*Laporan Mata Kuliah Teks dan Web Mining*

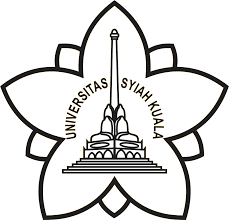
TUGAS 4: CRAWLING, SCRAPING, DAN EXTRACT CONTENT

disusun untuk memenuhi tugas Teks dan Web Mining

Oleh:

**SITI NURRAHMASITA**

**(2108107010015)**



# PROGRAM STUDI INFORMATIKA

# FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

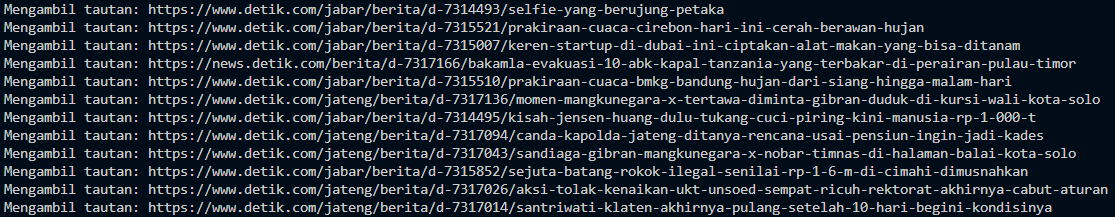
**UNIVERSITAS SYIAH KUALA**

**DARUSSALAM, BANDA ACEH**

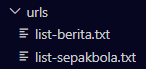
# 2024

1. **Penjelasan Umum**
2. Crawling (crawling.py):

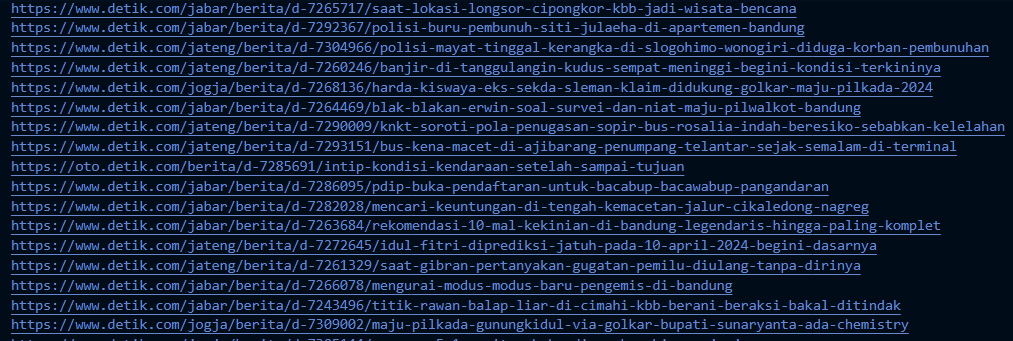
* Ini adalah langkah pertama dalam proses. Kode ini mengambil tautan dari situs detik.com menggunakan pencarian dan menyimpannya dalam file teks.
* Objek ‘DetikCrawler’ dibuat dengan topik pencarian dan jumlah data yang diinginkan.
* Kelas ini menggunakan library ‘requests’ untuk mengirimkan permintaan HTTP dan mendapatkan halaman web, dan ‘BeautifulSoup’ untuk mem-parsing HTML.
* Metode ‘get\_urls()’ mengambil tautan berita dari hasil pencarian detik.com dengan memeriksa halaman demi halaman.
* Metode ‘is\_topic\_related()’ memeriksa apakah tautan terkait dengan topik yang ditentukan.
* Metode ‘create\_url\_file()’ membuat file teks yang berisi tautan berita yang ditemukan.



**Gambar 1.a.** Output di terminal saat proses crawling data



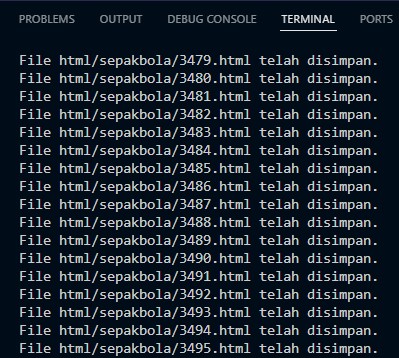
**Gambar 1.b.** Directory yang dihasilkan



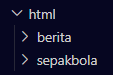
**Gambar 1.c.** Hasil crawling data

1. Scraping (scraping.py):

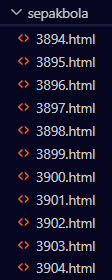
* Kode ini mengunduh seluruh konten dari setiap tautan-tautan telah berhasil dikumpulkan, yaitu dari tautan berita yang telah ditemukan oleh ‘crawling.py’.
* Menggunakan kembali library requests, file ini mengunduh halaman web dari setiap tautan yang telah dikumpulkan sebelumnya.
* Objek ‘DetikScraper’ dibuat dengan topik pencarian dan jumlah data yang diinginkan.
* Metode ‘extract\_html\_files()’ membaca file teks yang berisi tautan berita dan mengunduh halaman HTML untuk setiap tautan.
* Halaman HTML disimpan dalam direktori ‘html/{topic}’.



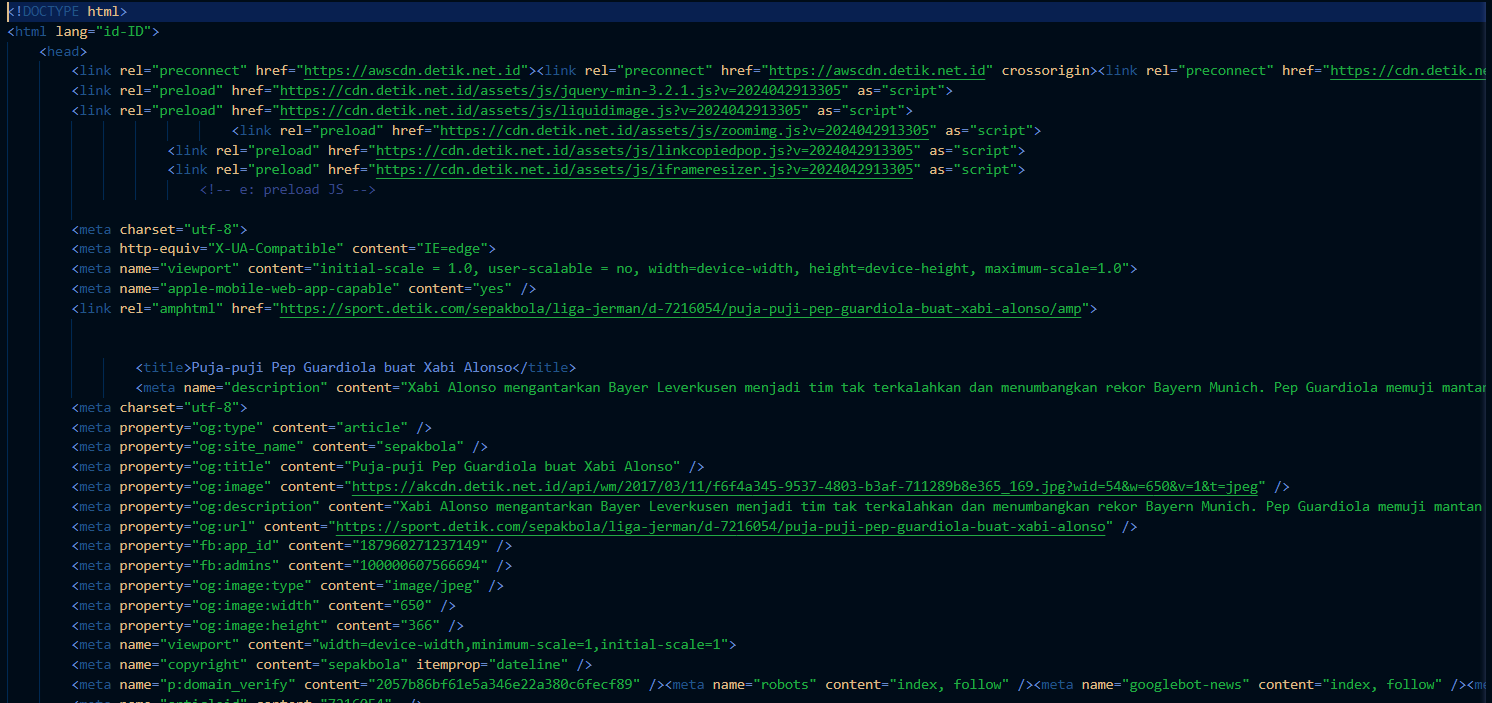
**Gambar 2.a.** Output di terminal saat proses scraping data



**Gambar 2.b.** Directory yang dihasilkan



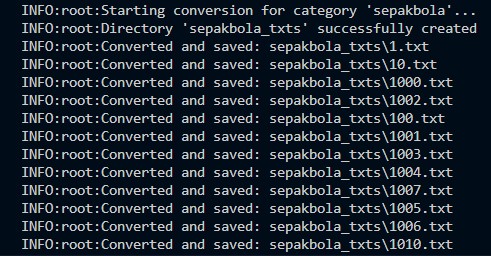
**Gambar 2.c.** Hasil scraping data



**Gambar 2.d.** Tampilan file yang telah diunduh

1. Extracting (extract.py):

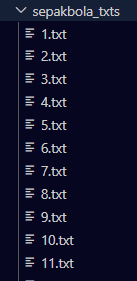
* Langkah ini bertanggung jawab untuk mengekstrak konten berita dari halaman HTML yang telah diunduh.
* Melalui metode ‘extract\_content\_from\_html’, judul dan isi berita diekstrak dari setiap halaman HTML.
* Konten ini kemudian disaring untuk menghilangkan elemen-elemen yang tidak diinginkan, seperti iklan atau konten tambahan, sebelum disimpan dalam file teks terpisah.
* Setiap halaman HTML diubah menjadi file teks yang berisi judul (Title) dan isi (Content) berita yang bersih.
* Direktori ‘html/{topic’ diubah menjadi file teks dengan format .txt.



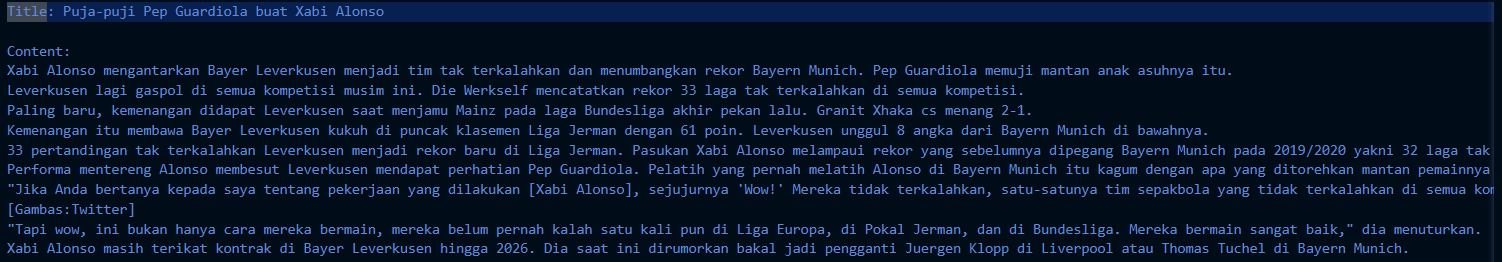
**Gambar 3.a.** Output di terminal saat proses extracting data dari format .html ke .txt



**Gambar 3.b.** Directory yang dihasilkan



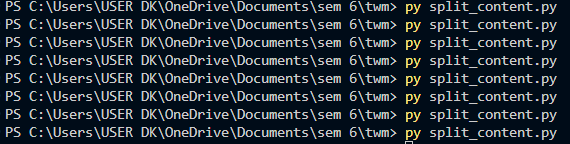
**Gambar 3.c.** Hasil extracting data



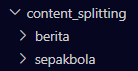
**Gambar 3.d.** Tampilan file yang telah diekstrak

1. Splitting Content (split\_content.py):

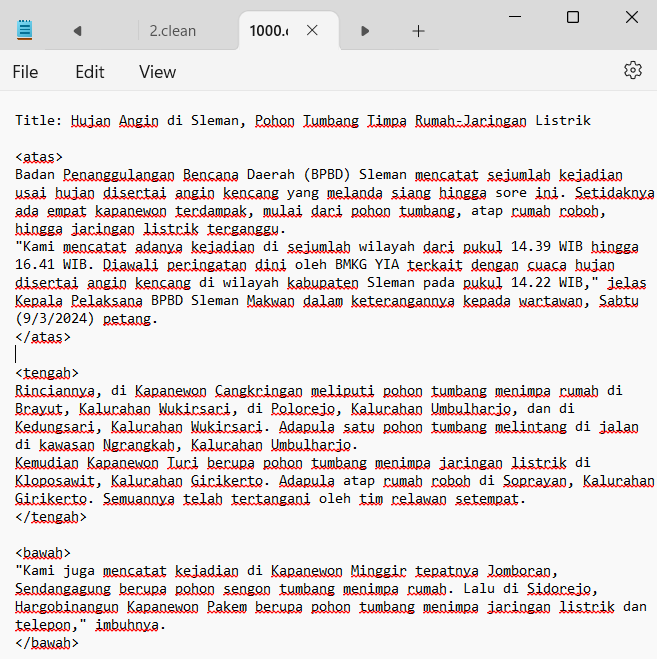
* Langkah terakhir ini membagi konten berita menjadi bagian atas, tengah, dan bawah.
* Bagian isi dibagi menjadi 3 bagian yaitu atas, tengah, dan bawah. Pembagian tersebut dilakukan berdasarkan jumlah baris. Jika isi dari file terdiri dari 12 kalimat, maka bagian atas memiliki 4 kalimat, bagian tengah 4 kalimat, dan bagian bawah 4 kalimat. Contoh lain, jika bagian isi terdiri dari 10 kalimat, maka bagian atas terdiri dari 4 kalimat, bagian tengah 3 kalimat, dan bagian bawah 3 kalimat. Gunakan tag <atas> </atas> untuk mengumpulkan kalimat-kalimat bagian atas, tag <tengah> </tengah> untuk menghimpun kalimat-kalimat bagian tengah, dan tag <bawah> </bawah> untuk menghimpun kalimat-kalimat bagian bawah.



**Gambar 4.a.** Output di terminal saat proses splitting isi/content data



**Gambar 4.b.** Directory yang dihasilkan



**Gambar 3.d.** Tampilan file yang telah dipisah dengan tag <atas> </atas>, <tengah> </tengah> , dan <bawah> </bawah>

1. **Unduhan 8000 File HTML**

Untuk mengunduh 8000 file HTML, langkah-langkahnya adalah sebagai berikut:

1. Jalankan ‘crawling.py’ dengan mengatur ‘num\_of\_data’ menjadi 4201 untuk setiap kategori yang ingin diunduh.
2. Setelah selesai mengunduh satu kategori, ubah topik pada kode ‘crawling.py’ (topic = "berita" atau topic = "sepakbola") untuk kategori berikutnya dan jalankan kembali kode tersebut.
3. Selanjutnya, jalankan scraping.py untuk mengunduh halaman HTML dari tautan yang telah ditemukan. Proses pengunduhan juga dilakukan bergantian antar kategori yang digunakan, yaitu dengan mengubah topik pada kode ‘scraping.py’ (topic = "berita" atau topic = "sepakbola")
4. Lampiran
5. Ss