

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

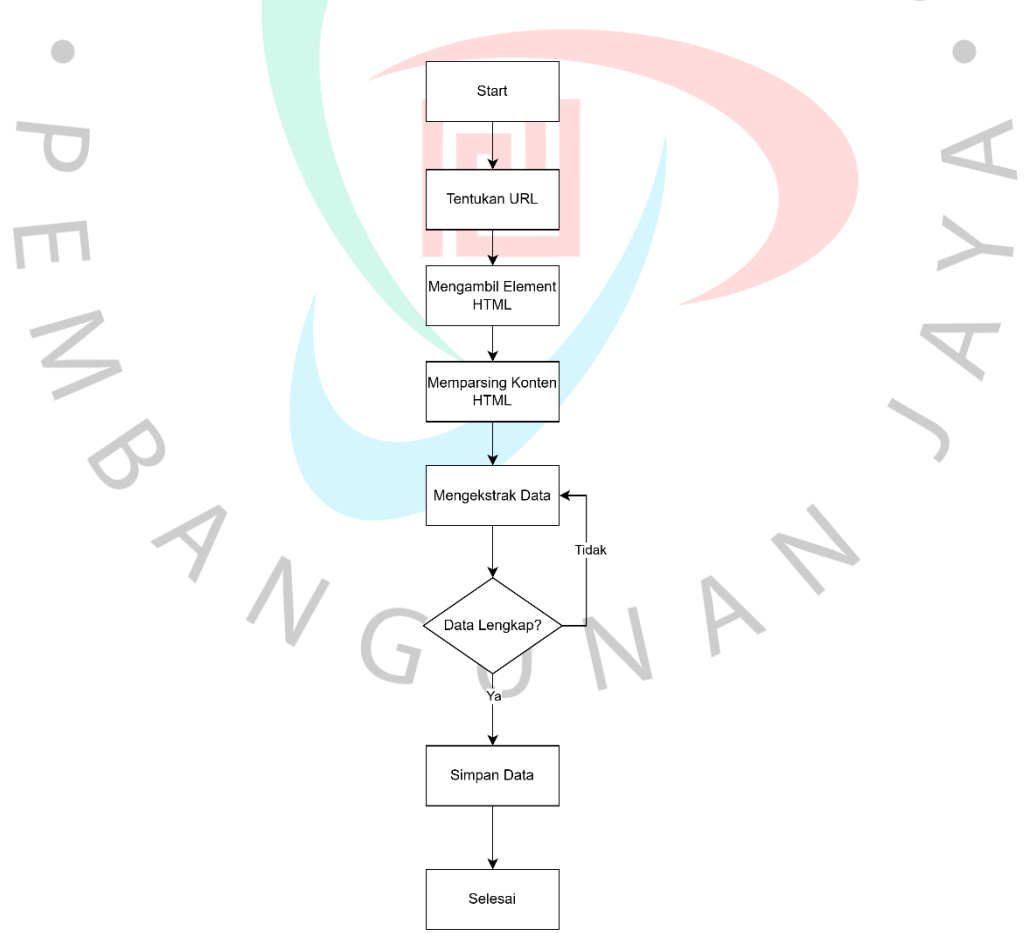
Pada pelaksanaan kegiatan kerja profesi, praktikan berada di unit *Internet Product and Traffic Management* dan praktikan memiliki tanggung jawab untuk mendukung pengembangan dan pengelolaan produk digital yang berfokus pada layanan internet. Salah satu tanggung jawab utama praktikan adalah melakukan web scraping dengan menggunakan *Selenium WebDriver*. Dalam tugas ini, praktikan mulai dengan mengidentifikasi situs web yang relevan dan penting untuk pengumpulan data. Setelah itu, praktikan merancang dan menulis skrip otomasi yang bertujuan untuk mengekstraksi data-data penting dari situs web tersebut, seperti informasi produk, harga, ulasan, dan data lainnya yang mendukung pengembangan produk. Proses scraping ini melibatkan berbagai langkah, data yang telah terkumpul kemudian disimpan dalam format yang sesuai untuk dianalisis lebih lanjut oleh tim, guna mendukung berbagai keputusan strategis dalam pengembangan produk. Dengan melakukan tugas ini, praktikan juga mempelajari berbagai aspek teknis dan tantangan dalam web scraping, yang berkontribusi terhadap pemahaman yang lebih mendalam tentang bagaimana data dapat dimanfaatkan untuk meningkatkan kualitas produk digital yang dikembangkan.

3.2 Pelaksanaan Kerja

Pelaksanaan kerja profesi telah dilakukan pada tanggal 1 Juli hingga 31 Agustus 2024 di Telkom Witel Jakarta Pusat, dengan posisi sebagai bagian dari unit *Internet Product and Traffic Management*. Selama periode ini, berbagai tugas dan tanggung jawab telah dilaksanakan, yang mencakup pengelolaan produk internet dan pemantauan lalu lintas data, serta keterlibatan dalam pengumpulan dan analisis data melalui *web scraping*. Berikut adalah Gann Chart dan Work Flow proses kerja kegiatan selama melaksanakan kerja profesi

Tabel 3. 1 Gann Chart Kegiatan

No	Kegiatan	Juli				Agustus				September			
		1	2	3	4	1	2	3	4	1	2	3	4
1.	Pengenalan profil, budaya kerja, dan unit	■											
2.	Mengerjakan tugas yang diberikan oleh unit terkait		■	■									
3.	Mengeksplor data atau sistem yang terkait dengan tugas			■	■								
4.	Mengimplementasikan tahap awal project					■	■	■					
5.	Melakukan pengujian lebih lanjut								■				
6.	Melakukan integrasi dan pengujian akhir									■			
7.	Identifikasi perbaikan dan pengujian										■		
8.	Penyusunan presentasi untuk memaparkan hasil kerja											■	
9.	Evaluasi kerja dan penyelesaian administrasi												■



Gambar 3. 1 Work Flow Kerja Profesi

3.2.1 Pengenalan Lingkungan Kerja

Telkom Indonesia adalah salah satu perusahaan telekomunikasi terbesar di Indonesia dengan sejarah panjang dalam menghadirkan berbagai layanan komunikasi dan teknologi informasi yang inovatif. Perusahaan ini memiliki visi untuk menjadi pemimpin dalam ekosistem digital di kawasan Asia Tenggara. Berbagai unit dan divisi di Telkom memiliki peran masing-masing dalam mencapai tujuan ini, salah satunya adalah unit *Internet Product and Traffic Management* yang bertanggung jawab mengelola produk internet sekaligus lalu lintas internet bagi pengguna Telkom.

Lingkungan kerja di Telkom Indonesia menawarkan budaya kerja yang dinamis, kolaboratif, dan berfokus pada inovasi. Para karyawan di sini tidak hanya didorong untuk berinovasi tetapi juga untuk bekerja dengan efisiensi, memaksimalkan penggunaan teknologi, serta menjaga komitmen pada kualitas layanan. Dengan semakin berkembangnya ekosistem digital, Telkom telah mengadopsi pendekatan modern dalam pekerjaan, seperti memungkinkan *remote working*, pengembangan keterampilan digital, dan kolaborasi lintas divisi. Hal ini membantu meningkatkan fleksibilitas, produktivitas, serta memberikan kesempatan bagi karyawan untuk terus belajar dan berkembang.

Telkom juga menekankan pentingnya mentoring dan pengembangan sumber daya manusia, dengan banyak karyawan senior yang bertindak sebagai mentor bagi karyawan baru maupun yang masih dalam tahap belajar. Bimbingan dari para mentor ini memainkan peran penting dalam memperkenalkan budaya kerja serta memberikan wawasan praktis dan teknis yang relevan dengan bidang masing-masing. Unit *Internet Product and Traffic Management* bertanggung jawab terhadap pengembangan dan pengelolaan produk internet Telkom, sekaligus mengatur lalu lintas data agar tetap stabil dan efisien. Unit ini memainkan peran penting dalam mendukung infrastruktur digital di Indonesia, terutama mengingat tingginya permintaan akan layanan internet yang cepat, stabil, dan terjangkau di berbagai wilayah di Indonesia. Tugas utama unit ini adalah memastikan produk internet Telkom selalu relevan dengan

kebutuhan pasar, memperhatikan aspek teknis maupun keamanan dalam penggunaan produk tersebut, serta memaksimalkan pengalaman pengguna. Selain itu, unit ini juga mengelola lalu lintas internet, yang berarti mereka bertanggung jawab mengatur penggunaan jaringan agar berjalan lancar dan tidak terbebani dengan permintaan yang melebihi kapasitas.

Dalam unit *Internet Product and Traffic Management*, Bapak Leo adalah salah satu mentor yang memiliki pengalaman panjang di bidang produk internet dan manajemen lalu lintas. Sebagai mentor, Pak Leo memiliki tanggung jawab besar untuk membantu membimbing dan memberikan tugas yang sesuai untuk para anggota baru yang bergabung di unit ini.

Sebagai seorang profesional di bidang *product development*, Pak Leo sering membagikan wawasan mendalam tentang bagaimana pengembangan produk harus dimulai dari pemahaman terhadap kebutuhan pengguna. Pak Leo juga membahas berbagai aspek teknis, mulai dari cara melakukan riset pasar, merancang fitur, hingga melibatkan tim pengembang dalam proses pengembangan produk yang efektif. Hal ini sangat penting karena pengembangan produk di Telkom harus mampu menjawab kebutuhan pasar yang terus berkembang, seperti kebutuhan akses internet cepat dan fitur-fitur yang mendukung kerja jarak jauh, hiburan online, dan pembelajaran daring.

Bapak Leo juga membimbing para anggota dalam tugas-tugas terkait *Internet Traffic Management*. Manajemen lalu lintas internet adalah salah satu aspek yang sangat kritis, mengingat tingginya volume lalu lintas yang harus dikelola oleh Telkom setiap harinya. Tugas-tugas di bagian ini mencakup pemantauan jaringan, mengidentifikasi area yang membutuhkan pengoptimalan, serta merancang kebijakan teknis untuk menangani lonjakan lalu lintas. Pak Leo juga memberikan penjelasan mendalam tentang teknik *load balancing*, cara mencegah terjadinya bottleneck, serta mengoptimalkan jaringan agar tetap lancar bagi seluruh pengguna.

3.2.2 Pembagian dan Pemberian Tugas

Di Telkom Indonesia, unit *Internet Product and Traffic Management* memegang peran kunci dalam mengelola layanan internet agar dapat memenuhi kebutuhan pengguna yang terus berkembang. Dalam mendukung tugas ini, pengumpulan data yang akurat dan up-to-date menjadi sangat penting. Salah satu metode efektif yang digunakan untuk memperoleh data ini adalah *scraping data*, atau proses pengambilan data secara otomatis dari berbagai sumber di internet. Tugas ini memungkinkan tim untuk mengumpulkan informasi terkini dan relevan mengenai lalu lintas internet, kebutuhan pengguna, serta berbagai metrik lainnya yang membantu pengembangan produk internet. Dalam unit *Internet Product and Traffic Management*, tugas *scraping data* diberikan sesuai dengan keahlian dan peran anggota tim. Beberapa anggota tim yang memiliki kemampuan teknis dalam pemrograman dan analisis data akan ● mendapatkan tugas *scraping* ini, terutama mereka yang memiliki pemahaman dalam bahasa pemrograman seperti Python. Pemantauan jaringan secara berkala memerlukan pengumpulan data dari berbagai situs eksternal, terutama dari platform yang menyediakan informasi mengenai kecepatan jaringan, waktu *downtime*, serta metrik-metrik kinerja internet. Data yang dikumpulkan akan membantu tim teknis dalam menentukan area yang memiliki masalah koneksi, mengidentifikasi pola penggunaan jaringan di jam-jam tertentu, mengambil langkah proaktif untuk mengatasi potensi masalah jaringan. Anggota unit yang bertanggung jawab dalam tugas ini perlu memiliki kemampuan dalam *scraping data* serta pengetahuan teknis tentang jaringan internet agar dapat memahami dan menganalisis data tersebut.

3.2.3 Penginstalan Tools

1. Python

Python adalah salah satu bahasa pemrograman yang paling sering digunakan untuk *web scraping*, yaitu teknik untuk mengumpulkan data dari situs web secara otomatis. Hal ini disebabkan oleh fleksibilitas Python, dukungan pustaka yang luas, dan kemudahan penggunaannya. Dengan *web scraping*, kita dapat mengakses data yang mungkin tidak tersedia

melalui antarmuka pemrograman aplikasi (API), atau jika API tersebut memiliki keterbatasan tertentu. Python membuat proses ini lebih mudah, mulai dari pengambilan data hingga pemrosesan dan analisisnya.



Gambar 3. 2 Logo Python

Python memiliki pustaka bawaan dan eksternal yang dirancang khusus untuk berbagai kebutuhan web scraping. Misalnya, pustaka seperti *requests* memungkinkan kita untuk mengakses halaman web dengan cara mengirim permintaan HTTP dan mendapatkan respons yang berisi konten HTML. Konten ini kemudian dapat diproses menggunakan pustaka seperti *BeautifulSoup*, yang memudahkan navigasi dan manipulasi dokumen HTML atau XML. Dengan *BeautifulSoup*, data yang kompleks dalam struktur HTML dapat dipecah menjadi potongan-potongan yang lebih mudah diakses, seperti teks di dalam tag tertentu atau atribut dari elemen-elemen tertentu.

2. Selenium



Gambar 3. 3 Logo Selenium

Selenium adalah library yang memungkinkan otomatisasi interaksi dengan browser. Library ini sering digunakan untuk web scraping terutama pada situs web yang menggunakan JavaScript untuk menampilkan data. Karena selenium dapat mengontrol browser secara langsung, selenium bisa mengatasi keterbatasan yang biasanya ada pada permintaan HTTP biasa (seperti yang dilakukan dengan library Requests). Selenium memungkinkan kita untuk mengekstrak data dari elemen-elemen tertentu di halaman web dengan cara yang sangat fleksibel. Misalnya, kita bisa memilih elemen berdasarkan ID, nama, kelas, atau bahkan jalur XPath. Hal ini memberikan kontrol yang lebih baik dibandingkan dengan pustaka lain, terutama ketika struktur HTML situs web rumit atau berubah-ubah.

Dalam kode ini, Selenium digunakan untuk mengakses halaman web dengan perintah `driver.get(url)`, yang akan membuka halaman web di browser dan memungkinkan seluruh konten halaman termuat, termasuk elemen yang dimuat secara dinamis. Selenium dapat diinstal melalui pip yaitu *pip install selenium*

3. *Webdriver Manager for Chrome*

Webdriver Manager for Chrome adalah pustaka Python yang digunakan untuk secara otomatis mengelola driver browser Chrome, yang

diperlukan oleh Selenium agar dapat menjalankan browser Chrome dalam proses web scraping atau otomatisasi tugas di web.

Selenium membutuhkan *driver browser* untuk berkomunikasi dengan browser tertentu, seperti Chrome, Firefox, atau Edge. Untuk Chrome, driver yang digunakan adalah *ChromeDriver*. Namun, tanpa Webdriver Manager, perlu secara manual mengunduh versi ChromeDriver yang sesuai dengan versi browser Chrome yang terinstal di komputer. Hal ini seringkali merepotkan karena setiap kali Chrome diperbarui, kita harus memastikan bahwa versi ChromeDriver juga diperbarui agar kompatibel.

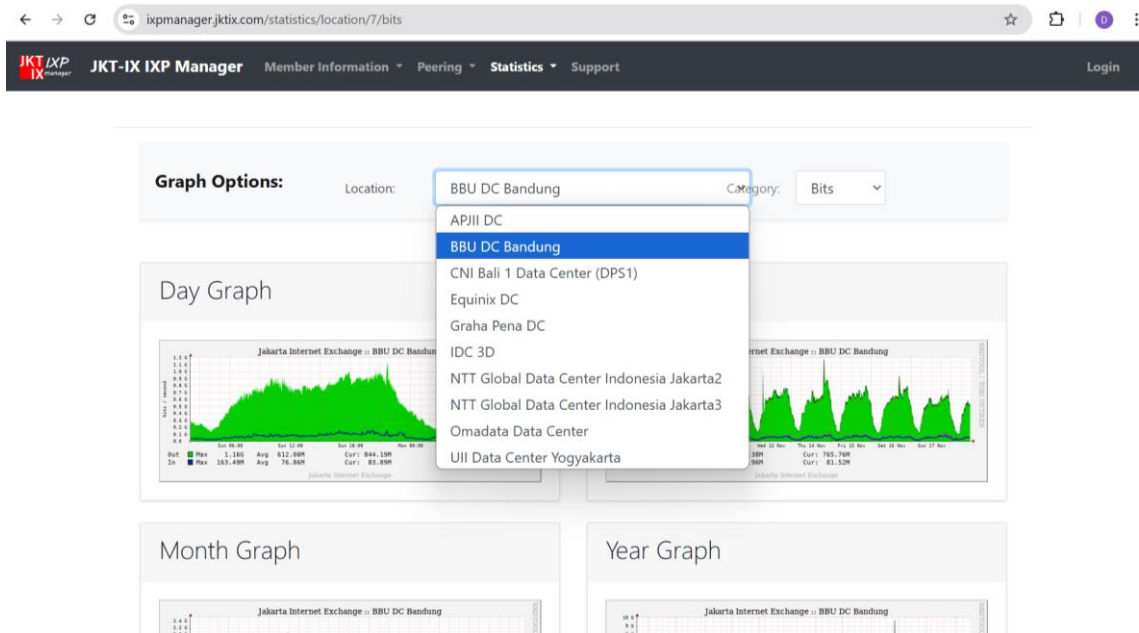
Dalam kode, webdriver-manager digunakan untuk mengatur driver Chrome secara otomatis. `Service(ChromeDriverManager().install())` akan memastikan versi ChromeDriver yang tepat diunduh dan digunakan. Install webdriver-manager dengan perintah berikut pip install webdriver-manager

3.2.4 Tampilan Website JKT IX

Web yang akan di scrap adalah JKT IX manager <https://ixpmanager.iktix.com/> yaitu adalah URL yang merujuk pada antarmuka atau platform yang berfungsi sebagai "IXP Manager" untuk JKTIX. Berdasarkan nama dan URL tersebut, IXP Manager ini kemungkinan merupakan sistem yang digunakan untuk mengelola Internet Exchange Point (IXP), tempat berbagai penyedia layanan internet (ISP) atau perusahaan yang memiliki jaringan sendiri bertemu dan bertukar lalu lintas data.

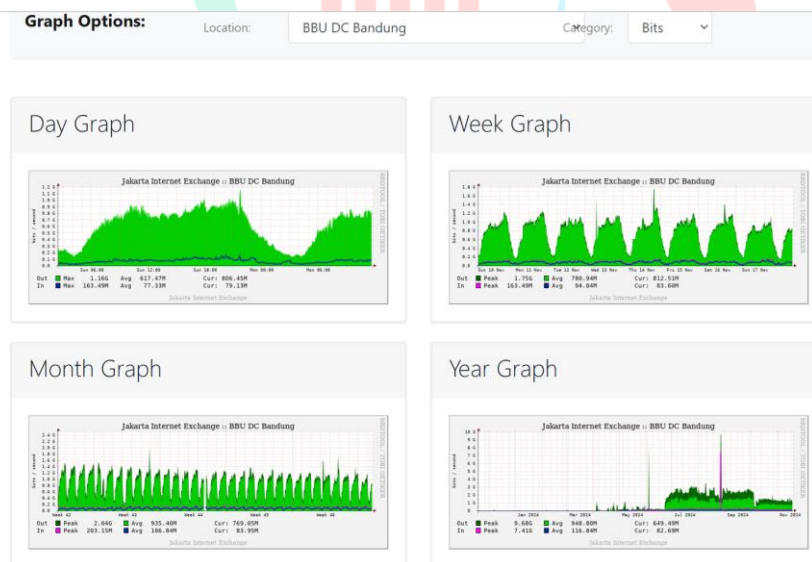
Secara umum, IXP Manager adalah perangkat lunak open-source yang banyak digunakan oleh IXPs untuk memfasilitasi pengelolaan, konfigurasi, dan monitoring dari jaringan pertukaran data mereka. Sistem ini membantu mengelola anggota IXP, sesi peering, statistik lalu lintas, dan banyak lagi. Platform semacam ini biasanya dirancang agar pengguna (misalnya ISP atau perusahaan anggota) dapat mengakses informasi teknis dan administrasi mengenai pertukaran data di jaringan tersebut.

Berikut adalah tampilan web JKT IX yang akan di scrap yang dimana terdapat beberapa lokasi internet di beberapa wilayah



Gambar 3. 4 Tampilan Web JKT IX

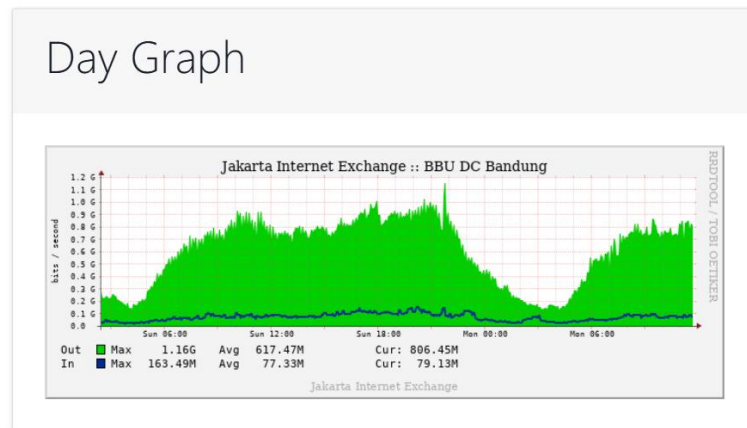
Setelah dipilih lokasinya bisa memilih category bits atau packets berikut adalah contoh kategori bits di lokasi BBU DC Bandung.



Gambar 3. 5 Tampilan Web Kategori Bits

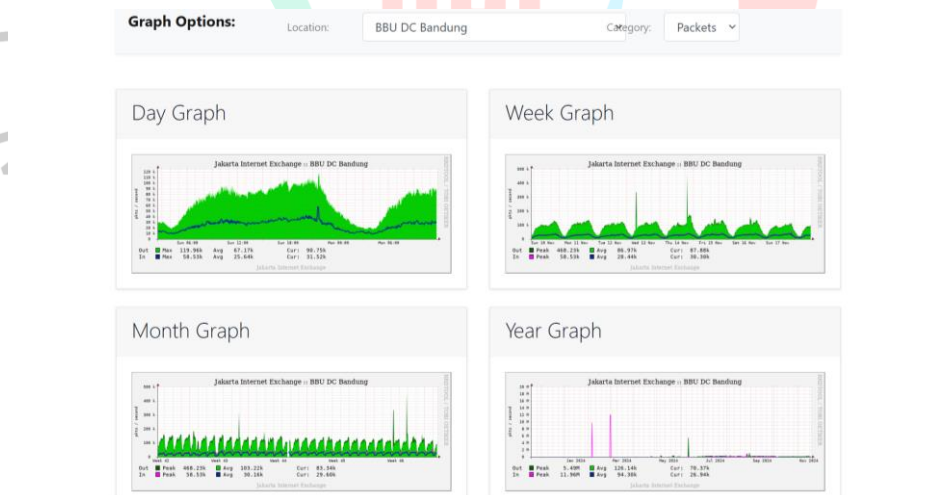
Ketika kategori grafik disetel ke bits, grafik menunjukkan jumlah data yang ditransfer dalam satuan bit per detik (bps). Ini menggambarkan volume total data yang mengalir melalui jaringan untuk memantau bandwidth yang digunakan dalam jaringan, karena bandwidth biasanya diukur dalam satuan bit (seperti Mbps atau Gbps). Grafik bits digunakan

untuk memantau seberapa besar kapasitas jaringan yang digunakan pada suatu waktu tertentu. Semakin tinggi nilai bits per detik, semakin banyak data yang melewati jaringan. Misalnya, grafik ini menunjukkan seberapa banyak data dalam bentuk bitstream yang keluar (out) atau masuk (in) ke pusat data.



Gambar 3. 6 Tampilan Web Day Graph kategori Bits di BBU DC Bandung

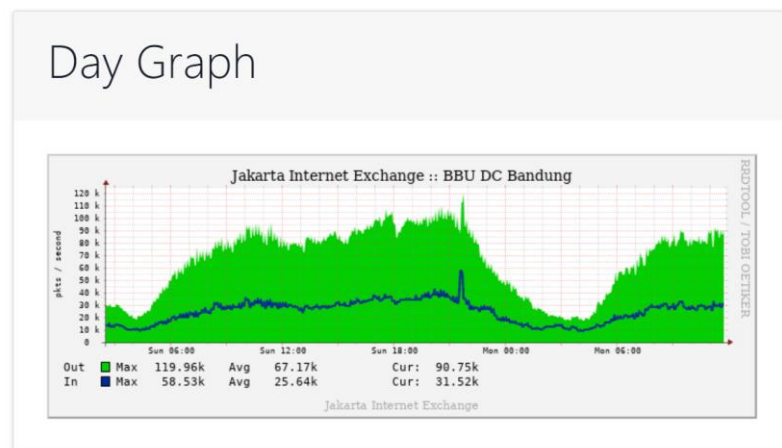
Berikut adalah contoh tampilan website di kategori packets



Gambar 3. 7 Tampilan Web Kategori Packets

Ketika kategori grafik disetel ke packets, grafik menunjukkan jumlah paket data yang ditransfer per detik (pps atau packets per second). Paket adalah unit data yang digunakan untuk mentransfer informasi dalam jaringan. Grafik packets lebih relevan untuk memantau intensitas aktivitas jaringan, yaitu berapa banyak unit data (paket) yang dikirim atau diterima

oleh jaringan. Grafik packets digunakan untuk melihat frekuensi lalu lintas, bukan volume data. Bahkan paket kecil (dengan data kecil di dalamnya) akan dihitung sama dengan paket besar dalam grafik ini. Misalnya, jika grafik packets menunjukkan lonjakan aktivitas tetapi grafik bits tetap stabil, ini bisa berarti bahwa ada banyak paket kecil yang dikirim (seperti ping atau permintaan kecil lainnya) tanpa menggunakan banyak bandwidth.



Gambar 3. 8 Tampilan Web Day Graph kategori Packets di BBU DC Bandung

3.2.5 Autodownload Gambar

Step pertama dalam autodownload gambar dari JKT IX adalah harus dilakukan adalah import library, berikut adalah beberapa import :

```
import base64
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from bs4 import BeautifulSoup
import os
import shutil
import requests
import time
```

Gambar 3. 9 Import Libary Python

import base64

Modul base64 adalah modul di Python yang memungkinkan untuk melakukan *encoding* dan *decoding* terhadap data dalam format Base64. Format ini sering digunakan untuk mengonversi data biner (seperti gambar

atau file lainnya) menjadi teks yang dapat disimpan atau ditransmisikan melalui protokol teks seperti HTTP.

Contohnya, ketika mengunduh gambar, kita bisa menyimpan gambarnya dalam format Base64 untuk memudahkan proses pengiriman atau penyimpanan di sistem yang mungkin tidak mendukung file biner. Dalam konteks kode ini, modul base64 bisa saja digunakan untuk menangani data gambar atau file lain dengan format teks aman.

from selenium import webdriver

Selenium adalah pustaka di Python untuk otomatisasi (browser automation). Di selenium bisa mengimpor webdriver dari pustaka selenium, yang merupakan bagian inti untuk mengendalikan peramban web secara otomatis. Dengan webdriver, bisa membuka halaman web, mengklik tombol, mengisi form, dan bahkan mengunduh konten dari halaman web.

Menggunakan Selenium webdriver sangat berguna ketika perlu mengakses halaman web yang memerlukan interaksi, misalnya login atau menekan tombol untuk menampilkan gambar. Jadi, pustaka ini memungkinkan kita untuk mengotomatisasi proses yang biasa dilakukan secara manual di browser.

from selenium.webdriver.chrome.service import Service

Bagian Service dari Selenium membantu mengelola layanan yang menjalankan browser secara lebih efisien. Service berfungsi untuk menyiapkan layanan atau proses yang mengontrol ChromeDriver, yang merupakan driver khusus untuk browser Chrome.

Service berguna karena memungkinkan untuk mengonfigurasi dan mengelola bagaimana ChromeDriver bekerja di latar belakang. Sehingga bisa mengatur lokasi driver, menjalankan pengaturan khusus, atau mengendalikan driver secara lebih fleksibel dibandingkan dengan langsung menjalankan ChromeDriver. Dalam konteks otomatisasi web, ini sangat membantu untuk stabilitas dan kontrol yang lebih baik.

```
from webdriver_manager.chrome import ChromeDriverManager
```

ChromeDriverManager dari webdriver_manager digunakan untuk otomatisasi proses pengunduhan ChromeDriver terbaru. Biasanya, perlu mendownload dan mengatur ChromeDriver secara manual untuk mengontrol Chrome melalui Selenium. Namun, ChromeDriverManager memungkinkan untuk mengelola ini secara otomatis.

Dengan mengimpor ChromeDriverManager, bisa memastikan bahwa selalu memiliki versi terbaru dari ChromeDriver, sesuai dengan versi Chrome di komputer. Ini mencegah masalah kompatibilitas antara Selenium dan Chrome yang bisa menyebabkan error dalam menjalankan otomatisasi.

```
from bs4 import BeautifulSoup
```

- BeautifulSoup adalah pustaka di Python yang mempermudah proses mem-parsing (menganalisis) dan memanipulasi HTML. Ketika mengambil halaman web, HTML-nya akan berupa teks biasa yang sulit untuk dibaca dan diambil informasinya. BeautifulSoup mengubah HTML menjadi struktur yang lebih mudah diakses sehingga kita bisa mengambil data yang diinginkan.

Misalnya, jika ingin mengunduh gambar dari halaman web, BeautifulSoup memudahkan kita untuk mencari semua elemen gambar () dalam halaman HTML dan mengakses URL gambar-gambar tersebut. Tanpa BeautifulSoup, mengambil data dari HTML bisa jadi lebih sulit dan memakan waktu.

```
import os
```

Modul os adalah modul bawaan di Python yang digunakan untuk berinteraksi dengan sistem operasi. Modul ini menyediakan berbagai fungsi yang membantu dalam bekerja dengan sistem file dan lingkungan (*environment*).

Dalam konteks pengunduhan gambar, modul os bisa digunakan untuk membuat folder baru di komputer untuk menyimpan gambar,

memeriksa apakah folder atau file sudah ada, menghapus file, dan mengatur struktur penyimpanan file. Modul ini sangat membantu ketika kita ingin membuat sistem pengunduhan yang rapi dan terorganisir.

import shutil

Shutil adalah pustaka lain yang berguna untuk operasi file di Python. Pustaka ini memungkinkan melakukan operasi tingkat tinggi seperti menyalin, memindahkan, atau menghapus file dan direktori dengan mudah.

Dalam proyek pengunduhan gambar, shutil bisa membantu jika ingin memindahkan file ke lokasi tertentu setelah diunduh, atau menghapus file sementara yang tidak lagi diperlukan. Contohnya, bisa memindahkan file yang telah diunduh ke folder terpisah berdasarkan kategori atau jenis gambar, atau menghapus gambar yang duplikat.

import requests

Pustaka requests adalah pustaka pihak ketiga yang sangat populer untuk melakukan permintaan HTTP di Python. Dengan requests, bisa mengirimkan permintaan GET atau POST ke URL dan mengambil data yang dikembalikan oleh server, seperti konten HTML atau file.

Dalam konteks ini, requests digunakan untuk mengunduh gambar langsung dari URL. Ketika menemukan URL gambar di halaman web, bisa menggunakan requests untuk mengirimkan permintaan ke URL tersebut dan mendapatkan konten gambar. Setelah itu, konten gambar bisa disimpan sebagai file di komputer.

import time

Modul time adalah modul bawaan Python yang menyediakan berbagai fungsi yang berhubungan dengan waktu. Salah satu fungsi yang paling sering digunakan adalah `time.sleep()`, yang memungkinkan kita untuk menambahkan jeda atau menunggu beberapa detik sebelum melanjutkan ke langkah berikutnya dalam kode.

Dalam otomatisasi pengunduhan gambar, modul time bisa digunakan untuk menambahkan jeda antara pengunduhan satu gambar dan gambar berikutnya. Ini berguna untuk menghindari pembatasan dari server (agar tidak terlihat seperti bot yang melakukan banyak permintaan dalam waktu singkat) dan untuk memberi waktu agar halaman web bisa sepenuhnya dimuat sebelum kita mengambil data.

Step selanjutnya adalah menginput function dalam code, function berguna untuk menyederhanakan, mengorganisasi, dan meningkatkan efisiensi kode dengan cara mengelompokkan logika atau instruksi yang sering digunakan ke dalam satu blok yang bisa dipanggil kapan saja

Function safe_filename(url)

```
def safe_filename(url):  
    """Generate a safe directory name from a URL."""  
    return url.replace("https://", "").replace("/", "_").replace(":", "")
```

Gambar 3. 10 Function safe_filename(url)

Fungsi ini membuat nama folder yang aman dari URL yang diberikan. Karena nama folder tidak boleh memiliki karakter tertentu, fungsi ini mengganti beberapa karakter dalam URL agar aman untuk digunakan sebagai nama folder. URL yang mengandung karakter seperti https://, /, dan : akan diubah menjadi format yang aman dengan mengganti karakter-karakter tersebut dengan tanda _ atau menghapusnya.

Function download_images_from_url(url, base_dir, driver)

```
def download_images_from_url(url, base_dir, driver):  
    safe_dir_name = safe_filename(url)  
    save_dir = os.path.join(base_dir, safe_dir_name)  
    if not os.path.exists(save_dir):  
        os.makedirs(save_dir)
```

Gambar 3. 11 Function download_images_from_url(url, base_dir, driver)

Selanjutnya, fungsi download_images_from_url(url, base_dir, driver) bertujuan untuk mengambil gambar dari halaman web dan menyimpannya dalam folder yang sesuai. Pertama, fungsi ini

menggunakan `safe_filename()` untuk membuat nama folder dari URL. Setelah nama folder aman dibuat, fungsi `os.path.join()` menggabungkannya dengan `base_dir`, yang merupakan direktori utama tempat gambar akan disimpan. Jika folder ini belum ada, fungsi akan membuatnya menggunakan `os.makedirs()`.

Mengakses Halaman Web dengan Selenium

```
print(f"Processing URL: {url}")
driver.get(url)

time.sleep(5)
```

Gambar 3. 12 Mengakses Halaman Web dengan Selenium

Setelah persiapan folder selesai, fungsi menggunakan Selenium untuk membuka halaman web yang diambil dari parameter url. Dengan perintah `driver.get(url)`, Selenium mengakses halaman tersebut. Agar halaman memiliki waktu untuk memuat semua kontennya (termasuk gambar), fungsi menambahkan `time.sleep(5)`, yang memberikan waktu tunggu 5 detik. Waktu tunggu ini bisa diatur sesuai kebutuhan, terutama jika halaman membutuhkan waktu lebih lama atau lebih singkat untuk dimuat.

Mengambil Konten HTML

```
page_source = driver.page_source

soup = BeautifulSoup(page_source, 'html.parser')
div_tags = soup.find_all('div', class_='card-body')

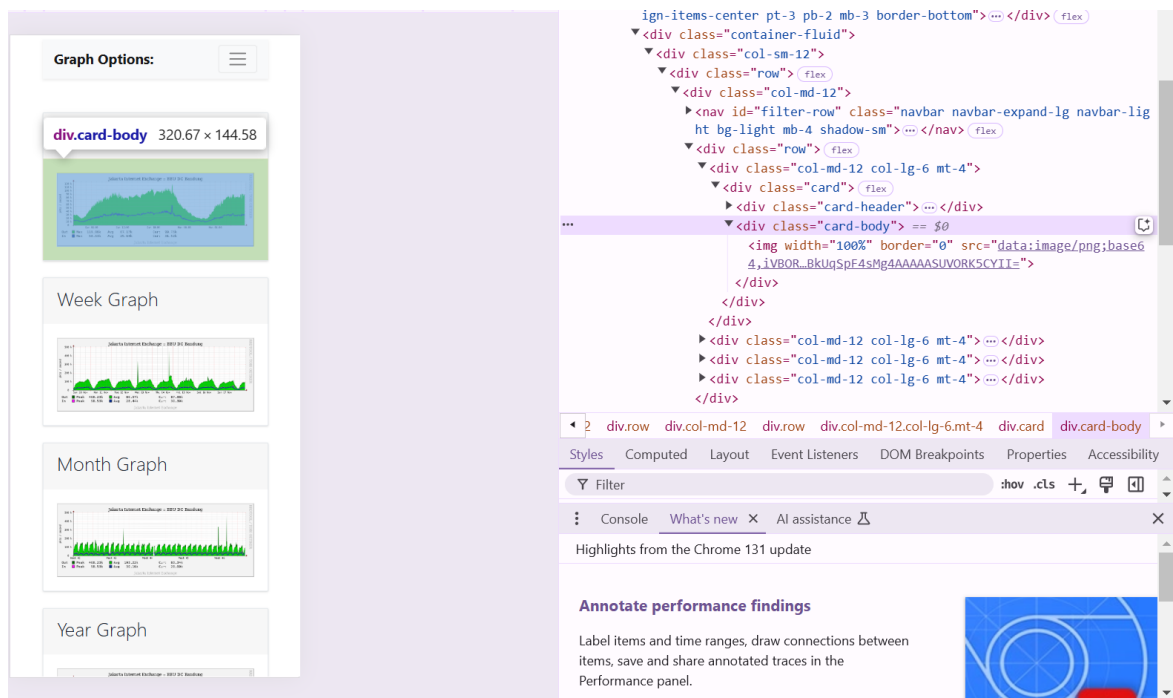
print(f"Number of div tags with class 'card-body': {len(div_tags)}")
```

Gambar 3. 13 Mengambil Konten HTML

Dalam proses pencarian gambar, fungsi BeautifulSoup `soup.find_all('div', class_='card-body')` mencari semua elemen `<div>` yang

memiliki kelas card-body. Variabel div_tags kemudian berisi daftar elemen-elemen ini. Selanjutnya, setiap elemen <div> ini diperiksa untuk menemukan elemen , yang merupakan elemen HTML untuk gambar.

Berikut adalah element html di website yang akan di scrap



Gambar 3. 14 Element HTML Web JKT IX

Element html adalah div dengan class card-body, yang di dalamnya terdapat elemen yang menampilkan gambar.

<div class="card-body"> Elemen ini adalah utama yang menggunakan kelas card-body. Yang digunakan untuk mengatur tata letak atau gaya area konten. Elemen-elemen di dalamnya biasanya adalah teks, gambar, atau data.

 Elemen gambar berada di dalam div.card-body, dengan atribut sebagai src="data:image/png;base64, Nilainya dimulai dengan data:image/png;base64,...., yang artinya gambar ini disimpan dalam format Base64.

width="100%" Atribut ini menyatakan bahwa lebar gambar akan menyesuaikan lebar elemen kontainer (dalam hal ini div.card-body)

Berikut adalah kode untuk mengambil konten html yang terdapat di web
JKT IX :

```
for i, div_tag in enumerate(div_tags):
    img_tags = div_tag.find_all('img')

    for j, img_tag in enumerate(img_tags):
        img_url = img_tag['src']
        if img_url.startswith('data:image'):

            header, encoded = img_url.split(",", 1)
            ext = header.split(";")[0].split("/")[1]
            save_path = os.path.join(save_dir, f'image_{i}_{j}.{ext}')
            with open(save_path, 'wb') as out_file:
                out_file.write(base64.b64decode(encoded))
            print(f"Base64 image saved to {save_path}")
        elif not img_url.startswith('http'):
            from urllib.parse import urljoin
            img_url = urljoin(url, img_url)

        response = requests.get(img_url, stream=True)
        if response.status_code == 200:
            save_path = os.path.join(save_dir, f'image_{i}_{j}.jpg')

            with open(save_path, 'wb') as out_file:
                response.raw.decode_content = True
                shutil.copyfileobj(response.raw, out_file)

            print(f"Image downloaded successfully and saved to {save_path}")
        else:
            print(f"Failed to download image from {img_url}")
```

Gambar 3. 15 Menginput element

```
for i, div_tag in enumerate(div_tags):  
img_tags = div_tag.find_all('img')
```

Kode ini melakukan iterasi (perulangan) pada semua elemen div_tags yang ada (kemungkinan hasil parsing dari suatu halaman HTML). Di setiap elemen div_tag, fungsi find_all('img') dipanggil untuk mencari semua tag .

```
for j, img_tag in enumerate(img_tags):  
img_url = img_tag['src']
```

Program melakukan iterasi pada setiap tag yang ditemukan di dalam sebuah div. Dari setiap tag, ia mengambil atribut src, yaitu URL gambar yang akan di-download, lalu menyimpannya ke dalam img_url.

```
if img_url.startswith('data:image'):  
header, encoded = img_url.split(",", 1)  
ext = header.split(";")[0].split("/")[1]
```

```

save_path = os.path.join(save_dir, f'image_{i}_{j}.{ext}')
with open(save_path, 'wb') as out_file:
out_file.write(base64.b64decode(encoded))
print(f"Base64 image saved to {save_path}")

```

img_url diawali dengan data:image, itu berarti gambarnya disimpan dalam format Base64 (gambar langsung disisipkan dalam HTML, bukan lewat URL). Base64 biasanya punya format seperti data:image/png;base64, ENCODED_STRING. Program memisahkan bagian sebelum, (header) dan sesudah, (data encoded). Dari header (contohnya image/png), mengambil format gambar (png, jpg, dll.). Data yang di-encode dalam Base64 didecode dengan base64.b64decode(encoded) lalu disimpan ke file di direktori save_dir.

```

elif not img_url.startswith('http'):

```

- **from urllib.parse import urljoin**

```

img_url = urljoin(url, img_url)

```

Program menggunakan fungsi urljoin dari pustaka urllib.parse untuk mengubahnya menjadi URL lengkap dengan menggabungkannya dengan URL utama halaman web.

```

response = requests.get(img_url, stream=True)

```

```

if response.status_code == 200:

```

```

save_path = os.path.join(save_dir, f'image_{i}_{j}.jpg')

```

```

with open(save_path, 'wb') as out_file:

```

```

response.raw.decode_content = True

```

```

shutil.copyfileobj(response.raw, out_file)

```

```

print(f"Image downloaded successfully and saved to {save_path}")

```

GET Request yaitu mengirim permintaan GET ke URL gambar dengan parameter stream=True, supaya data tidak langsung disimpan ke memori, tapi di-stream. Cek Status hanya download ketika server merespons dengan status kode 200 (artinya permintaan berhasil). response.raw.decode_content gambar ditulis ke file menggunakan shutil.copyfileobj.

else:

```
print(f"Failed to download image from {img_url}")
```

Kalau server tidak merespons dengan status kode 200 (contohnya, file tidak ditemukan atau server error), maka program akan mencetak pesan bahwa gambar gagal didownload.

Berikut adalah code untuk mengunduh gambar dari web JKT IX :

```
chrome_driver_path = "C:\Windows\chromedriver.exe"
service = Service(chrome_driver_path)
driver = webdriver.Chrome(service=service)

urls = [
    'https://ixpmanager.jktix.com/statistics/ixp'
]

base_dir = 'down'
if not os.path.exists(base_dir):
    os.makedirs(base_dir)

for url in urls:
    download_images_from_url(url, base_dir, driver)

driver.quit()

print("All images downloaded and saved into separate folders.")
```

Gambar 3. 16 Mengunduh Gambar

Gambar dengan URL langsung (dalam format http atau https) diunduh dengan menggunakan `requests.get()`. Jika respons dari server berhasil (status 200), gambar tersebut disimpan dalam folder yang telah dibuat. Jika ada error, misalnya jika server mengembalikan status selain 200, fungsi akan mencetak pesan error dan melanjutkan ke gambar berikutnya.

Bagian utama dari script ini melibatkan beberapa tahap tambahan. Pertama, jalur ke ChromeDriver (`chrome_driver_path`) disiapkan, yang merupakan driver yang memungkinkan Selenium mengontrol browser

Chrome. Dengan membuat instance `webdriver.Chrome()` menggunakan Service, browser Chrome akan terbuka dan siap mengakses halaman web.

Setelah itu, folder utama tempat semua gambar akan disimpan disiapkan dengan `base_dir`. Jika folder ini belum ada, `os.makedirs()` membuatnya. Script kemudian mengakses setiap URL dalam daftar urls, memanggil fungsi `download_images_from_url()` untuk mengunduh gambar dari halaman tersebut.

Terakhir, setelah semua gambar berhasil diunduh, `driver.quit()` digunakan untuk menutup browser yang sebelumnya dibuka oleh Selenium. Sebagai langkah akhir, script menampilkan pesan "All images downloaded and saved into separate folders," mengonfirmasi bahwa semua gambar berhasil diunduh dan disimpan dalam folder yang terpisah untuk setiap URL yang diakses.

3.2.6 Mengubah Gambar Menjadi Excel

Step selanjutnya, setelah gambar berhasil ke download, ubah gambar tersebut menjadi output yaitu file excel, dengan import di library python seperti :

```
import csv
import pytesseract
from PIL import Image
import pandas as pd
import os
```

Gambar 3. 17 Import Library img to csv

CSV

Modul csv digunakan untuk membaca dan menulis data dalam format file CSV (Comma Separated Values). Format CSV merupakan format teks yang umum digunakan untuk menyimpan data tabular di mana setiap baris mewakili satu entri, dan kolom-kolomnya dipisahkan dengan tanda koma atau pemisah lain.

Dengan menggunakan modul ini, user dapat dengan mudah membaca data dari file CSV, melakukan manipulasi data, dan menulis

kembali hasilnya ke dalam file CSV. Modul ini sangat bermanfaat ketika bekerja dengan data dalam bentuk tabel yang sering digunakan dalam berbagai aplikasi, seperti spreadsheet atau sistem manajemen database.

pytesseract

Pytesseract adalah sebuah wrapper Python untuk **Tesseract**, sebuah mesin OCR (Optical Character Recognition) yang dikembangkan oleh Google. Teknologi OCR ini memungkinkan pengenalan teks dalam gambar atau dokumen yang telah dipindai.

Modul ini sangat berguna untuk mengekstrak teks dari gambar yang mengandung informasi dalam bentuk teks, seperti foto kuitansi, dokumen yang dipindai, atau gambar lainnya. pytesseract dapat mengenali dan mengonversi teks yang terdapat dalam gambar menjadi format yang dapat diproses lebih lanjut. Kemampuannya dalam mengolah gambar dengan kualitas rendah menjadikannya alat yang sangat efektif dalam ekstraksi data dari gambar.

PIL(Pillow)

Library Pillow (yang merupakan penerus dari Python Imaging Library atau PIL) digunakan untuk membuka, memanipulasi, dan menyimpan berbagai jenis gambar dengan berbagai format file, seperti JPEG, PNG, GIF, dan lainnya. Dengan Pillow, pengguna dapat melakukan operasi dasar seperti mengubah ukuran gambar, memotong bagian gambar, mengubah warna dan kontras, serta mengaplikasikan berbagai filter.

Dalam konteks OCR, manipulasi gambar menggunakan Pillow sering diperlukan untuk meningkatkan kualitas gambar sebelum diproses oleh pytesseract, seperti memperbaiki kontras atau memotong bagian gambar yang tidak relevan.

pandas

Pandas adalah library Python yang digunakan untuk manipulasi dan analisis data yang terstruktur, seperti data dalam bentuk tabel. Data dalam DataFrame struktur data utama dalam pandas, memungkinkan

pengguna untuk melakukan berbagai operasi, seperti pengurutan, pemfilteran, agregasi, dan manipulasi data lainnya.

Library ini sangat berguna untuk memproses data dalam format CSV, Excel, atau SQL, dan memungkinkan analisis data secara efisien. Dalam konteks ekstraksi data dari gambar menggunakan pytesseract, pandas memungkinkan untuk menyimpan hasil ekstraksi teks dalam bentuk yang lebih terstruktur, seperti tabel, dan memudahkan pengolahan lebih lanjut.

os

Modul os menyediakan berbagai fungsi untuk berinteraksi dengan sistem operasi, seperti memeriksa apakah file atau folder ada, membuat direktori baru, menghapus file, dan mengakses path file. Modul ini sangat berguna ketika program membutuhkan manipulasi file atau pengelolaan folder, seperti ketika memeriksa apakah gambar yang akan diproses ada dalam direktori tertentu, atau ketika menyimpan hasil ekstraksi teks ke dalam file baru di direktori yang ditentukan. os juga memungkinkan untuk menjalankan perintah sistem langsung dari dalam skrip Python.

Step selanjutnya adalah menginput function dalam code, function berguna untuk menyederhanakan, mengorganisasi, dan meningkatkan efisiensi kode dengan cara mengelompokkan logika atau instruksi yang sering digunakan ke dalam satu blok yang bisa dipanggil kapan saja

Fungsi images_to_csv(image_dir, output_csv)

```
def images_to_csv(image_dir, output_csv):
    # List all image files in the directory
    image_files = [f for f in os.listdir(image_dir) if f.endswith(('.png', '.jpg', '.jpeg', '.bmp', '.gif'))]

    data = []
```

```

for image_file in image_files:
    image_path = os.path.join(image_dir, image_file)
    print(f"Processing image: {image_path}")

    # Open the image and extract text using pytesseract
    img = Image.open(image_path)
    text = pytesseract.image_to_string(img)

```

Gambar 3. 18 Fungsi `images_to_csv(image_dir, output_csv)`

Fungsi `images_to_csv(image_dir, output_csv)`

Fungsi ini bertujuan untuk memproses gambar-gambar dalam folder tertentu, mengekstrak teks yang terdapat di dalam gambar-gambar tersebut menggunakan teknologi OCR (Optical Character Recognition), dan menyimpan hasil ekstraksi dalam format CSV. Berikut adalah langkah-langkah yang dilakukan oleh fungsi ini secara lebih rinci:

```
image_files = [f for f in os.listdir(image_dir) if f.endswith(('.png', '.jpg', '.jpeg', '.bmp', '.gif'))]
```

Pada baris ini, fungsi `os.listdir(image_dir)` digunakan untuk mengambil daftar semua file yang ada di dalam folder yang ditentukan oleh parameter `image_dir`. Setelah itu, dilakukan penyaringan untuk hanya memilih file dengan ekstensi yang umum digunakan untuk gambar, seperti `.png`, `.jpg`, `.jpeg`, `.bmp`, dan `.gif`. Dengan cara ini, kita hanya akan memproses gambar-gambar yang relevan dan mengabaikan file selain gambar.

```
data = []
```

Sebelum memproses gambar, sebuah list kosong data dibuat. List ini akan digunakan untuk menyimpan hasil ekstraksi data dari setiap gambar yang akan diproses. Setiap elemen dalam list ini akan berisi dictionary yang menyimpan dua informasi penting: nama file gambar dan teks yang berhasil diekstrak dari gambar tersebut.

```
for image_file in image_files:
```

```
    image_path = os.path.join(image_dir, image_file)
```

```
    print(f"Processing image: {image_path}")
```


Di dalam loop for, program akan memproses setiap file gambar satu per satu. Setiap gambar yang ditemukan di direktori akan diproses dalam urutan yang sama dengan bagaimana mereka terdaftar di image_files. Setiap kali program mulai memproses gambar, pesan yang berisi path lengkap dari gambar tersebut akan dicetak ke layar menggunakan print(). Ini bertujuan untuk memberi tahu pengguna gambar mana yang sedang diproses saat itu.

Membuka Gambar dan Mengekstrak Teks

```
for image_file in image_files:
    image_path = os.path.join(image_dir, image_file)
    print(f"Processing image: {image_path}")

    # Open the image and extract text using pytesseract
    img = Image.open(image_path)
    text = pytesseract.image_to_string(img)

    # Preprocessing text (optional)
    text = text.replace("\n", " ") # Replace line breaks with spaces

    # Append the filename and text to the data list
    data.append({'filename': image_file, 'text': text})

# Convert the list of dictionaries to a DataFrame and save as CSV
df = pd.DataFrame(data)
df.to_csv(output_csv, index=False, encoding='utf-8', quoting=csv.QUOTE_ALL)
print(f"Text from images saved to {output_csv}")

# Path to the directory containing the images
image_dir = r'C:\xampp\htdocs\bismillah\down\ixpmanager.jktix.com_statistics_ixp'

# Output CSV file path
output_csv = r'C:\xampp\htdocs\bismillah\down\iput_text.csv'

# Convert images to CSV
images_to_csv(image_dir, output_csv)
```

Gambar 3. 19 Membuka Gambar dan Mengekstrak Teks

```
img = Image.open(image_path)
```

```
text = pytesseract.image_to_string(img)
```

Gambar yang akan diproses dibuka menggunakan library Pillow dengan perintah Image.open(image_path). Setelah gambar dibuka,

teknologi OCR yang disediakan oleh pytesseract digunakan untuk mengekstrak teks yang ada di dalam gambar tersebut. Fungsi `pytesseract.image_to_string(img)` mengonversi gambar menjadi string teks yang ada di dalam gambar tersebut. Teks ini kemudian disimpan dalam variabel `text`.

```
text = text.replace("\n", " ") # Replace line breaks with spaces
```

Teks yang diekstrak dari gambar mungkin mengandung baris baru (newline), yang bisa membuat teks menjadi tidak rapi. Oleh karena itu, dalam langkah ini, setiap karakter baris baru (`\n`) akan diganti dengan spasi. Ini dilakukan untuk memastikan bahwa teks yang diekstrak menjadi satu paragraf yang lebih mudah dibaca dan diolah.

```
data.append({'filename': image_file, 'text': text})
```

Setelah teks berhasil diekstrak dan dibersihkan, informasi ini dimasukkan ke dalam dictionary yang berisi dua kunci: `'filename'` untuk nama file gambar dan `'text'` untuk teks yang diekstrak dari gambar tersebut. Dictionary ini kemudian ditambahkan ke dalam list `data`. Dengan cara ini, setiap gambar akan memiliki entri tersendiri di dalam list yang berisi nama file dan teks yang berhasil diekstrak.

```
df = pd.DataFrame(data)
```

```
df.to_csv(output_csv,index=False,encoding='utf-8',  
quoting=csv.QUOTE_ALL)
```

```
print(f"Text from images saved to {output_csv}")
```

Setelah semua gambar diproses dan data berhasil dikumpulkan dalam list `data`, list tersebut diubah menjadi `DataFrame` menggunakan `pandas`. `DataFrame` ini memungkinkan kita untuk mengorganisir data dalam bentuk tabel yang terstruktur dengan baris dan kolom.

Selanjutnya, `DataFrame` tersebut disimpan dalam file CSV menggunakan fungsi `to_csv()`. File CSV ini akan menyimpan hasil ekstraksi teks dalam format yang bisa dibuka dengan aplikasi seperti Microsoft Excel atau

Google Sheets. File CSV akan memiliki dua kolom: satu untuk nama file gambar dan satu lagi untuk teks yang diekstrak dari gambar.

Parameter `index=False` digunakan untuk mencegah pandas menulis nomor indeks dalam file CSV, sehingga hanya data yang relevan yang disimpan. Parameter `encoding='utf-8'` memastikan bahwa teks disimpan dengan encoding yang umum digunakan dan mendukung karakter khusus, seperti karakter non-latin. `quoting=csv.QUOTE_ALL` memastikan bahwa setiap nilai dalam file CSV dibungkus dengan tanda kutip (quote), yang melindungi data yang mengandung koma atau karakter lain yang dapat mengganggu format CSV.

3.2.7 Hasil Excel

Berikut adalah hasil setelah di scrap, output excel akan otomatis tersimpan ke dalam file yang telah dibuat

filename	text
image_0	image_0 (bits / second) out In Jakarta Internet Exchange :: Jakarta Internet Exchange Bmax 2.01T Avg 1.26T cur: 1.317 Wimax 2.027 Avg 1.26T cur: 1.317
image_1	image_1 (bits / second) Out BI Peak In DI Peak Jakarta Internet Exchange :: Jakarta Internet Exchange Tur 10 Sep Wed 11 Sep Thu 12 Sep Fri 13 Sep Sat 4 Sep Sun 15 Sep 4.291 MAvg 1.317 cur: 1.387 10.0
image_2	image_2 (out In I Peak DI Peak Jakarta Internet Exchange :: Jakarta Internet Exchange Week 24 Week 25, Week 26 Week 27 6.63T MAvg 1.427 cur: 1.427 10.037 MI Avg = 1.53T cur: 1.417
image_3	image_3 (bits / second) out MI Peak In DI Peak Jakarta Internet Exchange :: Jakarta Internet Exchange Nov 2023, Sen 2624 ey 2028 ul 2824 6.871 MAvg 1.297 1.26 alesT MAvg 1.477 1.26T

Gambar 3. 20 Output Excel

Tabel ini menunjukkan ringkasan data yang masuk (In) dan keluar (Out) dalam jaringan yang diawasi oleh JKTIX pada berbagai rentang waktu, mulai dari harian (Day), mingguan (Week), bulanan (Month), hingga tahunan (Year). Setiap rentang waktu menunjukkan tiga metrik utama, yaitu:

1. Max/Peak adalah nilai puncak atau penggunaan data tertinggi yang tercatat selama periode waktu yang ditentukan. Misalnya, pada kolom "Day", Max/Peak Out adalah 443.79G, yang berarti selama sehari, jumlah data keluar maksimum mencapai 443.79 gigabyte.
2. Avg adalah nilai rata-rata data yang digunakan selama periode tersebut. Pada rentang mingguan (Week), misalnya, rata-rata data

keluar (Out) adalah 271.31G, yang menunjukkan penggunaan data keluar secara umum setiap minggu.

3. Cur adalah data yang menunjukkan nilai terkini atau terakhir yang tercatat dalam periode waktu tersebut. Misalnya, pada rentang bulanan (Month), Cur Out adalah 298.36G, yang menunjukkan penggunaan data keluar terkini pada bulan tersebut.

Berikut adalah tabel yang telah dirapikan :

Tabel 3. 2 Hasil Excel

Graph	Metric	Out	In
Day	Max/ Peak	443.79G	441.50G
	Avg	238.15G	237.49G
	Cur	327.67G	324.49G
Week	Max/ Peak	483.15G	487.34G
	Avg	271.31G	270.38G
	Cur	323.45G	322.54G
Month	Max/ Peak	494.59G	502.23G
	Avg	300.73G	300.59G
	Cur	298.36G	297.43G
Year	Max/ Peak	655.34G	659.33G
	Avg	272.65G	269.88G
	Cur	249.70G	249.09G

Out mewakili data yang keluar dari jaringan (misalnya, data yang diunggah atau dikirim ke luar jaringan). In mewakili data yang masuk ke jaringan (misalnya, data yang diunduh atau diterima dari luar jaringan).

Data ini sangat berguna bagi pengelola jaringan atau administrator IT JKTIX untuk memantau pola penggunaan bandwidth dalam jangka pendek maupun panjang. Dari data ini, keputusan strategis dapat diambil, terutama dalam hal mengetahui pola puncak dan rata-rata penggunaan data, tim IT bisa menilai apakah infrastruktur yang ada saat ini mampu menangani volume data yang dibutuhkan, terutama saat penggunaan data mencapai puncaknya. Jika penggunaan mendekati atau melebihi kapasitas

yang ada, ini bisa menjadi sinyal bahwa peningkatan kapasitas jaringan mungkin diperlukan.

Data ini bisa melihat rata-rata dan puncak penggunaan bandwidth pada interval harian, mingguan, bulanan, dan tahunan, manajemen bisa merencanakan alokasi bandwidth secara lebih efisien. Misalnya, jika terlihat adanya peningkatan signifikan pada periode tertentu, JKTIX bisa mempertimbangkan untuk menambah bandwidth pada waktu-waktu tersebut agar layanan tetap lancar.

Penggunaan data yang tinggi dalam suatu periode bisa berpotensi membebani jaringan, yang bisa menyebabkan penurunan performa atau bahkan downtime. Dengan mengetahui data ini, JKTIX dapat melakukan tindakan preventif untuk menjaga stabilitas jaringan, seperti melakukan pembatasan pada pengguna tertentu atau menambah kapasitas. Data ini ● bisa membantu dalam penganggaran untuk upgrade atau penyesuaian fasilitas jaringan. Jika data menunjukkan adanya peningkatan penggunaan data tahunan yang signifikan, manajemen bisa merencanakan investasi lebih lanjut, misalnya dengan membeli perangkat tambahan atau meningkatkan layanan penyedia internet yang lebih tinggi.

Dengan mengidentifikasi pola penggunaan data secara rinci, JKTIX bisa memahami kapan pengguna aktif menggunakan jaringan mereka. Ini penting dalam pengambilan keputusan terkait penawaran layanan atau promosi pada jam-jam sibuk, serta mengembangkan strategi manajemen lalu lintas data agar tetap optimal.

3.3 Kendala Yang Dihadapi

Berikut beberapa kendala yang dihadapi praktikan saat kerja profesi diantaranya :

1. Situs web sering diperbarui atau diubah strukturnya, seperti perubahan pada elemen HTML, URL, atau layout halaman, yang dapat menyebabkan skrip scraping yang sebelumnya berfungsi menjadi tidak bisa di *run*.
2. Situs web dapat mendeteksi pola akses yang mencurigakan dari scraper, seperti frekuensi permintaan yang tinggi, dan dapat

memblokir alamat IP yang dianggap mencurigakan. Hal ini bisa diatasi dengan menggunakan proxy, tetapi menambah kompleksitas.

3. Data yang diperlukan berada dalam bentuk gambar, scraping menjadi lebih rumit karena diperlukan teknik tambahan seperti OCR (*Optical Character Recognition*) atau pengunduhan file media.
4. Situs web mungkin mengalami downtime atau merespons dengan kode status yang tidak diharapkan (seperti 404 atau 500). Mengelola error dan exceptions ini penting agar proses scraping tidak terhenti tiba-tiba.

3.4 Cara Mengatasi Kendala

Cara mengatasi kendala praktikan saat kerja profesi diantaranya :

1. Secara rutin memeriksa dan memperbarui skrip scraping agar sesuai dengan perubahan pada struktur HTML atau layout situs
2. Jika situs web membatasi akses berdasarkan IP, gunakan rotating proxies untuk mendistribusikan permintaan dari berbagai alamat IP
3. Jika data berupa teks dalam gambar, gunakan OCR (*Optical Character Recognition*) seperti Tesseract untuk mengekstrak teks dari gambar. Ini memerlukan langkah tambahan dalam pipeline scraping.
4. Implementasikan caching untuk menghindari pengulangan permintaan yang tidak perlu dan tambahkan logika retry untuk mengatasi error sementara, seperti timeout atau kode status 500.

3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Selama praktikan mengikuti kerja profesi di unit *Internet Product and Traffic Management* di PT. Telekomunikasi Indonesia terdapat pembelajaran yang didapatkan selama perkuliahan yang diterapkan ketika sedang melakukan kerja profesi, seperti :

1. Bahasa Pemrograman

Bahasa pemrograman adalah dasar utama dalam web scraping. Dalam mata kuliah bahasa pemrograman, dipelajari struktur dasar seperti variabel, tipe data, kondisi, perulangan, dan fungsi. Keterampilan ini sangat penting saat menulis skrip untuk mengambil data dari situs web. Tanpa pemahaman tentang cara kerja kode, akan sulit untuk membuat skrip yang

dapat mengakses dan mengolah data dari web. Salah satu aspek penting dalam web scraping adalah memanipulasi data yang telah diambil, seperti mengonversi data ke format yang mudah dianalisis (misalnya CSV, JSON, atau database). Dalam mata kuliah bahasa pemrograman, dipelajari mengenai struktur data dan algoritma yang berguna untuk mengelola dan mengolah data tersebut. Pengetahuan ini sangat berguna ketika berhadapan dengan data yang diambil dari web. Dalam mata kuliah bahasa pemrograman, biasanya dipelajari library Python, seperti requests dan BeautifulSoup, yang sering digunakan dalam web scraping. Pemahaman tentang cara kerja library ini dan bagaimana mengintegrasikannya dalam skrip Python sangat penting untuk melakukan scraping dengan baik.

2. Business Intelligence

Salah satu komponen utama dalam Business Intelligence adalah pengumpulan data yang relevan untuk analisis. Web scraping dengan Python memungkinkan pengambilan data secara otomatis dari berbagai sumber di internet, seperti situs web, forum, atau platform e-commerce. Data ini dapat berupa informasi tentang produk, harga, tren pasar, atau opini pelanggan yang dapat digunakan untuk mendukung keputusan bisnis yang lebih baik. Setelah data dikumpulkan, data tersebut perlu dianalisis dan diproses agar dapat memberikan wawasan yang berarti. Mata kuliah BI mengajarkan teknik analisis data, seperti pemodelan statistik dan visualisasi, yang sangat berguna dalam mengolah data yang diperoleh melalui web scraping. Salah satu tujuan utama dari Business Intelligence adalah untuk memberikan informasi yang mendalam bagi pengambilan keputusan yang lebih tepat. Data yang diambil melalui web scraping, setelah dianalisis, dapat digunakan untuk memberikan wawasan bisnis yang lebih akurat. Misalnya, tren harga produk, analisis sentimen konsumen, atau evaluasi kompetitor dapat diperoleh untuk mendukung keputusan terkait strategi pemasaran, pengelolaan inventaris, atau pengembangan produk. Web scraping memungkinkan otomatisasi proses pengambilan data dari situs web yang dapat digunakan secara berkala. Hal ini mendukung proses BI dengan menyediakan data yang selalu terbaru tanpa perlu intervensi manual. Dengan mengotomatisasi pengumpulan

data, perusahaan dapat lebih fokus pada analisis dan pengambilan keputusan berdasarkan data yang relevan dan terkini.

3. Keamanan Informasi dan Administrasi Jaringan

Administrasi jaringan melibatkan pemantauan lalu lintas dan kesehatan jaringan untuk memastikan kinerja yang optimal dan mencegah masalah seperti serangan atau kebocoran data. Web scraping dapat digunakan untuk mengumpulkan data dari berbagai sumber yang berkaitan dengan lalu lintas jaringan, misalnya, status server, log akses, atau metrik penggunaan bandwidth yang tersedia di web atau API. Data ini kemudian dapat dianalisis untuk melihat pola atau anomali yang menunjukkan potensi masalah dalam jaringan.