbentuk persegi disertai gambar produk ditengah.



Gambar 3.26 Tampilan Halaman Produk Sephora USA dengan Variant Color Versi 2

Alur dari program *scraping* gambar produk situs web https://www.sephora.com sama dengan Sephora ID, namun awalnya program uji coba/*trial and error* hanya dibuat untuk pengunduhan *variant size*. Ditemukan bahwa Sephora.com memiliki dua mode *user interface* untuk menyesuaikan dengan kebutuhan pengguna, yakni mode *mobile* dan *desktop*. Kedua mode ini memiliki perbedaan dari segi struktur dan semua element *class* yang digunakan. WebDriver membuka jendela website sebesar setengah layar, sehingga menyebabkan program mengalami kegagalan. Setelah mencari informasi lebih, fungsi maximize_window dari Selenium WebDriver ditambahkan (Gambar 3.27**Error! Reference source not found.**) sehingga program dapat membuat situs web menjadi mode UI desktop dan berhasil dijalankan.

```
def restart_webdriver():
    global driver
    driver.quit()
    driver = webdriver.Chrome()
    driver.maximize_window()
```

Gambar 3.27 Potongan Kode Penyesuaian WebDrive

Dengan jumlah tautan produk yang lebih banyak, ditemukan bahwa jika terdapat dua atau lebih element "div.css-b83rh7", maka produk tersebut memiliki banyak varian ukuran yang bentuk wadahnya berbeda-beda. Program perlu mencatat informasi ini untuk diproses lebih lanjut sebagai "multiple variant size". Element HTML yang memuat tautan gambar dan atributnya berada pada "li.css-sautd6.eanm77i0". Namun, element ini hanya akan muncul jika element carousel gambar "css-j5sy1m.eanm77i0" telah termuat terlebih dahulu. Oleh karena itu, program harus memastikan bahwa element carousel gambar telah termuat sebelum mencoba mengakses tautan gambar dan atribut yang dimilikinya.

Selain itu, jika muncul kalimat "Sorry, this product is not available.", ini menandakan bahwa tautan produk sudah tidak aktif atau telah diubah oleh

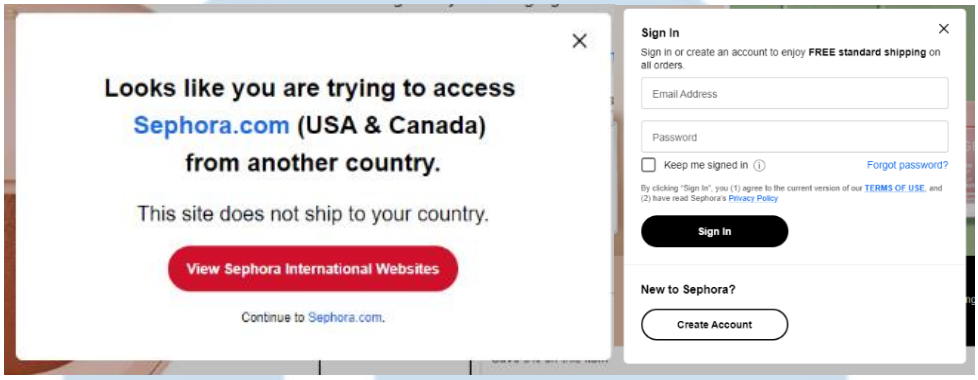
Sephora. Informasi ini juga perlu dicatat oleh program untuk memastikan keakuratan dan kelengkapan data.Pustaka-pustaka dan modul Python dipanggil untuk program *image scraping* Sephora USA dengan susunan sebagai berikut.

1. PIL Image : Membuka dan menyimpan gambar.
2. OS : Manajemen direktori dokumen dan manipulasi Path.
3. Time : Pengaturan waktu operasi program.
4. IO : Manipulasi di dalam aliran data memori *device*.
5. Request : Mengunduh gambar dari website.
6. JSON : Manipulasi variabel string menjadi objek Python JSON.
7. Regex : Pencocokkan pola dalam string.
8. Pandas : Baca file CSV dan menulisnya ke dalam *data frame*.
9. WebDriver : Mengontrol browser website.
10. BY : Mendefinisikan kriteria pemilihan element dengan XPath.
11. WebDriverWait : Menjeda proses WebDriver.
12. NoSuchElementException : Pengecualian tautan jika element tidak ada.
13. ActionChains : Melakukan otomasi perintah klik element website.

Program *scraping* dijalankan dari kantor KSH yang berlokasi di Indonesia, sehingga website Sephora.com menampilkan pop up untuk mengalihkan pengguna ke website mereka untuk region Indonesia (Sephora.co.id), seperti pada Gambar 3.28. Pop up lain yang tampil secara otomatis ialah Sign In akun Sephora. Maka program dirancang untuk melakukan click kedua tombol close di bawah ini yang secara otomatis tampil memenuhi layar, yaitu element "button[data-at='modal_close']" dan "button[data-at='close_button']". Pentingnya merancang program yang adaptif terhadap *pop-up* ini adalah untuk memastikan kelancaran dan efisiensi proses scraping. Tanpa pengelolaan otomatis untuk menutup *pop-up* tersebut, program scraping akan terganggu dan tidak dapat berjalan dengan optimal.

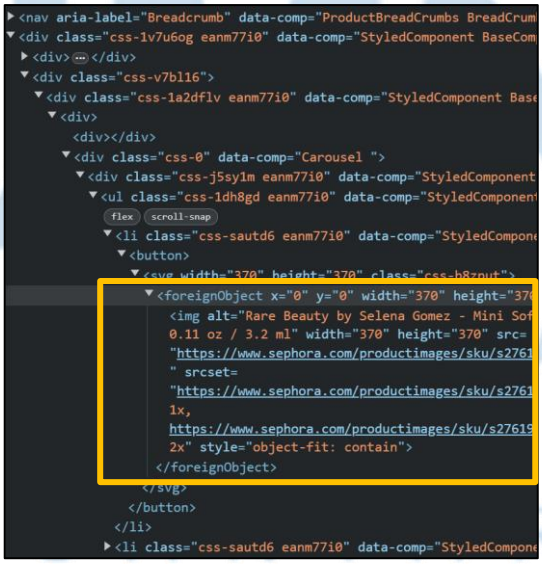
Dengan menutup *pop-up* secara otomatis, program dapat fokus pada tugas utamanya, yaitu mengumpulkan data gambar produk dari situs Sephora.com. Solusi ini menunjukkan perlunya fleksibilitas dan penyesuaian

dalam pemrograman scraping untuk mengatasi berbagai hambatan yang mungkin muncul selama proses pengumpulan data dari situs web internasional yang memiliki fitur regionalisasi dan interaksi pengguna yang kompleks.



Gambar 3.28 Tampilan Pop Up Saat Tautan Produk Sephora USA Dibuka

Setelah melaui rangkaian tahapan *trial & error*, akhirnya kode program berhasil disempurnakan agar dapat memenuhi kebutuhan untuk melakukan *scraping* gambar produk Sephora USA. Namun, logika program ini cukup berbeda dengan program *scraping* Sephora ID, yakni program ini mencatat tautan dari gambar seperti pada Gambar 2.4.1 dan dibatasi dengan bingkai warna kuning. Kumpulan tautan gambar ini nanti akan diunduh dengan program lain setelah semuanya terkumpul.



Gambar 2.4.1 Lokasi Tautan Gambar Pada Script Website Sephora USA

Setelah menulis pengimporan pustaka yang digunakan dan menentukan direktori tempat tautan produk, penulisan kode untuk proses-proses sebelum *scraping* tautan gambar dilanjutkan. Gambar 3.29 menampilkan potongan kode untuk menutup kedua close buttons yang telah dijelaskan semua dan memeriksa apakah produk masih tersedia.

```
# Inisialisasi DataFrame untuk menyimpan informasi gambar
image_info_df = pd.DataFrame(columns=["web_link", "image_link", "alt"])

# Loop through each link
for link in links:
    print(f"Memproses link: {link}")

    # Bersihkan link dari karakter newline
    link = link.strip()

    try:
        # Buka link website
        driver.get(link)

        # Tunggu sampai tombol modal close dimuat dan klik
        try:
            modal_close_button = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.CSS_SELECTOR,
                                                                                              "button[data-at='modal_close']")))
            modal_close_button.click()
        except TimeoutException:
            print("")

        # Tunggu sampai tombol close_button dimuat
        try:
            close_button = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.CSS_SELECTOR,
                                                                                        "button[data-at='close_button']")))
            close_button.click()
        except TimeoutException:
            print("")

        # Tunggu sampai gambar dimuat
        time.sleep(10)

        # Periksa apakah produk tidak tersedia
        if "Sorry, this product is not available." in driver.page_source:
            print("Produk tidak tersedia:", link)
            failed_links.append({"link": link, "reason": "Produk tidak tersedia"})
            continue
```

Gambar 3.29 Potongan Kode Otomasi Klik Close Buttons

Kemudian diputuskan bahwa untuk menjalankan kode program untuk produk dengan *variant color* seperti ditampilkan pada Gambar 3.30. Fungsi ActionChains digunakan untuk menekan setiap button varian color yaitu "button.css-8za8gj" atau "button.css-16quvt9". Dimulai dari yang pertama dalam hirarki tombol varian di dalam div data-comp="SwatchGroup ". Setiap sekali klik tombol varian, fungsi find_elements dari Selenium akan mencari lokasi tautan gambar produk dengan target CSS_SELECTOR yang telah ditentukan. Atribut-atribut dari element gambar yang diambil ialah img, src, dan alt. Ketiga class CSS ini menyimpan kebutuhan yaitu tautan gambar dan teks *alternative* yang menjelaskan gambar apakah yang dimaksud. Teks alt juga akan digunakan dicatat ke dalam CSV. File CSV yang dimaksud

image_link.csv, berisi kolom web_link (tautan produk), image_link (tautan gambar), dan alt (teks alt gambar). Bagian terakhir dari cuplikan kode di Gambar 3.30 ialah proses pencatatan informasi yang sama namun ketika Selenium tidak menemukan tanda produk memiliki *variant color*. Proses ini berfungsi agar apabila ada produk yang hanya memiliki satu ukuran, produk tersebut akan tetap dicatat oleh program.

```
        # Periksa apakah terdapat banyak variant size
        swatch_groups = driver.find_elements(By.CSS_SELECTOR, "div.css-b83rh7")
        if len(swatch_groups) >= 2:
            print("Produk memiliki banyak variant size:", link)

        # Periksa apakah terdapat variant shade
        shade_buttons = driver.find_elements(By.CSS_SELECTOR, "button.css-8za8gj, button.css-16quvt9")
        if shade_buttons:
            print("Produk memiliki variant shade:", link)
            # Inisialisasi ActionChains
            action = ActionChains(driver)
            # Loop melalui setiap button variant shade
            for shade_button in shade_buttons:
                try:
                    # Klik button variant shade
                    action.click(shade_button).perform()
                    # Tunggu sebentar agar halaman mengganti gambar
                    time.sleep(5)
                    # Tunggu sampai elemen dengan class "css-j5sy1m eanm77i0" muncul
                    image_div = WebDriverWait(driver, 2).until(EC.presence_of_element_located((By.CSS_SELECTOR,
                                                                                               "div.css-j5sy1m.eanm77i0")))
                    image_elements = image_div.find_elements(By.CSS_SELECTOR, "li.css-sautd6.eanm77i0")
                    # Loop melalui semua elemen gambar dan ambil link serta atribut lainnya
                    for image_element in image_elements:
                        img_tag = image_element.find_element(By.TAG_NAME, "img")
                        image_src = img_tag.get_attribute("src")
                        image_alt = img_tag.get_attribute("alt")
                        # Tambahkan informasi gambar ke dalam list
                        image_info_list.append({
                            "web_link": link,
                            "image_link": image_src,
                            "alt": image_alt
                        })
                except Exception as e:
                    print(f"Error saat mengklik shade button: {e}")
                    continue

        else:
            # Tunggu sampai elemen dengan class "css-j5sy1m eanm77i0" muncul
            image_div = WebDriverWait(driver, 2).until(EC.presence_of_element_located((By.CSS_SELECTOR, "div.css-j5sy1m.eanm77i0"
            image_elements = image_div.find_elements(By.CSS_SELECTOR, "li.css-sautd6.eanm77i0")
            # Loop melalui semua elemen gambar dan ambil link serta atribut lainnya
            for image_element in image_elements:
                img_tag = image_element.find_element(By.TAG_NAME, "img")
                image_src = img_tag.get_attribute("src")
                image_alt = img_tag.get_attribute("alt")
                # Tambahkan informasi gambar ke dalam list
                image_info_list.append({
                    "web_link": link,
                    "image_link": image_src,
                    "alt": image_alt
                })

        success_links.append(link)
```

Gambar 3.30 Potongan Kode Scraping Tautan Gambar Sephora USA Variant Color

Sebelumnya telah dijelaskan bahwa program mencatat tautan produk yang *multiple variant size*. Tautan-tautan tersebut diambil dari output program *variant color* untuk digunakan pada program khusus untuk *multiple variant size* (Gambar 3.31). Perubahan yang dilakukan hanya pada isi dari CSS SELECTOR dan fungsi exception saat tombol varian gagal ditekan.

```
        # Periksa apakah terdapat banyak variant size
        swatch_groups = driver.find_elements(By.CSS_SELECTOR, "button.css-1sn75vo, button.css-1tvsewh")
        if swatch_groups:
            print("Produk memiliki banyak variant size:", link)
            action = ActionChains(driver)
            # Loop melalui setiap button variant color
            for swatch_group in swatch_groups:
                try:
                    # Klik button variant shade
                    action.click(swatch_group).perform()
                    # Tunggu sebentar agar halaman mengganti gambar
                    time.sleep(5)
                    # Tunggu sampai elemen dengan class "css-j5sy1m eanm77i0" muncul
                    image_div = WebDriverWait(driver, 2).until(EC.presence_of_element_located((By.CSS_SELECTOR,
                                                                                               "div.css-j5sy1m.eanm77i0")))
                    image_elements = image_div.find_elements(By.CSS_SELECTOR, "li.css-sautd6.eanm77i0")
                    # Loop melalui semua elemen gambar dan ambil link serta atribut lainnya
                    for image_element in image_elements:
                        img_tag = image_element.find_element(By.TAG_NAME, "img")
                        image_src = img_tag.get_attribute("src")
                        image_alt = img_tag.get_attribute("alt")
                        # Tambahkan informasi gambar ke dalam list
                        image_info_list.append({
                            "web_link": link,
                            "image_link": image_src,
                            "alt": image_alt
                        })
                except Exception as e:
                    print(f"Error saat mengklik size button: {e}")
                    continue
```

Gambar 3.31 Potongan Kode Scraping Tautan Gambar Sephora USA Multiple Variant Size

Tautan-tautan gambar yang telah di-*scraping* dicatat ke dalam file CSV beserta dengan tautan produknya dan atribut “alt” yang dimiliki gambar ke dalam file image_link.csv. Adapun setiap tautan produk yang gagal diproses akan dicatat ke dalam file failed_links.csv seperti pada Gambar 3.32. Contoh output saat program *scraping* tautan gambar Sephora USA dijalankan disajikan pada Gambar 3.33.

```
    except TimeoutException:
        print("TimeoutException: Gagal memuat elemen pada halaman:", link)
        failed_links.append({"link": link, "reason": "TimeoutException"})
    except Exception as e:
        print("Terjadi kesalahan:", e)
        failed_links.append({"link": link, "reason": "Terjadi kesalahan"})

    # Tutup dan buka kembali webdriver
    restart_webdriver()


# Jika ada informasi gambar baru, tambahkan ke DataFrame dan tulis ke file CSV
if image_info_list:
    new_image_info_df = pd.DataFrame(image_info_list)
    image_info_df = pd.concat([image_info_df, new_image_info_df], ignore_index=True)
    image_info_df.to_csv("C:\\Users\\Unnispick\\Downloads\\image_link.csv", index=False)

# Simpan failed_links ke dalam file CSV baru atau tambahkan ke file yang sudah ada
failed_links_df = pd.DataFrame(failed_links)
if os.path.exists("C:\\Users\\Unnispick\\Downloads\\failed_links.csv"):
    failed_links_df.to_csv("C:\\Users\\Unnispick\\Downloads\\failed_links.csv", mode='a', header=False, index=False)
else:
    failed_links_df.to_csv("C:\\Users\\Unnispick\\Downloads\\failed_links.csv", index=False)
```

Gambar 3.32 Potongan Kode Penutup Program Scraping Tautan Gambar Sephora USA

Gambar 3.33 Contoh Tampilan Output Program Scraping Tautan Gambar Produk Sephora USA

Tahapan berikutnya adalah membuat program (Gambar 3.34.) untuk menghapus baris data yang pada atribut “alt”nya bernilai “Video”. Sebab tautan video tersebut hanyalah *thumbnail* dari video produk yang disediakan Sephora USA. Kemudian seluruh tautan dipisahkan berdasarkan variannya dan mengecek jumlah produk yang gagal ter-*scraping*. Dengan metode ini, dapat diketahui produk mana yang perlu diproses ulang baik dengan program yang disesuaikan atau secara manual. Program pemisahan data tautan gambar ini dapat dilihat pada Gambar 3.35. Data tautan produk yang telah diproses akan dikelompokkan berdasarkan statusnya varian dan gagal tidaknya saat diproses.

Cleaned from "Video" images

In [1]:
```
import pandas as pd

# Baca file CSV
file_path = "C:\\Users\\Unnispick\\Downloads\\image_info.csv"
df = pd.read_csv(file_path)

# Hapus baris dengan nilai 'video' di kolom 'alt'
df = df[df['alt'] != 'Video']

# Simpan kembali dataframe ke file CSV
df.to_csv("C:\\Users\\Unnispick\\Downloads\\image_info_cleaned.csv", index=False)

print("Data yang memiliki nilai 'Video' di kolom 'alt' telah dihapus.")
```

```
Data yang memiliki nilai 'Video' di kolom 'alt' telah dihapus.
```

Gambar 3.34 Cleaning Data Tautan Gambar Produk Sephora USA

Categorized Links

In [12]:

```
import pandas as pd

# Membaca file Excel
file_path = r'C:\Users\Unnispick\Downloads\ProductStatus.xlsx'
df = pd.read_excel(file_path)

# Membuat DataFrame kosong untuk setiap kategori
df_size = pd.DataFrame(columns=['ProductStatus'])
df_shade = pd.DataFrame(columns=['ProductStatus'])
df_not_found = pd.DataFrame(columns=['ProductStatus'])
df_timeout = pd.DataFrame(columns=['ProductStatus'])

# Memisahkan link produk ke dalam masing-masing kategori DataFrame
for index, row in df.iterrows():
    product_link = row['ProductStatus'].split(': ')[1]
    if 'banyak variant size' in row['ProductStatus']:
        df_size = pd.concat([df_size, pd.DataFrame({'ProductStatus': [product_link]})], ignore_index=True)
    elif 'variant shade' in row['ProductStatus']:
        df_shade = pd.concat([df_shade, pd.DataFrame({'ProductStatus': [product_link]})], ignore_index=True)
    elif 'tidak tersedia' in row['ProductStatus']:
        df_not_found = pd.concat([df_not_found, pd.DataFrame({'ProductStatus': [product_link]})], ignore_index=True)
    elif 'TimeoutException' in row['ProductStatus']:
        df_timeout = pd.concat([df_timeout, pd.DataFrame({'ProductStatus': [product_link]})], ignore_index=True)

# Menyimpan ke dalam file CSV
df_size.to_csv('ProductVariantSize.csv', index=False)
df_shade.to_csv('ProductVariantShade.csv', index=False)
df_not_found.to_csv('ProductNotFound.csv', index=False)
df_timeout.to_csv('TimeoutException.csv', index=False)

print("Files have been created successfully.")
```

```
Files have been created successfully.
```

Gambar 3.35 Kode untuk Validasi Hasil Scraping Tautan Gambar dan Memisahkannya

Tahapan terakhir dari proyek 3 *scraping* gambar produk Sephora USA ialah mengunduh gambar produk menggunakan tautan yang telah didapatkan. Pustaka Python OS, CSV, Request, dan urlparse digunakan dalam program ini. Kemudian variabel User-Agent diberikan sebagai Headers program, yaitu berupa “Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.0.0”. Pemberian variabel ini berguna untuk mengidentifikasi WebDriver dengan ciri khas sebuah komputer saat membuat permintaan HTTP. Pemberian identitas ini juga berguna untuk melewati pemblokiran Sephora karena aktivitas yang mencurigakan yakni mengunduh file gambar yang sangat banyak menggunakan fungsi request.get. Selanjutnya program ditulis untuk memeriksa status aktif dari tautan. *Status code* 200 akan ditampilkan oleh website menandakan bahwa server web dapat memberikan file gambar yang di-*request* oleh kode program pengunduhan gambar, kode ini dapat dilihat pada Gambar 3.36.

```python
def create_folder(folder_name):
    if not os.path.exists(folder_name):
        os.makedirs(folder_name)

def download_image(image_url, folder_name, file_name):
    try:
        response = requests.get(image_url, headers=headers)
        if response.status_code == 200:
            with open(os.path.join(folder_name, file_name), 'wb') as f:
                f.write(response.content)
            print(f"Downloaded: {file_name} in folder {folder_name}")
        else:
            print(f"Failed to download: {file_name}. Response code: {response.status_code}")
    except Exception as e:
        print(f"Failed to download: {file_name}. Error: {e}")
```

Gambar 3.36 Potongan Kode untuk Mengunduh Seluruh Gambar

Pengguna program dapat menyesuaikan sumber file CSV tautan gambar dan alamat folder penyimpanan gambar dengan menyesuaikan variabel csv_file dan IMAGE_DIR. Adapun penamaan gambar ialah berdasarkan seperti yang terlihat pada Gambar 3.37, yaitu menggunakan teks “alt” yang dimiliki setiap gambar dan semua karakter yang tidak dapat dijadikan nama file diubah menggunakan fungsi Replace. Misalnya karakter spasi menjadi simbol *underscore* (_), tanda (<) dihapus, dan lain sebagainya.

```python
# Open CSV file
with open(csv_file, 'r', newline='', encoding='latin-1') as file:
    reader = csv.DictReader(file)
    for row in reader:
        web_link = row['web_link']
        image_link = row['image_link']
        alt = row['alt']

        # Create folder based on web_link
        folder_name = urlparse(web_link).path.strip('/')
        folder_path = os.path.join(IMAGE_DIR, folder_name)
        create_folder(folder_path)

        # Download image
        image_name = alt.replace(' ', '_').replace('/', '_').replace(',', '').replace('?','_').replace('<','').replace('>','')
        download_image(image_link, folder_path, image_name)
```

Gambar 3.37 Potongan Kode untuk Menyimpan Gambar Produk Sephora USA

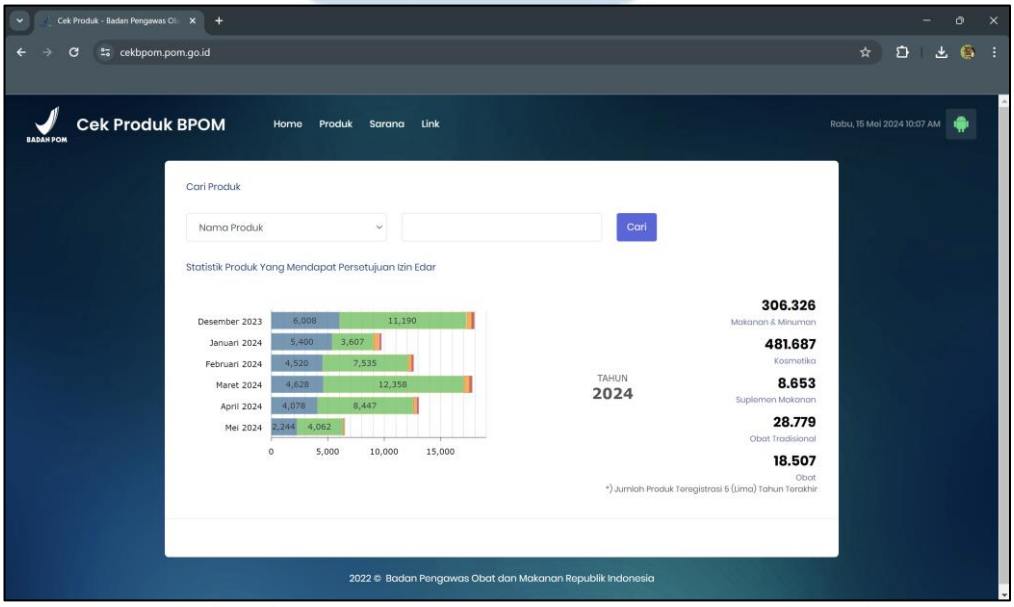
Akhir dari ketiga proyek utama yang dilakukan oleh *data engineer intern* adalah mengompres kualitas setiap file gambar yang diunduh menjadi 30% dari kualitas aslinya. Proses ini dilakukan untuk menghemat ruang penyimpanan dan mempercepat proses pengunduhan serta pengelolaan data. Untuk mengompres gambar, digunakan fungsi save(quality) dari pustaka Python Imaging Library (PIL), yang sekarang dikenal sebagai Pillow. Fungsi ini sangat berguna untuk mengecilkan ukuran file gambar tanpa mengorbankan

kualitas secara signifikan, seperti yang ditunjukkan Gambar 3.38.

```python
import os
from PIL import Image

def compress_images_in_folder(folder_path, quality):
    # Iterate through all folders and subfolders
    for root, dirs, files in os.walk(folder_path):
        # Iterate through each file in the current folder
        for file in files:
            # Check if the file is an image
            if file.endswith(('png', 'jpg', 'jpeg', 'gif')):
                try:
                    # Open the image
                    image_path = os.path.join(root, file)
                    img = Image.open(image_path)

                    # Compress the image
                    img.save(image_path, quality=quality)

                except Exception as e:
                    print(f"Error compressing {file}: {e}")

# Folder containing images
folder_path = r"D:\Kerja\recommendation_skin_analyst\28-07-2023_ProductScraping\data\Sephora_ID\product_images"

# Quality setting (0-100, where 100 is the best quality)
quality = 30  # Adjust as needed

# Compress images in all subfolders
compress_images_in_folder(folder_path, quality)
```

Gambar 3.38 Kode untuk Kompresi Gambar

3.2.5 Tugas Sampingan 1 Pencatatan Nomor BPOM Indonesia

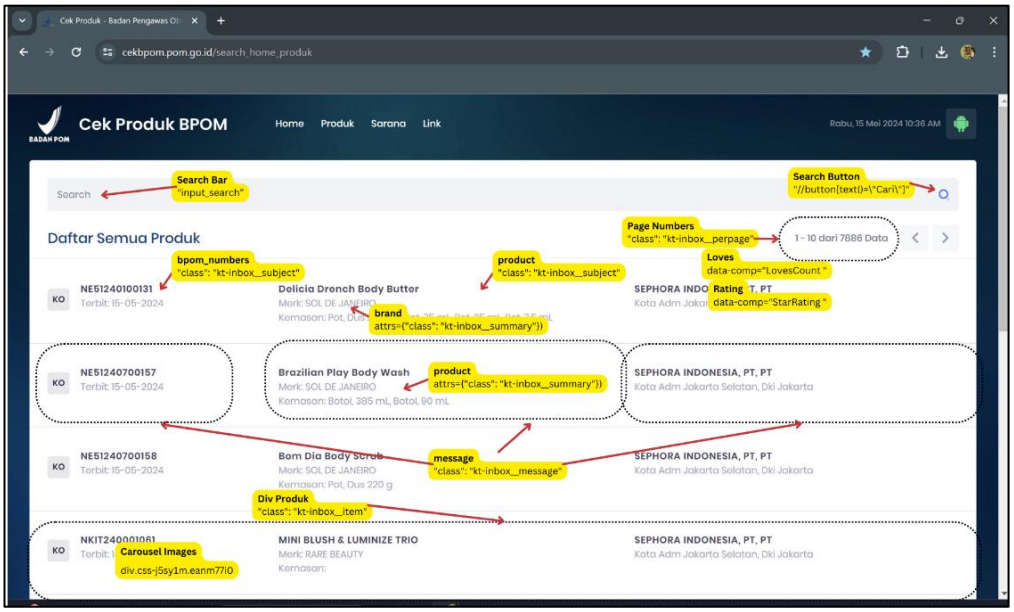


Gambar 3.39 Tampilan Halaman Utama Cek Produk BPOM

Untuk melengkapi data produk rekomendasi yang telah didapat KSH, nomor BPOM untuk produk-produk yang telah dicari pada situs web Sephora Indonesia, Sephora Global, dan Olive Young juga perlu untuk dicari.

Diperlukan metode terbaik untuk membantu pengisian data nomor BPOM produk yang sangat banyak. Web resmi Badan Pengawas Obat dan Makanan (BPOM) Indonesia sendiri dapat diakses oleh publik pada laman cekbpom.pom.go.id. Laman Cek Produk BPOM berfungsi sebagai penyimpan daftar produk-produk makanan maupun obat dan kosmetik yang telah lolos uji standar keamanan BPOM.

Tugas sampingan pertama dimulai dengan peninjauan struktur website Cek Produk BPOM, yang tampilan halaman utamanya dapat dilihat pada Gambar 3.39. Pada halaman tersebut, terdapat dropdown filter pencarian yang mencakup beberapa kriteria, yaitu Nomor Registrasi, Nama Produk, Merk, Jumlah & Kemasan, Bentuk Sediaan, Nama Pendaftar, dan NPWP Pendaftar produk. Filter ini memungkinkan pencarian produk yang lebih spesifik dan akurat untuk mendapatkan informasi nomor BPOM yang dibutuhkan. Adapun pada Gambar 3.40 merupakan struktur dari halaman web Cek Produk BPOM ketika pengguna mencari nomor-nomor BPOM produk menggunakan kata kunci Merk yang didaftarkan.



Gambar 3.40 Tampilan Struktur Halaman Hasil Pencarian pada Web Cek Produk BPOM

Penulisan kode otomatis pengisian nomor BPOM dapat dilihat pada Gambar 3.41. Pustaka-pustaka Python berupa BeautifulSoup dari bs4, Pandas, Request, Times, dan beberapa modul dari Selenium digunakan. Kode di bawah ini merupakan sebuah script yang bertujuan untuk mengambil data registrasi produk dari situs web resmi Badan Pengawas Obat dan Makanan (BPOM) Indonesia, yaitu cekbpom.pom.go.id. Langkah-langkah yang dilakukan oleh kode ini adalah sebagai berikut.

```
brand_names = []
product_names = []
bpom_numbers = []

webpage_link = "https://cekbpom.pom.go.id/"
options = webdriver.ChromeOptions()
options.add_argument('--disable-web-security')
options.add_argument('--allow-running-insecure-content')
driver = webdriver.Chrome(options=options)
driver.maximize_window()
driver.get(webpage_link)
WebDriverWait(driver, 20).until(EC.any_of(EC.presence_of_element_located((By.CSS_SELECTOR, "#input_search")),
                                          EC.visibility_of_element_located((By.CSS_SELECTOR, "#st_filter")),
                                          EC.visibility_of_element_located((By.CSS_SELECTOR, ".col-4"))))

driver.find_element(By.XPATH, "//select[@name='st_filter']/option[text()='Merk']").click()

query = driver.find_element(By.ID, "input_search")
query.send_keys("Sephora")

btn = driver.find_element(By.XPATH, "//button[text()=\"Cari\"]")
btn.click()

time.sleep(5)

html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')
page_end = soup.find('span', attrs={"class": "kt-inbox__perpage"}).get_text().split("-")[1].split("dari")[0]
total_data = soup.find('span', attrs={"class": "kt-inbox__perpage"}).get_text().split("dari")[1].split("Data")[0]

while page_end.strip() != total_data.strip():
    divs = soup.find_all('div', attrs={"class": "kt-inbox__item"})
    for div in divs:
        message = div.find_all('div', attrs={"class": "kt-inbox__message"})
        brand = message[1].find('span', attrs={"class": "kt-inbox__subject"}).get_text()
        product = message[1].find('span',
                                  attrs={"class": "kt-inbox__summary"}).get_text().split("Merk: ")[1].split("Kemasan: ")[0]
        bpom_number = message[0].find('span', attrs={"class": "kt-inbox__subject"}).get_text()
        brand_names.append(brand)
        product_names.append(product)
        bpom_numbers.append(bpom_number)
    btn_next = driver.find_element(By.XPATH, "//button[@title='Halaman Selanjutnya']")
    btn_next.click()
    time.sleep(20)
    action = webdriver.ActionChains(driver)
    action.move_by_offset(10, 20).perform()
    html = driver.page_source
    soup = BeautifulSoup(html, 'html.parser')
    print(page_end.strip() + " / " + total_data.strip())
    page_end = soup.find('span', attrs={"class": "kt-inbox__perpage"}).get_text().split("-")[1].split("dari")[0]

driver.quit()
```

Gambar 3.41 Potongan Kode untuk Scraping Data BPOM – Otomasi Webdriver

Pertama, kode membuka sebuah browser dan mengarahkannya ke halaman utama situs cekbpom.pom.go.id. Ini dilakukan menggunakan WebDriver dari Selenium, dengan menggunakan browser Mozilla Firefox. Setelah halaman dimuat, kode menunggu hingga element-element penting muncul menggunakan WebDriverWait dari Selenium. Element input pencarian

yang dimaksud adalah XPath CSS SELECTOR dari opsi filter pencarian, dan daftar produk ("#input_search", "#st_filter", dan ".col-4". Selanjutnya, kode mengatur filter pencarian untuk mencari produk berdasarkan nama mereknya menggunakan XPath "//select[@name='st_filter']/option[text()='Merk]".

Kemudian, dibuat kode mengirimkan query pencarian berupa nama Merk perusahaan ke element input dengan mengisi element ID "input_search". Pengguna juga dapat mengubah filter yang diinginkan sesuai kebuuthan. Setelah itu, Tombol Cari kemudian ditekan untuk memuat daftar sepuluh produk pertama, yaitu element "//button[text()=\"Cari\"]". Setelah melakukan pencarian, kode menunggu beberapa detik untuk memastikan halaman telah dimuat sepenuhnya dengan menggunakan fungsi time.sleep selama 5 detik.

Program akan mengambil sumber halaman web dan melakukan parsing HTML menggunakan BeautifulSoup. Kemudian, kode menemukan informasi tentang jumlah halaman dan total data yang tersedia dari beberapa class berikut. Variabel page_end atau jumlah data terakhir dari halaman web diambil dari "class": "kt-inbox__perpage" di sebelah kiri kata “dari”. Sedangkan total_data diambil dari class yang sama di sebelah kanan kata “dari” dan di sebelah kiri kata “Data”. Jika mengambil contoh seperti pada Gambar 3.40, maka page_end diisi “10” dan total_data diisi “7886”.

Selama halaman belum mencapai akhir, kode mengumpulkan data produk dari setiap halaman. Termasuk nama merek, nama produk, dan nomor registrasi BPOM dengan element yang juga dicantumkan pada Gambar 3.40. Nama produk itu sendiri terdiri dari gabungan span.class="kt-inbox__subject" dan span. class="kt-inbox__summary" di dalam kolom kedua div message. Sedangkan nomor BPOM dari produk berada pada kolom pertama div message.

Program pun melakukan klik tombol "//button[@title='Halaman Selanjutnya']" untuk beralih ke halaman berikutnya apabila data seluruh baris produk telah didapat. Kode action.move_by_offset(10, 20).perform() digunakan digunakan untuk menggerakkan kursor *mouse* sejauh tertentu dari

posisi saat ini di layar sejauh 10 pixel ke kanan dan 20 ke bawah atau menjauh dari tombol tersebut agar tidak ada tambahan element saat *mouse* meng-*hover*.

Data yang sudah dikumpulkan disimpan dalam dataframe dengan kolom berupa brand_names, product_names, dan bpom_numbers seperti pada Gambar 3.42. Kode akan terus mengulangi proses pengambilan data hingga semua halaman telah dicatat ke dalam *dataframe*. Setelah selesai mencatat semua data yang ada, data akan disimpan kode ke dalam file CSV yang akan digunakan oleh sistem rekomendasi aplikasi UNNIS.

```
divs = soup.find_all('div', attrs={"class": "kt-inbox__item"})
for div in divs:
    message = div.find_all('div', attrs={"class": "kt-inbox__message"})
    product = message[1].find('span', attrs={"class": "kt-inbox__subject"}).get_text()
    brand = message[1].find('span', attrs={"class": "kt-inbox__summary"}).get_text().split("Merk: ")[1].split("Kemasan: ")[0]
    bpom_number = message[0].find('span', attrs={"class": "kt-inbox__subject"}).get_text()
    brand_names.append(brand)
    product_names.append(product)
    bpom_numbers.append(bpom_number)
print(page_end.strip() + " / " + total_data.strip())

bpom_xlsx = pd.DataFrame({'brand_name' : brand_names,
                          'product_name' : product_names,
                          'bpom_number' : bpom_numbers },
                          columns=['brand_name','product_name', 'bpom_number'])
bpom_xlsx.to_csv(#File No BPOM, index=False)
```

Gambar 3.42 Potongan Kode untuk Menyimpan Seluruh Nilai Element yang Didapat

Untuk menyocokkan nomor BPOM yang telah didapat seperti pada sebagian data yang tertera pada Tabel 3.4 Tabel Contoh Hasil Nomor BPOM yang Dikumpulkan Program, dibutuhkan program untuk mencari nama produk yang ada di metadata informasi produk dengan nama produk dari website BPOM. Setelah melakukan banyak percobaan, ditemukan bahwa nama produk di website resmi dan format penulisannya di website BPOM sangat tidak konsisten. Sephora.co.id menjadi web yang paling banyak menampilkan nama produk yang juga namanya telah didaftarkan ke BPOM, meskipun tidak seratus persen sama persis dengan data di website BPOM akibat penulisan nama produk dapat ditolak oleh BPOM dengan berbagai macam alasan.

Hingga waktu penulisan laporan magang ini belum ditemukan metode terbaik untuk pencocokkan data produk dengan nomor BPOM-nya jika ada. Namun program otomasi pengumpulan nomor BPOM ini telah membantu Regulator perusahaan karena sebelumnya dilakukan secara manual dan memakan waktu kerja yang lama. Dengan menggunakan program ini Regulator

sebagai pengguna program dapat melakukan kegiatan lain selagi mengumpulkan data BPOM, karena program dilakukan menggunakan *device* komputer yang sedang tidak terpakai.

Tabel 3.4 Tabel Contoh Hasil Nomor BPOM yang Dikumpulkan Program

Product_Name	Brand_Name	Bpom_Number
nars	Kiss The Starts Matte Lip Duo	NKIT220001027
nars	Kiss The Starts Matte Lip Duo	NKIT220001028
nars	O Rising Eyeshadow Palette (Limited Edition)	NE11221200022
nars	Four Play Blush Quad Palette (Limited Edition)	NC24221200226
nars	O Thrills Lip & Cheek Set (Limited Edition)	NKIT220001413
nars	Mini O Blush & Lip Duo	NKIT220001528
nars	Climax Liquid Eyeliner	NA22231200214
nars	Mini Blush	-
nars	Mini Blush	-
nars	Yachiyo Brush	-
nars	Precision Lip Liner	NC16191305035
nars	Precision Lip Liner	NC16191305013
nars	Precision Lip Liner	NC16191305008
nars	Precision Lip Liner	NC16191305022
nars	Precision Lip Liner	NC16191305011
nars	Natural Radiant	NA26190305116

	Longwear Cushion Foundation (Refills)	
nars	Natural Radiant Longwear Cushion Foundation (Refills)	NA26190305119
nars	Natural Radiant Longwear Cushion Foundation (Refills)	NA26190305120
nars	Natural Radiant Longwear Cushion Foundation (Refills)	NA26190305115
nars	Aqua Glow Cushion Foundation Case	-
nars	Climax Extreme Mascara	NE11231200006
nars	Kiss The Starts Matte Lip Duo	NE11231200006
nars	Kiss The Starts Matte Lip Duo	NKIT220001027
nars	O Rising Eyeshadow Palette (Limited Edition)	NKIT220001028
nars	Four Play Blush Quad Palette (Limited Edition)	NE11221200022

3.2.6 Tugas Sampingan 2 Scraping Konten Video Review Produk Korea Selatan dari Youtube dan TikTok

Tugas sampingan diberikan untuk mengumpulkan video yang mengulas produk hasil produksi Korea Selatan. Video-video ini akan menambah koleksi video rekomendasi di aplikasi UNNIS. KSH membebaskan untuk

menggunakan metode kerja apapun yang dianggap lebih cepat dan mudah. Aplikasi YouTube dan TikTok menjadi target pencarian video-video yang dimaksud dengan memasukkan kata-kata kunci seperti “KBeauty review produk Bahasa Indonesia”, “*nama merek* review Bahasa Indonesia”, dan lain sebagainya. Hingga akhir waktu penulisan laporan magang, telah dikumpulkan sebanyak 444 tautan video dan nama akun *influencer* secara manual dengan mencatat tautan serta nama akun pemilik konten seperti pada Tabel 3.5 Tabel Sebagian Tautan Video Rekomendasi yang Dikumpulkan.

Tabel 3.5 Tabel Sebagian Tautan Video Rekomendasi yang Dikumpulkan

Nama Akun	Link
cewekbanget	https://youtu.be/qrioQ5Iu1dQ?si=JzuzGzDhyWx938af
Nakita Channel	https://youtu.be/dCIpNJIDu18?si=-CeRvCJT61dbs2zo
Nakita Channel	https://youtu.be/xX6_j5VGUuw?si=SHCH4GoRUgweMxDO
Tribun Shopping Official	https://youtu.be/2yTFBEZmYCg?si=RPCYIq6OSy6D8Haw
Denny Ma 马丹	https://youtu.be/WLFghNg23fM?si=rjK7tDbBynpbRNO-
by Shofura	https://youtu.be/ZXNwaJCCCdI?si=LAwHFTWS8W5mDB-o
Tribun Shopping Official	https://youtu.be/Ld0emDEMmIc?si=pG7aOrmiGe370Kd-
Beauty Bae Indonesia	https://youtube.com/shorts/TEoLPX1eY2s?si=Q1pcVsAHouOsE2KW
Summer Beauty House	https://youtu.be/QxbWXwlPeCQ?si=BLEzWsQyYTmQ4Fyt
iestri kusumah	https://youtu.be/QuWIuW1Eyik?si=sW6dEc5p_Nuyf69Q

vinna gracia	https://youtu.be/tJz4_MhzSy8?si=6gml40xkVX83g7VK
livjunkie	https://youtu.be/JS18qp5Td54?si=G9rtobpNp1vxsgoW
Cerita Cantik	https://youtu.be/EyULS0GG6JA?si=-XSvwrADPTzHccbX
Gel angelicca	https://youtu.be/MY4rUq850vo?si=ikx9DJCCRrxOsOfM
K-Beauty Academy Indonesia	https://youtu.be/fraS3PsLpyE?si=2h818S71-4QCvfSO
Alifah Ratu Saelynda	https://youtu.be/iBrPCJiC4rE?si=pt7aG_-y2foUZkdS
lucyana santoso	https://youtu.be/RWL_A3nY5tE?si=M_3brbl2Feyq-0v9
Nadya Aqilla	https://youtu.be/93JCxj9h19o?si=c0_zDVOWI-DI7BSZ
Sherly Vinenzia	https://youtu.be/nAa4Z_IoJ54?si=7ELdH3qCUEnDteu-
Gabriella Lianna	https://youtu.be/8vQP1zU2HGA?si=symNh0o9NMViOQgb
Deny Ardi	https://youtu.be/xuo5CgYt8v4?si=29x5aT6Ab9DxGVRe
_chicpick	https://www.tiktok.com/@_chicpick/video/7275792853236796714?is_from_webapp=1&sender_device=pc
@novies.id	https://www.tiktok.com/@novies.id/video/7298994250924969221?is_from_webapp=1&sender_device=pc
@pristiaandhita	https://www.tiktok.com/@pristiaandhita/video/7207306256372010266?is_from_webapp=1&sender_device=pc

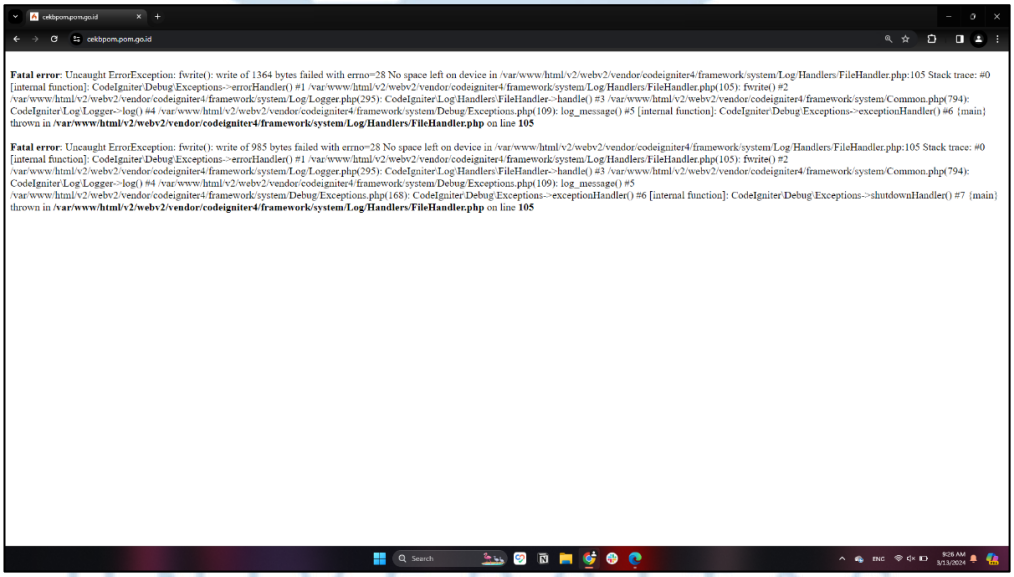
@luridaoksaviaa17	https://www.tiktok.com/@luridaoksaviaa17/video/7297519602362240262?is_from_webapp=1&sender_device=pc
@shintafalenciaa	https://www.tiktok.com/@shintafalenciaa/video/7075178075373292827?is_from_webapp=1&sender_device=pc
@melynrs_	https://www.tiktok.com/@melynrs_/video/7241955450458017030?is_from_webapp=1&sender_device=pc
@cindy.dewinta	https://www.tiktok.com/@cindy.dewinta/video/7147639029876460827?is_from_webapp=1&sender_device=pc
@nataniaimanuel.09	https://www.tiktok.com/@nataniaimanuel.09/video/7277487082141011206?is_from_webapp=1&sender_device=pc
@fanuelsabu	https://www.tiktok.com/@fanuelsabu/video/7225569678519831834?is_from_webapp=1&sender_device=pc
@aulliasha	https://www.tiktok.com/@aulliasha/video/7156795511473802523?is_from_webapp=1&sender_device=pc
@review.cicishelly	https://www.tiktok.com/@review.cicishelly/video/7286432148813352197?is_from_webapp=1&sender_device=pc
@jessicafentisa	https://www.tiktok.com/@jessicafentisa/video/7247841376329551110?is_from_webapp=1&sender_device=pc
@bianca.kartika	https://www.tiktok.com/@bianca.kartika/video/7276012563739987207?is_from_webapp=1&sender_device=pc
@shyacaa	https://www.tiktok.com/@shyacaa/video/7270100773403512069?is_from_webapp=1&sender_device=pc

3.1 Kendala yang Ditemukan

Selama proses kerja magang *data engineer* di PT Keindahan Sejahtera Hutama, ditemukan beberapa kendala yang memperlambat proses pengerjaan proyek pengumpulan data eksternal perusahaan. Kendala-kendala ini berhubungan dengan sistem *server* dari website yang menjadi taget pengerjaan sebagai berikut.

3.1.1 Server Target Website Sering Mengalami Down

Pada tanggal 13 Maret 2024, terjadi kendala halaman website Cek Produk BPOM mengalami *down* dengan tampilan seperti pada Gambar 3.43 ketika program *scraping* nomor BPOM sedang dijalankan. Setelah mencari tahu apa yang terjadi dengan menelusuri mesin pencarian Google dengan beberapa potongan kode error dari website, ditemukan bahwa server website BPOM dijalankan kehabisan ruang. Pesan "*No space left on device*" menandakan bahwa server tidak memiliki ruang kosong yang cukup untuk menulis data baru, termasuk log aplikasi.



Gambar 3.43 Tampilan Website Cek Produk BPOM Mengalami Down

3.1.2 Website Memutus Request Program Python

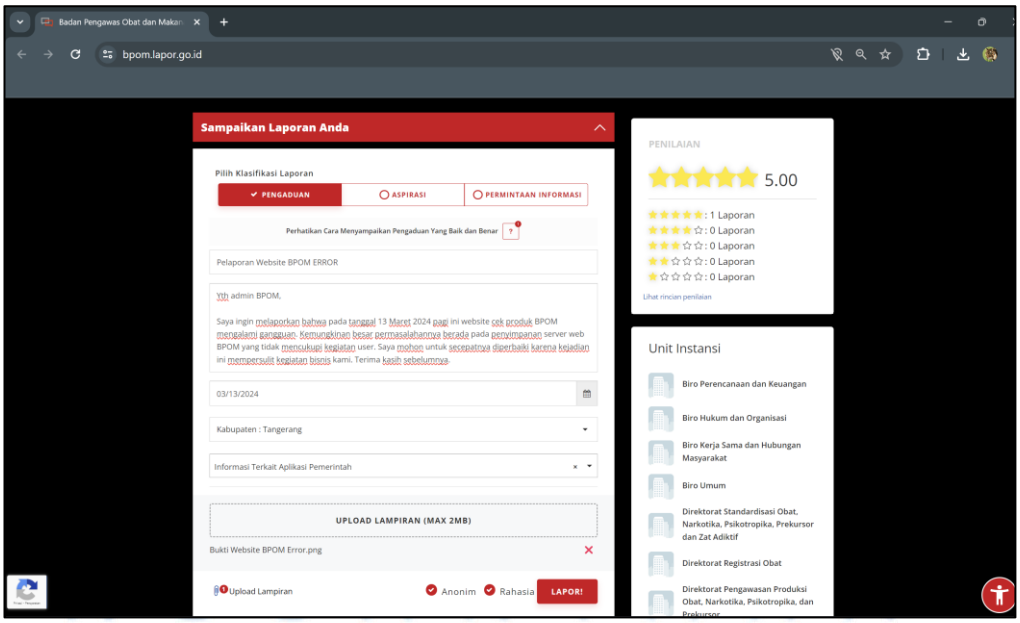
Kendala lain yang terjadi selama pengerjaaan proyek otomasi pengumpulan gambar produk ialah website Olive Young dan Sephora USA berulangkali memutus proses *request* dari program yang sedang dijalankan.

Pada saat program *scraping* dari website Olive Young, program kerap terhenti dengan pernyataan “ConnectionResetError: ('Connection aborted.', ConnectionResetError(10054, 'An existing connection was forcibly closed by the remote host', None, 10054, None))”. Sedangkan pada tahap awal pengerjaan program pengunduhan gambar produk Sephora USA, program tidak dapat dijalankan dan memberi pernyataan “ConnectionError: ('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))”.

3.2 Solusi atas Kendala yang Ditemukan

3.2.1 Berkoordinasi dengan Tim dan Melaporkan Kepada Pihak Website

Diputuskan bahwa untuk melaporkan kejadian ini ke laman resmi BPOM untuk pelaporan. Pengisian form dilakukan yang ditampilkan pada Gambar 3.44. Sehari setelah pelaporan ini, website https://cekbpom.pom.go.id/ menampilkan pesan bahwa website sedang dalam tahapan pemeliharaan. Pemeliharaan ini berlangsung selama 4 hari kerja yang berakibat ditundanya pengerjaan Tugas Sampingan *Scraping* Nomor BPOM setelah melakukan koordinasi bersama supervisor dan AI developer.



Gambar 3.44 Tampilan Halaman Form Pelaporan User Kepada BPOM

3.2.2 Penyesuaian Kode Program Python

Solusi yang telah dilakukan untuk mengatasi pemutusan *ConnectionResetError* terhadap proses *request* oleh website Olive Young ialah memberikan code function penambahan waktu tunggu (*time sleep*) di antara tugas, yaitu selama 15 detik setelah sebuah tautan dibuka oleh WebDriver. Adapun untuk mengatasi gagalnya program pengunduhan gambar Sephora USA, identitas User Agent diberikan pada bagian *headers* kode berupa "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.0.0". Solusi-solusi ini membuahkan hasil yang baik dan seluruh program *scraping* gambar produk dapat sukses dijalankan dan mencapai tujuan yang diinginkan.

Keberhasilan implementasi program otomasi pengumpulan gambar produk untuk sistem rekomendasi aplikasi UNNIS ini menunjukkan bahwa pelaksanaan tugas magang sebagai Data Engineer Intern telah berhasil dilakukan di PT Keindahan Sejahtera Hutama. Program ini tidak hanya memastikan bahwa seluruh gambar produk yang diperlukan telah terkumpul dengan baik, tetapi juga telah meningkatkan kualitas dari sistem rekomendasi UNNIS.

Selama masa magang, berbagai tantangan teknis berhasil diatasi, termasuk pengelolaan *pop-up*, deteksi varian produk, dan pengelolaan data gambar dengan metode kompresi yang efektif. Hal ini membuktikan bahwa kemampuan teknis dan keterampilan *problem-solving* yang diterapkan selama proyek sangat memadai. Selain itu, keberhasilan ini juga mencerminkan kemampuan kolaborasi dan komunikasi yang baik dengan tim divisi IT di PT Keindahan Sejahtera Hutama.

Secara keseluruhan, pengalaman magang ini tidak hanya memberikan kontribusi signifikan bagi perusahaan tetapi juga menjadi pengalaman belajar yang berharga bagi pengembangan karier sebagai *Data Engineer*. Kesuksesan proyek ini menegaskan bahwa pelaksanaan tugas magang telah memenuhi atau bahkan melampaui harapan yang ditetapkan, membuktikan kompetensi dan dedikasi dalam menjalankan peran sebagai *Data Engineer Intern.*