

LAPORAN
PRAKTIKUM BIG DATA ANALYTIC
Pertemuan Ke - 5



Dosen :
Sri Redjeki, S.Si., M.Kom.

Disusun oleh :
RAHADIYAN BONDAN PERMADI
215411119

Universitas Teknologi Digital Indonesia
UTDI
YOGYAKARTA
2022

COLLECTING DATA

Dasar Teori

Web Scrapping *web harvesting*, atau *web data extraction* merupakan kegiatan yang dilakukan untuk mengambil data tertentu secara semi-terstruktur dari sebuah halaman [situs web](#). Halaman tersebut umumnya dibangun menggunakan bahasa markup seperti [HTML](#) atau [XHTML](#), proses akan menganalisis dokumen sebelum memulai mengambil data.

Biasanya teknik *scraping* diimplementasikan pada sebuah bot agar bisa membuat proses yang harusnya dilakukan secara manual menjadi otomatis. Ketika kita menjumpai sebuah situs yang membatasi kuota [API](#) (*application programming interface*) atau bahkan tidak menyediakan sama sekali, maka perayapan web akan sangat dibutuhkan sebagai langkah pengambilan data.

https://id.wikipedia.org/wiki/Web_scraping

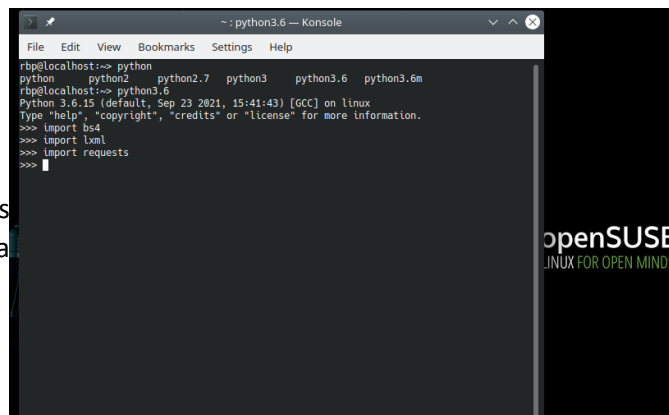
Kebutuhan Alat

1. Python (Anaconda / Miniconda)
2. Jupyter Notebook / Google Colab
3. BeautifulSoup4
4. Lxml
5. Request

Langkah – Langkah dalam praktikum install modul pandas dan xlrd pada python

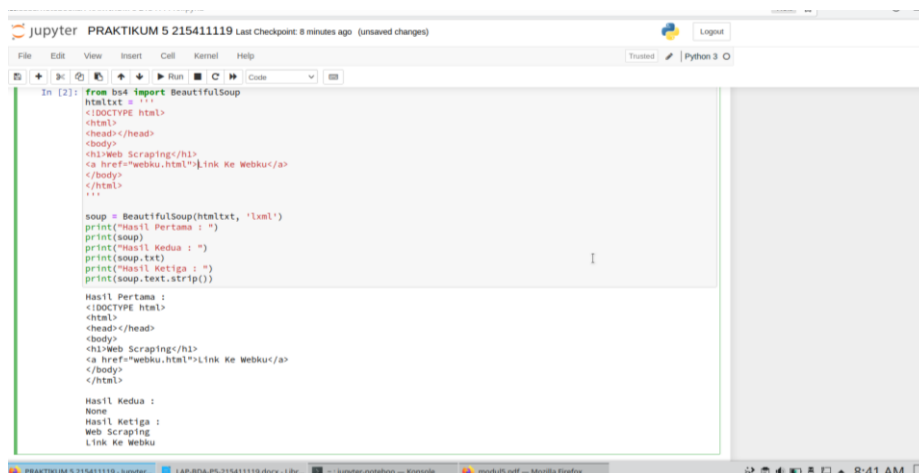
1. Install BeautifulSoup4
2. Install lxml
3. Install request

Pandas dan Xlrd sukses terinstall tidak ada error ketika import bs4, lxml dan request.



```
~: python3.6 — Konsole
File Edit View Bookmarks Settings Help
rbp@localhost:~$ python
python python2 python2.7 python3 python3.6 python3.6m
rbp@localhost:~$ python3.6
Python 3.6.15 (default, Sep 23 2021, 15:41:43) [GCC] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import bs4
>>> import lxml
>>> import requests
>>>
```

4. Fungsi `.text` dan `.strip` adalah untuk menghapus karakter didepan dan dibelakang yang ditentukan :

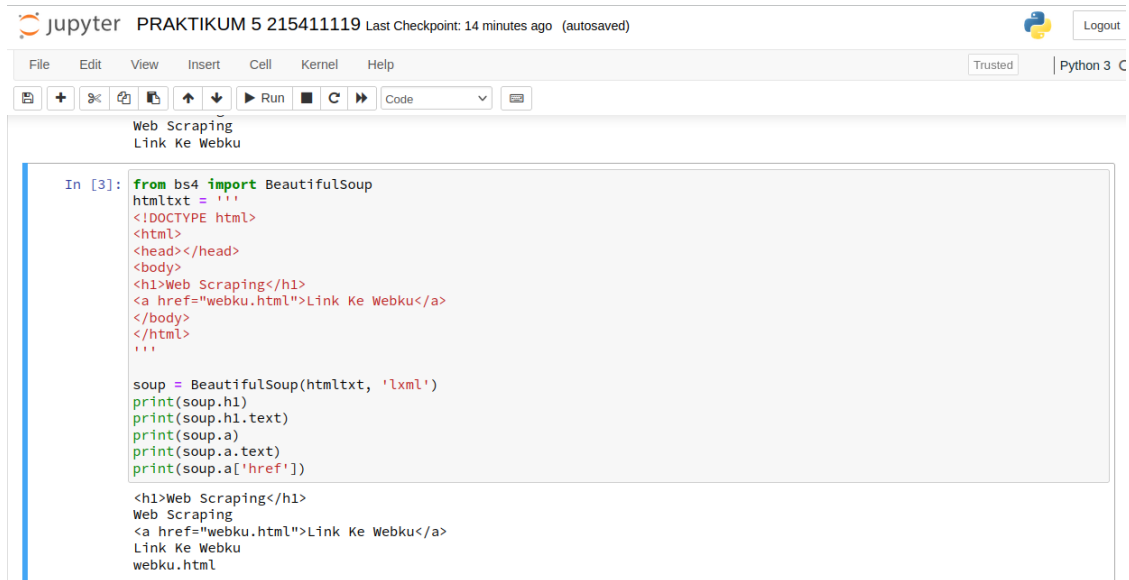


```
jupyter PRAKTIKUM 5 215411119 Last Checkpoint: 8 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Help Trusted Python 3
In [2]: from bs4 import BeautifulSoup
htmltext = '''
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>Web Scrapping</h1>
<a href="webku.html">Link Ke Webku</a>
</body>
</html>
'''
soup = BeautifulSoup(htmltext, 'lxml')
print("Hasil Pertama : ")
print(soup)
print("Hasil Kedua : ")
print(soup.text)
print("Hasil Ketiga : ")
print(soup.text.strip())

Hasil Pertama :
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>Web Scrapping</h1>
<a href="webku.html">Link Ke Webku</a>
</body>
</html>

Hasil Kedua :
None
Hasil Ketiga :
Web Scrapping
Link Ke Webku
```

5. Modifikasi script nomor 4 diatas menjadi seperti dibawah ini :



```
In [3]: from bs4 import BeautifulSoup
htmltxt = '''
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>Web Scraping</h1>
<a href="webku.html">Link Ke Webku</a>
</body>
</html>
'''

soup = BeautifulSoup(htmltxt, 'lxml')
print(soup.h1)
print(soup.h1.text)
print(soup.a)
print(soup.a.text)
print(soup.a['href'])

<h1>Web Scraping</h1>
Web Scraping
<a href="webku.html">Link Ke Webku</a>
Link Ke Webku
webku.html
```

Jalankan dan jelaskan !

print(soup.h1) – Untuk menampilkan seluruh karakter pada tag <h1>

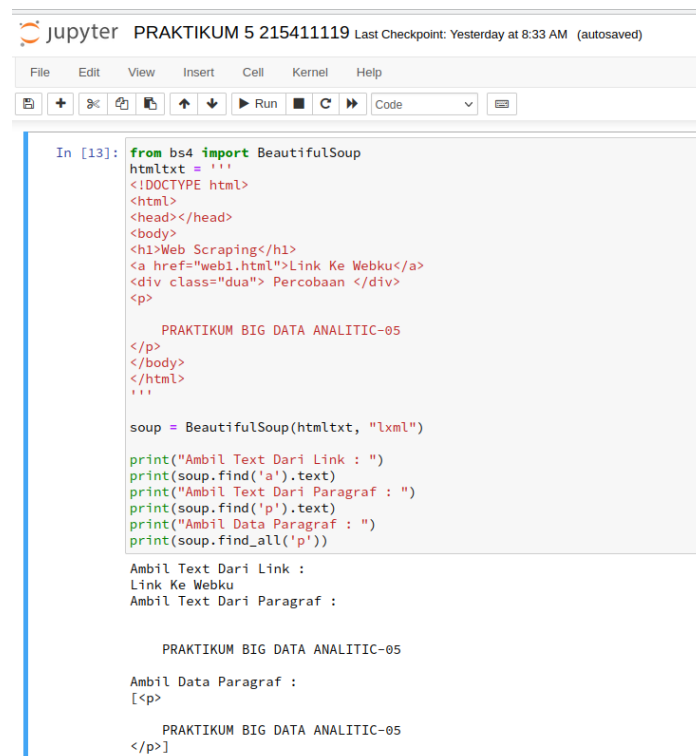
print(soup.h1.text) – Hanya untuk menampilkan text didalam tag <h1></h1>

print(soup.a) - Untuk menampilkan seluruh karakter pada tag <a>

print(soup.a.text) Hanya untuk menampilkan text didalam tag <a>

print(soup.a['href']) – ['href'] Hanya untuk menampilkan link didalam tag <a>

6. Gunakan file web1.html perbedaan hasil find dan find_all adalah : find untuk menemukan data didalam paragraf, sedangkan find_all() untuk mengambil seluruh data (source) yang memuat paragraf.



```
In [13]: from bs4 import BeautifulSoup
htmltxt = '''
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>Web Scraping</h1>
<a href="web1.html">Link Ke Webku</a>
<div class="dua"> Percobaan </div>
<p>

PRAKTIKUM BIG DATA ANALITIC-05
</p>
</body>
</html>
'''

soup = BeautifulSoup(htmltxt, "lxml")

print("Ambil Text Dari Link : ")
print(soup.find('a').text)
print("Ambil Text Dari Paragraf : ")
print(soup.find('p').text)
print("Ambil Data Paragraf : ")
print(soup.find_all('p'))

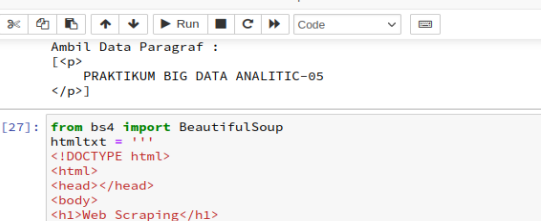
Ambil Text Dari Link :
Link Ke Webku
Ambil Text Dari Paragraf :

PRAKTIKUM BIG DATA ANALITIC-05

Ambil Data Paragraf :
[<p>

PRAKTIKUM BIG DATA ANALITIC-05
</p>]
```

7. Membuat atau menarik data dari Class Dua



The screenshot shows a Jupyter Notebook interface. At the top, the Jupyter logo is on the left, and the text "PRAKTIKUM 5 21541119" is in the center. To the right of the text is a status bar that says "Last Checkpoint: Yesterday at 8:33 AM (unsaved changes)". Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, and Help. Under the "Cell" menu, the "Run" option is selected, indicated by a play button icon. Below the menu bar is a toolbar with various icons for cell operations. The main area of the notebook contains a code cell with the following text: "Ambil Data Paragraf :". Below this text is a code block enclosed in XML-style tags: `<p>PRAKTIKUM BIG DATA ANALITIC-05</p>`. The code cell is followed by an input prompt "In [27]:" and a code block that defines a BeautifulSoup object and extracts text from an HTML string. The HTML string is a snippet of a web page containing a link. The code then prints the extracted text and assigns it to a variable named "link".

Jupyter PRAKTIKUM 5 21541119 Last Checkpoint: Yesterday at 8:33 AM (unsaved changes)

File Edit View Insert Cell Kernel Help

Run

Ambil Data Paragraf :

```
<p>PRAKTIKUM BIG DATA ANALITIC-05</p>
```

In [27]:

```
from bs4 import BeautifulSoup
htmltxt = '''
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>Web Scraping</h1>
<a href="web1.html">Link Ke Webku</a>
<div class="dua"> Percobaan </div>
<p> PRAKTIKUM BIG DATA ANALITIC-05
</p>
</body>
</html>
'''

soup = BeautifulSoup(htmltxt, "lxml")

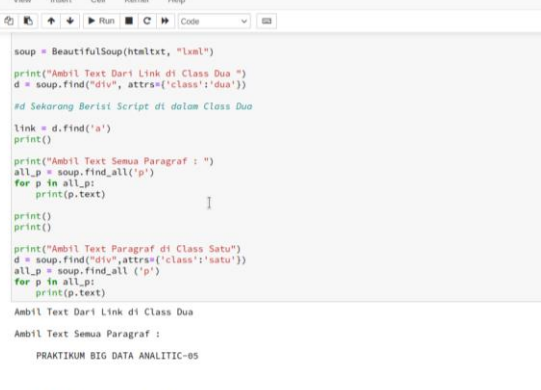
print("Ambil Text Dari Link di Class Dua ")
d = soup.find("div", attrs={'class':'dua'})

#d Sekarang Berisi Script di dalam Class Dua

link = d.find('a')

Ambil Text Dari Link di Class Dua
```

8. Modifikasi script nomor 7



```
 soup = BeautifulSoup(htmltxt, "lxml")

print("Ambil Text Dari Link di Class Dua ")
d = soup.find('div', attrs={'class':'dua'})

#d Sekarang Berisi Script di dalam Class Dua

link = d.find('p')
print()

print("Ambil Text Semua Paragraf : ")
all_p = soup.find_all('p')
for p in all_p:
    print(p.text)

print()
print()

print("Ambil Text Paragraf di Class Satu")
d = soup.find('div', attrs={'class':'satu'})
all_p = soup.find_all('p')
for p in all_p:
    print(p.text)
```

Ambil Text Dari Link di Class Dua

Ambil Text Semua Paragraf :

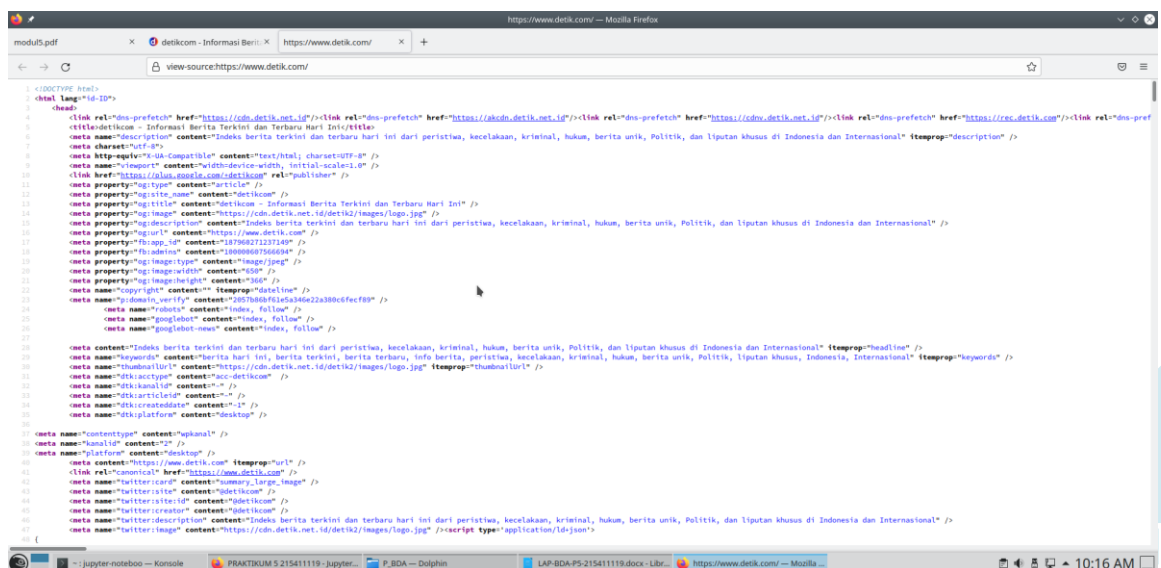
PRAKTIKUM BIG DATA ANALITIC-05

Ambil Text Paragraf di Class Satu

PRAKTIKUM BIG DATA ANALITIC-05

9. Membuka halaman detik.com dan inspect pada bagian “terpopuler”

10. Klik kanan dan buka page source



11. Membuat File baru menggunakan web detik.com dari tag (Terpopuler dari Detik.com)

```
jupyter PRAKTIKUM 5 215411119 Last Checkpoint: Yesterday at 8:33 AM (autosaved)

AttributeError                                Traceback (most recent call last)
<ipython-input-78-6c318985d748> in <module>
    12 b = s.find('div', attrs={'class':'box cb-mostpop'})
    13 #di dalam class tsb, temukan semua class = 'media_title'
----> 14 judul = b.find_all('h3', attrs={'class':'media_title'})
    15 #baca tiap bagian dari judul, dan tampilkan
    16 print("Terpopuler dari Detik.com")

AttributeError: 'NoneType' object has no attribute 'find_all'

In [78]: from bs4 import BeautifulSoup
import requests
import html5lib
import time

url = "https://www.detik.com/"
web = requests.get(url).text
s = BeautifulSoup(web, "lxml")
b = s.find('div', attrs={'class':'box cb-mostpop'})
judul = b.find_all('h3', attrs={'class':'media_title'})
print("Terpopuler dari Detik.com")

for j in judul:
    print('Judul : ',j.text)

Terpopuler dari Detik.com
```

12. Praktikum no 12 scraping web ini sedikit memodifikasi dikarenakan source pada modul tidak sesuai dengan versi python yang saya gunakan, adalah python3

```
jupyter PRAKTIKUM 5 215411119 Last Checkpoint: a day ago (autosaved)

Defaulting to user installation because normal site-packages is not writeable
[WARNING] Invalid requirement: 'response'
Note: you may need to restart the kernel to use updated packages.

In [21]: import urllib3
import xlsx

result = xlsx.Workbook()
sheet = result.add_sheet("product info")
sheet.write(0,1, "Product Name")
sheet.write(0,2, "Price")
wiki = "https://www.frankana.de/de/multimedia.html"

from bs4 import BeautifulSoup
http = urllib3.PoolManager()
page = http.request('GET', wiki)
#page = requests.get(wiki)
#page = urllib3(wiki)
soup = BeautifulSoup(page, "html.parser")
soup = BeautifulSoup(page.data)
all_products = soup.find_all("li", {"class": "item product product-item"})
index = 0
for product in all_products:
    productname = product.find('div', {'class':'product details product-item-details'}).find("a", {"class": "product-
index = index + 1
sheet.write(index, 1, productname)
price = product.find('div', {'class':'price-box price-final_price'}).find_all("span", {"class": "price"})[-1].tex
sheet.write(index, 2, price)
result.save("product listproduk.xls")
```

HASIL :

The screenshot shows a Windows desktop environment. On the left, a file explorer window displays the 'Home' directory, with 'product listproduk.xls' highlighted. On the right, the LibreOffice Calc application is open, displaying the contents of 'product listproduk.xls'. The spreadsheet has two columns: 'Product Name' and 'Price'. The data is as follows:

Product Name	Price
12-Volt-Anschlusskabel Megasat	6,90 €
Abisolierwerkzeug für Satkabel	6,90 €
Ablagefach für CDs für Fiat Ducato ab Baujahr 07/12	9,90 €
Ablagefach für Fiat Ducato ab Baujahr 07/2006, ur	9,90 €
Abschlusskappe, oben für Aluminiummast HDM 1	5,50 €
Abschlusskappe, unten für Aluminiummast HDM 1	5,50 €
Adapter Lenkradfernbedienung	45,00 €
Adapter Lenkradfernbedienung mit TMC	59,90 €
Adapterplatte VESA für Wandhalterung Sky N und	39,90 €
Aktiv-Subwoofer ESX quantum Q168A	199,00 €
alphatronics CTS SL-Line	699,00 €
alphatronics K-Line SB+	329,00 €

Karena lib yang terbaru lib3 ter up to date pada dokumentasi python3 dan di source modul5 tidak mendukung sehingga amemodifikasi menggunakan `http = urllib3.PoolManager()`

<https://pypi.org/project/urllib3/>

dimana sebelumnya menginstall urllib3 menggunakan perintah “pip install urllib3”

Demikian laporan Pertemuan Ke-Lima yang dapat saya rangkum dan saya kerjakan, saya dapat mempraktekkan penggunaan scraping web dan penggunaan library urllib3.

=====Terimakasih=====