Laporan Data Scraping & Preprocessing: Countries from scrapethissite.com

1. Penjelasan Data

- Sumber: [Scrape This Site — Simple Countries Page](https://www.scrapethissite.com/pages/simple/)

-Fitur yang diambil:

- Name (nama negara)
- Capital (ibukota)
- Population (populasi)
- Area (luas wilayah)
- Jumlah data: 32 negara

Data ini diambil dari halaman contoh yang memang disediakan untuk latihan scraping. Situs ini digunakan karena menyediakan struktur HTML yang sederhana dan jelas, sehingga memudahkan proses ekstraksi data.

2. Proses Scraping

- Tools

- Python

- `requests` untuk mengambil HTML

- `BeautifulSoup` untuk parsing dan ekstraksi data

- Flow

1. Mengirim HTTP request ke URL target menggunakan library requests.
2. Mendapatkan source code HTML dari halaman.
3. Parse HTML menggunakan BeautifulSoup untuk menemukan tag/tag class yang sesuai.
4. Mengekstrak data nama negara, ibukota, populasi, dan area.

5. Menyimpan hasil ke dalam pandas DataFrame dan kemudian disimpan dalam format CSV agar mudah digunakan pada tahap selanjutnya, misalnya analisis atau pemodelan.

Proses scraping ini cukup straightforward karena halaman sudah statis, tidak memerlukan Selenium atau render JavaScript. Selain itu, struktur HTML yang konsisten memudahkan proses seleksi tag.

3. Proses Preprocessing

1. Tahapan

- Lowercase: Semua teks dikonversi ke huruf kecil agar seragam.

- Menghapus angka: Menghapus angka dari teks jika ada, misalnya pada nama.

- Menghapus tanda baca: Menghilangkan tanda baca supaya teks lebih bersih.

- Menghapus stopwords: Menghapus kata-kata umum dalam bahasa Inggris yang tidak memiliki makna penting (contoh: "the", "and", "of").

- Stemming: Mengubah kata ke bentuk dasarnya menggunakan Snowball Stemmer, agar kata-kata serupa bisa diproses sebagai satu bentuk (misalnya "running" menjadi "run").

2. Contoh Sebelum & Sesudah

- Sebelum

Name: United Arab Emirates

Capital: Abu Dhabi

- Sesudah

Name: united arab emirate

Capital: abu dhabi

4. Kendala

- Beberapa website tidak mengizinkan scraping, memberikan proteksi tambahan seperti CAPTCHA, atau menggunakan JavaScript untuk load data (sehingga memerlukan Selenium atau headless browser).

- Struktur HTML dapat berubah sewaktu-waktu, yang membuat selector CSS atau tag XPath yang digunakan menjadi tidak valid.

- Terkadang server memblokir request jika dianggap mencurigakan (misalnya karena frekuensi akses terlalu cepat).

5. Solusi

- Menambahkan header User-Agent saat mengirim request agar request terlihat seperti berasal dari browser normal, bukan bot.

- Memeriksa dan memperbarui selector secara berkala jika struktur HTML berubah.

- Menyimpan salinan halaman HTML secara lokal sebelum melakukan parsing, sehingga jika website tiba-tiba tidak bisa diakses, data tetap bisa diolah.

- Mengatur waktu delay di antara request untuk menghindari pemblokiran IP.

6. Kesimpulan

Proyek ini membantu memahami alur end-to-end dalam data pipeline: mulai dari pengambilan data (scraping), pembersihan data (preprocessing), hingga penyimpanan data untuk digunakan pada analisis lebih lanjut atau machine learning.

Selain melatih kemampuan teknis Python (requests, BeautifulSoup, pandas, NLP preprocessing), proyek ini juga mengajarkan pentingnya etika scraping dan perlunya adaptasi saat menghadapi website dinamis atau yang memiliki proteksi.