LAPORAN PRAKTIKUM

PRAKTIK PEMROGRAMAN PYTHON

**PRAKTIKUM**

**WEB SCARPPING**



Disusun Oleh:

Lia Fitriani (V3923013)

**Dosen**

Yusuf Fadila Rachman. S.Kom., M.Kom

**PS D-III TEKNIK INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS SEBELAS MARET**

**2024**

1. HASIL DAN PEMBAHASAN

INPUT

1. Data Gambar



* Mendefinisikan sebuah fungsi bernama scrape\_proxyway\_images.
* Menyimpan URL halaman web yang akan di-scrape ke dalam variabel url.
* Memanggil fungsi fetch\_content(url) untuk mendapatkan konten halaman. Fungsi fetch\_content diharapkan mengembalikan isi halaman web jika permintaan berhasil atau None jika gagal.
* Memeriksa apakah konten berhasil diambil (content tidak None).
* Membuat objek BeautifulSoup dari konten HTML untuk memudahkan manipulasi dan pencarian elemen dalam HTML.Membuat list kosong images\_list yang akan digunakan untuk menyimpan URL dan atribut alt dari gambar.Menggunakan metode select dari BeautifulSoup untuk menemukan semua elemen <img> dalam halaman.
* Melakukan iterasi melalui setiap tag <img> yang ditemukan:
* Mengambil nilai atribut src dari tag <img> dan menyimpannya dalam variabel src.
* Mengambil nilai atribut alt dari tag <img> dan menyimpannya dalam variabel alt.
* Menambahkan pasangan [src, alt] ke dalam images\_list.
* Memanggil fungsi save\_to\_csv untuk menyimpan images\_list ke file CSV dengan nama data\_gambar.csv. Fungsi ini juga menulis header CSV ['src', 'alt'].
* Melakukan iterasi melalui images\_list dan mencetak setiap elemen. Setiap elemen dalam images\_list adalah list yang berisi URL gambar dan teks alt.

1. Data Subjudul



* Mendefinisikan sebuah fungsi bernama scrape\_proxyway\_subtitles.
* Menyimpan URL halaman web yang akan diambil kontennya dalam variabel url.
* Memanggil fungsi fetch\_content(url) untuk mengirim permintaan HTTP ke URL dan mengambil kontennya. Fungsi fetch\_content diharapkan mengembalikan isi halaman jika permintaan berhasil atau None jika gagal.
* Mengecek apakah content tidak bernilai None. Jika content bernilai None, berarti ada kegagalan dalam mengambil konten dari URL.
* Membuat objek BeautifulSoup dari konten HTML yang diambil sehingga dapat dengan mudah diolah dan diekstraksi.
* Menggunakan metode find\_all dari BeautifulSoup untuk menemukan semua elemen <h2> dalam halaman web. Metode find\_all mengembalikan daftar semua elemen yang cocok dengan tag yang diberikan
* Menggunakan list comprehension untuk membuat daftar subtitles yang berisi teks dari setiap elemen <h2>. Setiap teks disimpan sebagai elemen tunggal dalam sebuah list, sehingga subtitles adalah list of lists.
* Memanggil fungsi save\_to\_csv untuk menyimpan subtitles ke dalam file CSV dengan nama subjudul.csv. Fungsi ini menulis data dalam format CSV.
* Melakukan iterasi melalui subtitles dan mencetak setiap elemen. Karena setiap elemen dalam subtitles adalah list yang berisi satu elemen teks, kita mencetak elemen pertama dari list tersebut (subtitle[0]).

1. Data Keterangan

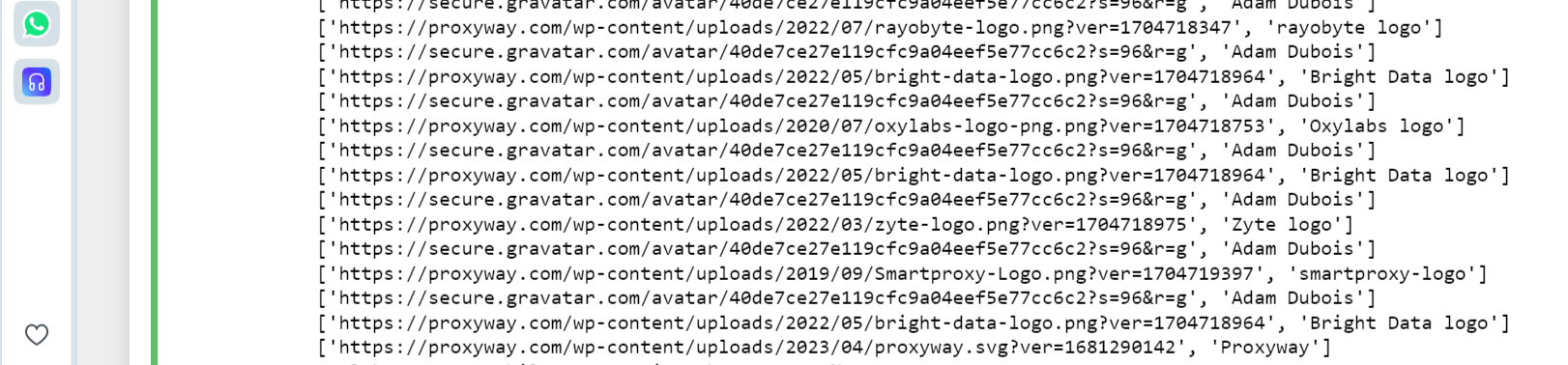


* Mendefinisikan sebuah fungsi bernama scrape\_proxyway\_descriptions.
* Menyimpan URL halaman web yang akan diambil kontennya dalam variabel url.
* Memanggil fungsi fetch\_content(url) untuk mengirim permintaan HTTP ke URL dan mengambil kontennya. Fungsi fetch\_content diharapkan mengembalikan isi halaman jika permintaan berhasil atau None jika gagal.
* Mengecek apakah content tidak bernilai None. Jika content bernilai None, berarti ada kegagalan dalam mengambil konten dari URL.
* Membuat objek BeautifulSoup dari konten HTML yang diambil sehingga dapat dengan mudah diolah dan diekstraksi.
* Menggunakan metode find\_all dari BeautifulSoup untuk menemukan semua elemen <div> yang memiliki atribut data-widget\_type dengan nilai theme-post-excerpt.default. Metode find\_all mengembalikan daftar semua elemen yang cocok dengan atribut yang diberikan
* Membuat list kosong descriptions yang akan digunakan untuk menyimpan deskripsi yang ditemukan.
* Melakukan iterasi melalui setiap elemen <div> yang ditemukan:
* inner\_div = div\_element.find('div', class\_='elementor-widget-container'): Mencari elemen <div> anak dengan kelas elementor-widget-container.
* if inner\_div:: Memeriksa apakah inner\_div ditemukan.
* text\_content = inner\_div.get\_text(strip=True): Mengambil teks dari inner\_div dan menghapus spasi kosong di awal dan akhir teks.
* descriptions.append([text\_content]): Menambahkan teks deskripsi ke dalam descriptions sebagai elemen tunggal dalam sebuah list.
* print(text\_content): Mencetak teks deskripsi ke konsol.
* Memanggil fungsi save\_to\_csv untuk menyimpan descriptions ke dalam file CSV dengan nama keterangan.csv. Fungsi ini menulis data dalam format CSV.

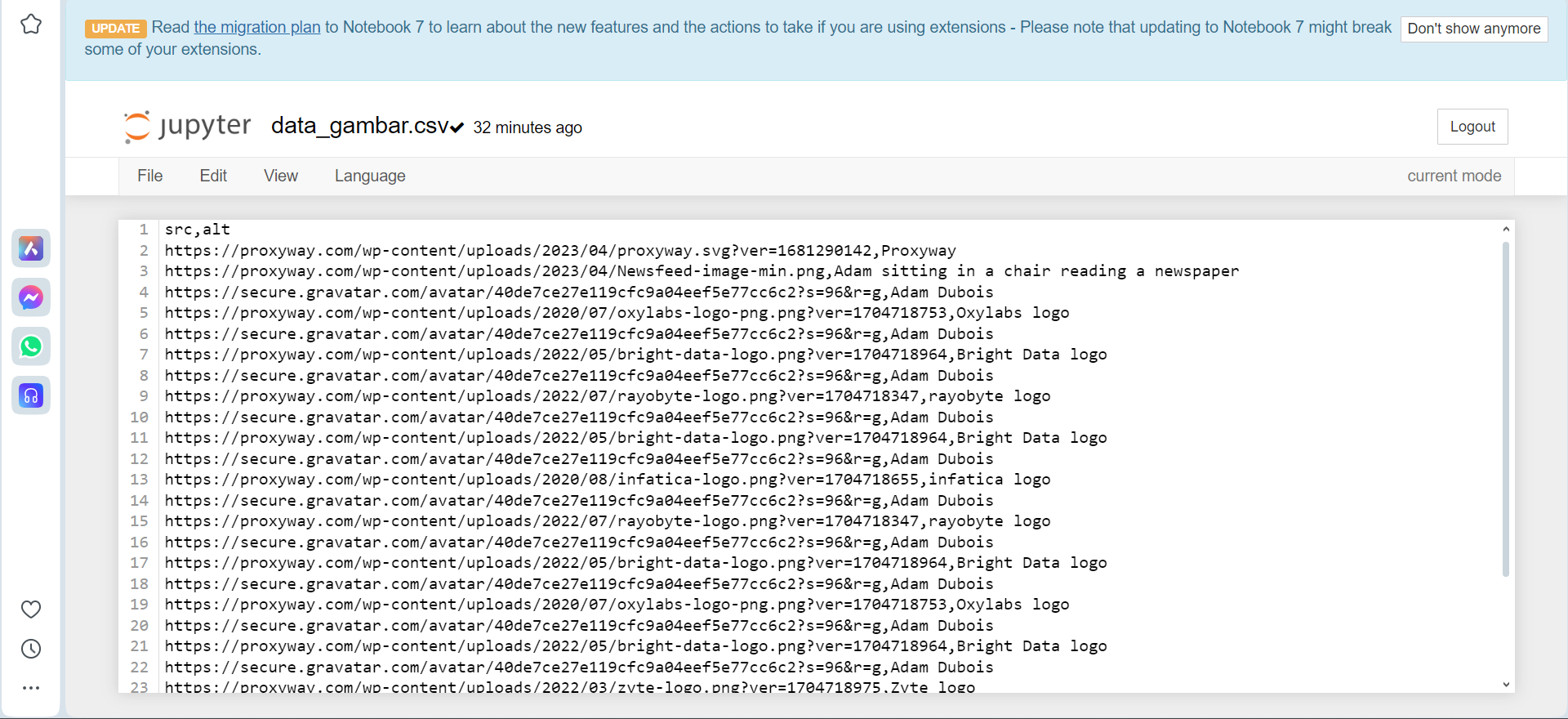
Output

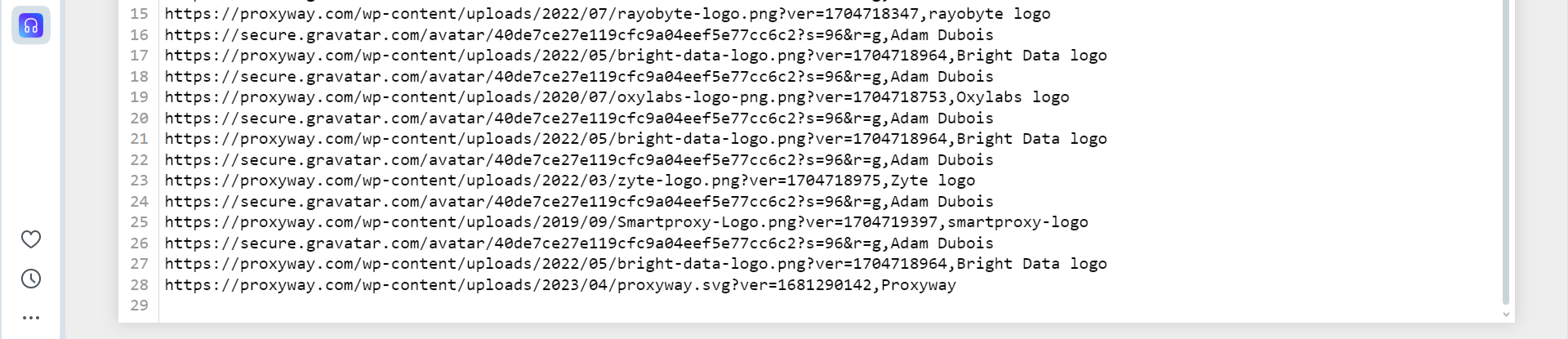
1. Data Gambar



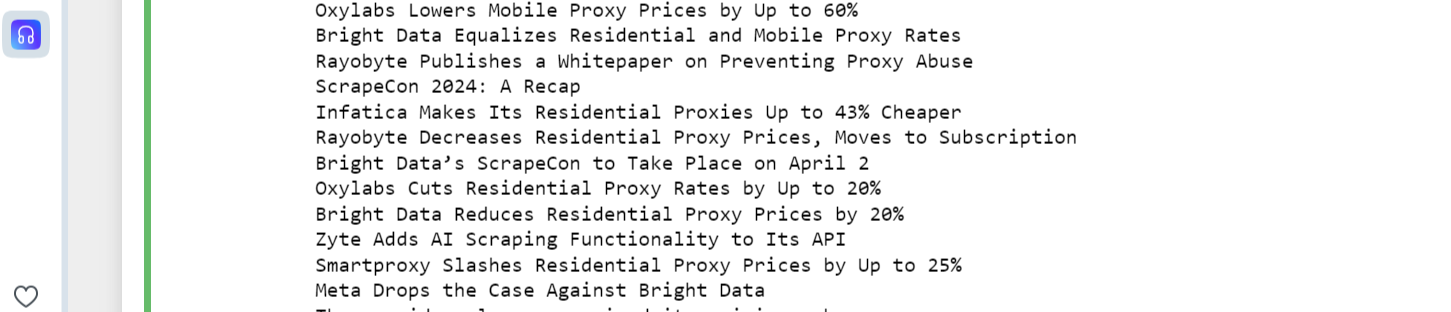


Dalam Bentuk CSV

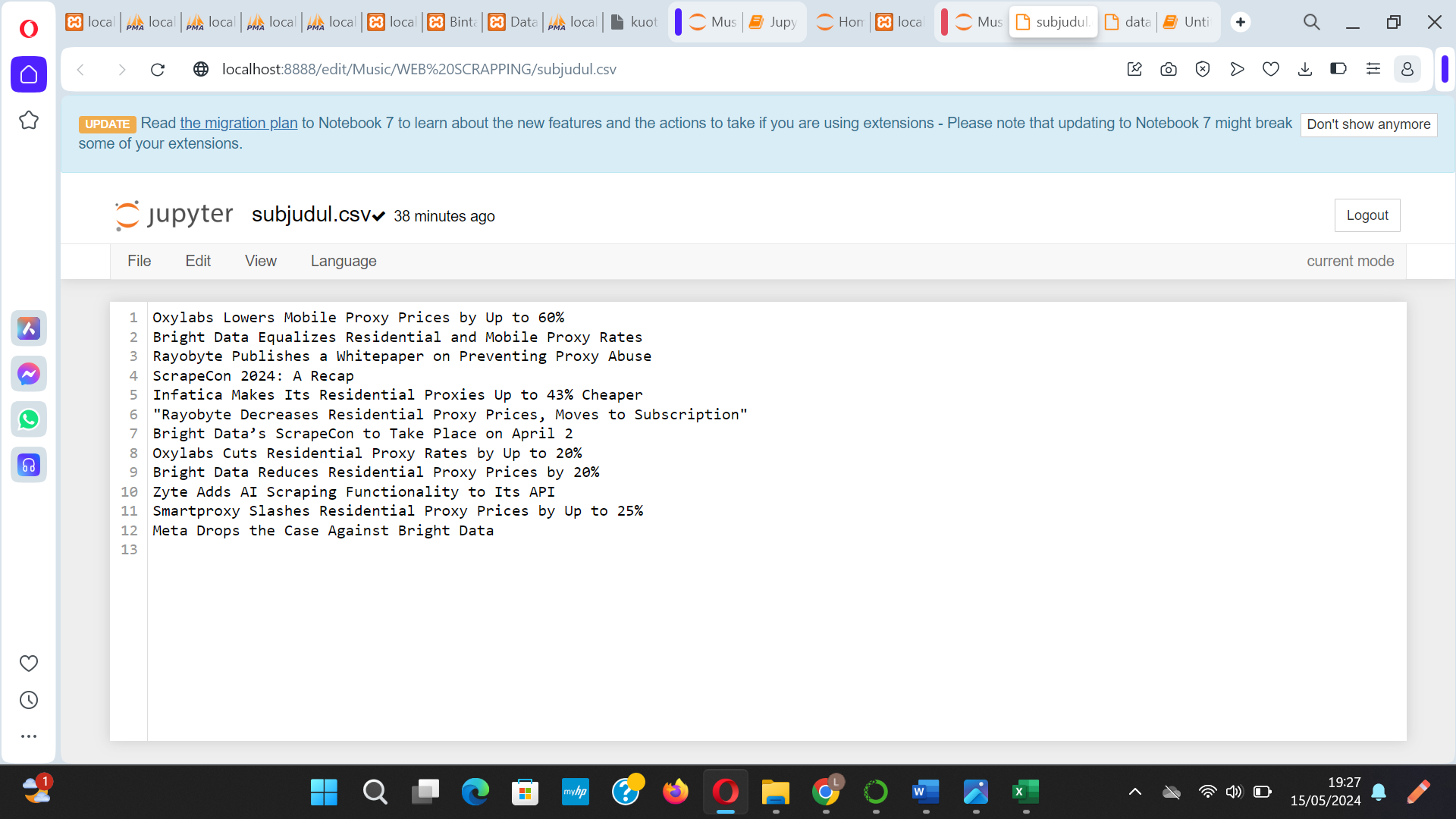




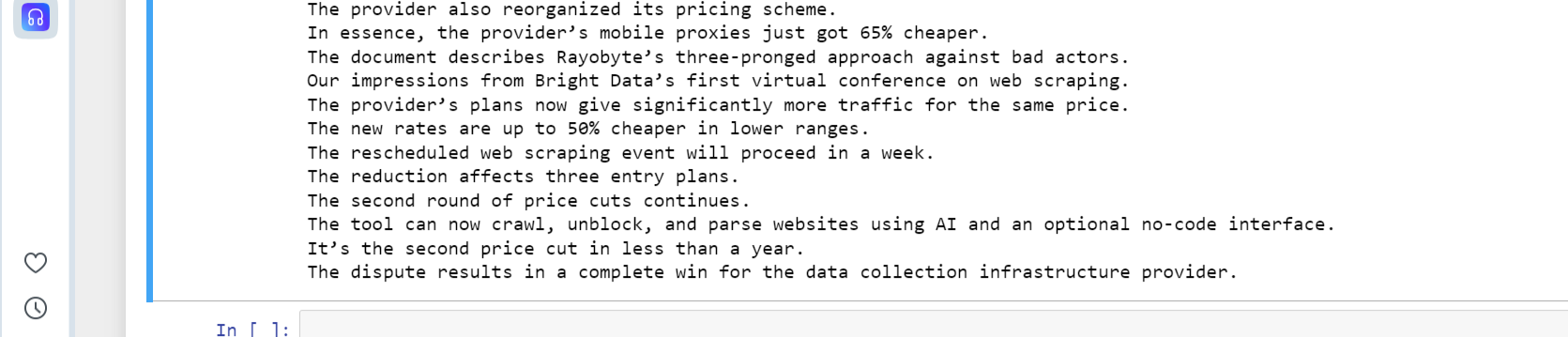
1. Data Subjudul



Dalam bentuk CSV



1. Data Keterangan



Dalam Bentuk CSV

