


# BAB I

## COLLECTING DATA

### PEMBAHASAN

#### Praktik

1. Masuk ke folder c:\python
2. Instal module pandas dan xlrd, dengan menjalankan :

 C:\python>conda install BeautifulSoup4

```
(base) C:\Users\Asus\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Python 3
.8>conda install beautifulsoup4
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\Asus\anaconda3

added / updated specs:
- beautifulsoup4

The following packages will be downloaded:

package | build | size
-----|-----|-----
conda-4.12.0 | py39haa95532_0 | 14.5 MB
-----|-----|-----
Total: | | 14.5 MB

The following packages will be UPDATED:

conda 4.10.3-py39haa95532_0 --> 4.12.0-py39haa95532_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.12.0 | 14.5 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

#### Install lxml

```
(base) C:\Users\Asus\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Python 3
.8>conda install lxml
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

#### Install requests

```
(base) C:\Users\Asus\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Python 3
.8>conda install requests
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

3. Untuk memastikan instalasi berhasil, masuk ke REPL python

(base) C:\Users\ASUS\python

```
(base) C:\Users\Asus>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anacon
da, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import bs4
>>> import lxml
>>> import requests
>>>
```

*Jika tidak ada pesan Error, berarti instalasi pandas dan xlrd sukses*

4. Bukalah Jupyter Notebook, ketikkan script berikut ini :

```
In [1]: 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("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 Scraping</h1>
<a href="webku.html">Link ke Webku</a>
</body>
</html>
```

```
Hasil Kedua :
```

```
Web Scraping
Link ke Webku
```

```
Hasil Ketiga :
Web Scraping
Link ke Webku
```

Uraian :

Pada langkah ini melakukan scrapping terhadap semua data yang berada di dalam tag <html>. Dalam source code tersebut diketahui bahwa beautiful soup membutuhkan 2 argumen yaitu pada argumen pertama adalah markup yang ingin diproses dan argumen yang kedua adalah parser yang ingin digunakan. Soup berfungsi untuk menampilkan keseluruhan hasil scraping sedang soup.text akan menampilkan hasil scraping data yang berupa teks saja dan soup.text.strip() berfungsi untuk menampilkan data teks serta menghilangkan jarak diantara hasil teksnya.

5. Buatlah file baru dan copykan script program diatas, kemudian modifikasilah seperti dibawah ini :

```
In [1]: 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
```

Activate Windows

Uraian :

Pada langkah ini akan mengambil data pada suatu tag <html> dengan ketentuan tertentu sehingga hasil yang ditampilkan lebih spesifik.

- Soup.h1 digunakan untuk mengambil semua data yang mempunyai tag <h1> , sedangkan soup.h1.text berfungsi untuk melakukan mengambil semua data text yang berdata didalam tag <h1> .
  - Soup.a berfungsi untuk mengambil data yang memiliki tag a sedangkan soup.a.text berfungsi untuk mengambil data text yang berada didalam tag <a> ,
  - Sedangkan soup.a['href'] berfungsi untuk mengambil data link pada tag <a> .
6. Gunakan file web1.html, kemudian buatlah file baru dan ketikkan perintah seperti dibawah ini :

```
In [11]: from bs4 import BeautifulSoup

html = open('web1.html', 'r')

s = BeautifulSoup(html, 'lxml')

print("Ambil Text dari link : ")
print(s.find('a').text)
print("Ambil Text Paragraf : ")
print(s.find('p').text)
print("Ambil Data Paragraf : ")
print(s.find_all('p'))
```

---

```
-----
TypeError                                Traceback (most recent
call last)
~\AppData\Local\Temp\ipykernel_9844\3512195391.py in <module>
      1 from bs4 import BeautifulSoup
      2
----> 3 html = open('web1.html', 'r')
      4
      5 s = BeautifulSoup(html, 'lxml')
```

**TypeError:** 'tuple' object is not callable

Uraian :

Ketika menggunakan source code pada modul terdapat kendala yakni pada saat ingin membaca web html web1.html, source code tersebut sudah tidak dapat digunakan lagi. Maka , saya menggunakan source code pada langkah 5 agar dapat mengambil data yang ada pada web1.html.

```
In [34]: 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"> tes </div>
<p>
    Praktikum Big Data Analytic-05
</p>
</body>
</html>
'''

soup = BeautifulSoup(htmltxt, 'lxml')
print("Ambil Text dari link : ")
print(soup.find('a').text)
print("Ambil Text Paragraf : ")
print(soup.find('p').text)
print("Ambil Data Paragraf : ")
print(soup.find_all('p'))
```

```
Ambil Text dari link :
Link ke Webku
Ambil Text Paragraf :

    Praktikum Big Data Analytic-05

Ambil Data Paragraf :
[<p>
    Praktikum Big Data Analytic-05
</p>]
```

Apa perbedaan fungsi dan hasil dari find() dan find\_all() ?

Uraian :

Fungsi find('a').text berfungsi untuk menemukan tag <a> dan mengambil data text yang berada didalamnya , sedangkan fungsi find('p').text digunakan untuk menemukan tag paragraf atau tag <p> dan mengambil data text yang berada didalamnya. Untuk fungsi find\_all('p') digunakan untuk mengambil semua data yang memiliki tag <p>.

7. Buatlah file baru dan ketikkan script berikut ini :

```
In [35]: from bs4 import BeautifulSoup
htmltxt = '''
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<h1>Web Scraping</h1>
<div class="dua">
    <a href="webku.html">Link ke Webku</a>
</div>
</body>
</html>
'''

soup = BeautifulSoup(htmltxt, 'lxml')
print("Ambil Text dari link di Class dua")
d = soup.find("div", attrs={'class':'dua'})

link = d.find('a')
print(link.text)

Ambil Text dari link di Class dua
Link ke Webku
```

Uraian :

Pada langkah ini akan mengambil text dari link yang ada pada web1.html. source code yang digunakan tidak jauh berbeda menggunakan source code pada langkah 5, karena pada modul, source code tersebut tidak dapat dijalankan. Fungsi "find("div", attrs={'class':'dua'})" mengambil data yang memiliki tag <div class="dua"> kemudian datanya akan disimpan kedalam variabel bernama d . Hasil data yang disimpan pada variabel d ini kemudian diambil lagi datanya yang memiliki tag <a> dan hasilnya

disimpan pada variabel link. Pada akhir scribt data text yang berada pada isi dari variabel link akan ditampilkan

8. Kemudian modifikasilah file dari no.07 , tambahkan perintah seperti berikut ini :

```
In [58]: from bs4 import BeautifulSoup
html = '''
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<h1>Web Scraping</h1>
<div class="dua">
    <a href="webku.html">Link ke Webku</a>
</div>
</body>
</html>
'''

soup = BeautifulSoup(html, "lxml")

print("Ambil Text dari link di Class dua")
d = s.find("div", attrs={'class': 'dua'})

print()
print("Ambil Text Semua Paragraf : ")
for p in all_p :
    print(p.text)

print()

print("Ambil text paragraf di class Satu")
d = s.find("div", attrs={'class' : 'satu'})

for p in all_p :
    print(p.text)
```

Ambil Text dari link di Class dua

Ambil Text Semua Paragraf :  
Rabu, 16 Mar 2022 19:28 WIB  
Rabu, 16 Mar 2022 13:02 WIB

Ambil text paragraf di class Satu  
Rabu, 16 Mar 2022 19:28 WIB  
Rabu, 16 Mar 2022 13:02 WIB

Uraian :

Pada source code diatas akan mengambil beberapa data berupa text dari laman webku.html dimana pada saat pengambilan data nanti akan dibagi menjadi 2 yaitu class 1 dan class 2. Pada langkah ini saya tidak mengikuti perintah source code yang ada dimodul, dikarenakan beberapa perintah source code tersebut telah usang dan tidak dapat digunakan, oleh karena itu saya modifikasi beberapa source code yang ada pada langkah 8 ini.

9. Bukalah Alamat Website : <https://www.detik.com>. Inspect element pada bagian “Terpopuler” :

The image shows two screenshots of the detik.com website. The top screenshot is the homepage, featuring a blue header with the detik.com logo, a search bar, and navigation links. Below the header is a large banner for 'BAYAR CUMA 10%' and a row of category links. The main content area displays several news thumbnails with headlines such as 'Seriusan Stiker Ms Glow&399 Corp Dicotop dari Pesawat Juragan 99?', 'Aleix Espargaro Lunasi Janji Lempar Helm ke Tribune', and 'Citaris R Ditangkap dengan Barang Bukti Ganja 8 Gram'. The bottom screenshot shows the search results for 'terpopuler', displaying three news items: 'Sirkuit Mandalika Dapat Grade A-Moto GP Pakai Teknologi sampai Pawang Hujan', 'Gus Mus Bicara Esensi Jumatana hingga Kegiatan Politik Berbungkus Ibadah', and 'Teluk Ada Alas dan Bekas Kapal yang Berupa di Ujung'.

detik.com

BAYAR CUMA 10%

detikNews detikFinance detikHot detikinet detikSport detikOto detikTravel detikFood detikHealth Wolipop 20Detik Daerah

Live TV Adsmart Foto detikX Sepakbola Pasangmata Hikmah Edukasi berbuatbaik.id Live Streaming MotoGP

Seriusan Stiker Ms Glow&399 Corp Dicotop dari Pesawat Juragan 99?

Aleix Espargaro Lunasi Janji Lempar Helm ke Tribune

Citaris R Ditangkap dengan Barang Bukti Ganja 8 Gram

Profil Rara Istiati Pawang Hujan Mandalika, Bayarannya Tembus Ratusan Juta

Klasemen MotoGP 2022 Usai Oliveira Juara di Mandalika

Candi Prambanan-Borobudur Siap Sambut Delegasi G20, Ini...

3 Isu Prioritas yang Dibawa RI di Forum Ekonomi Digital G20

detik.com

Hasil pencarian "terpopuler", 7859 hasil ditemukan

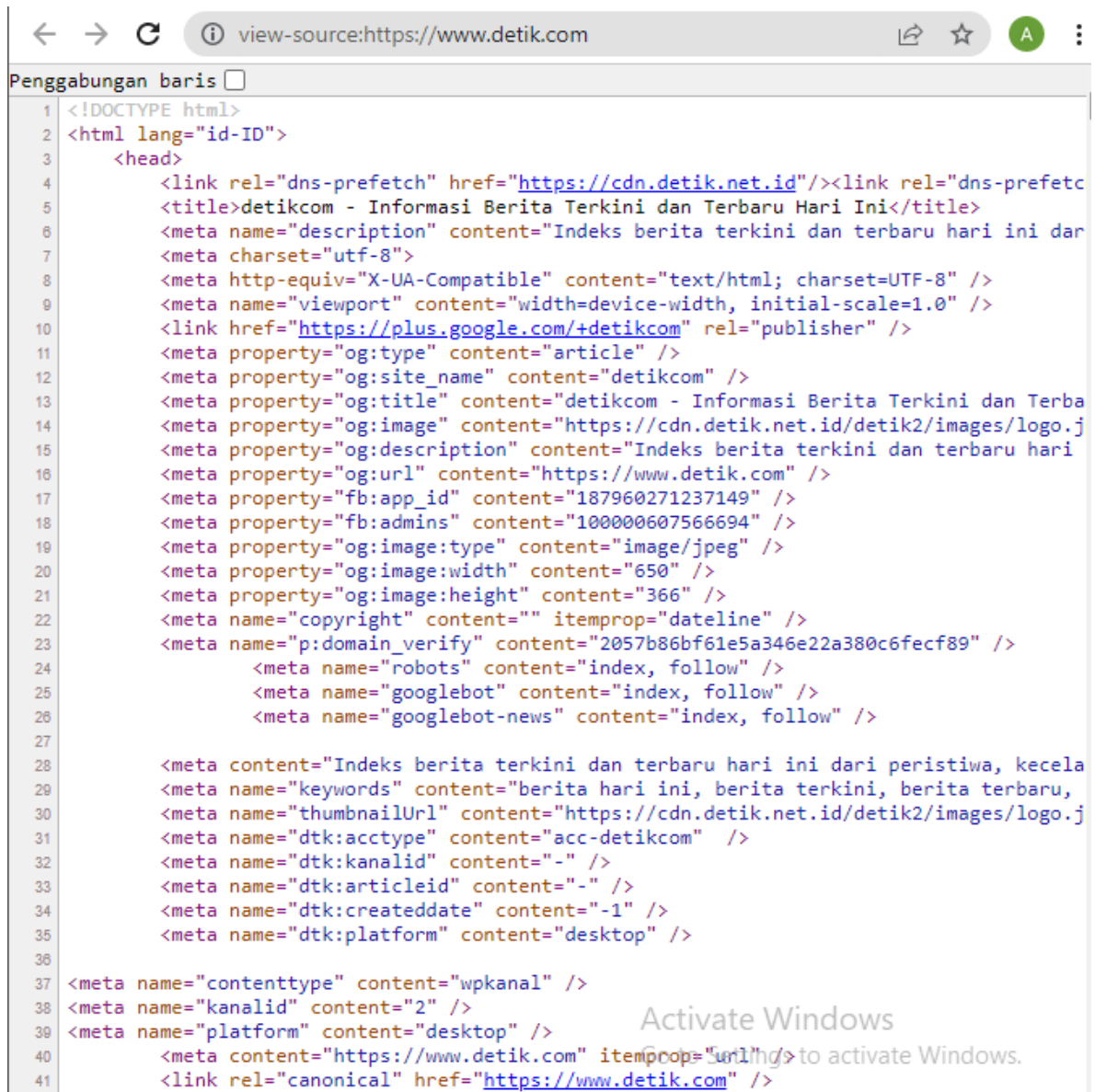
Sirkuit Mandalika Dapat Grade A-Moto GP Pakai Teknologi sampai Pawang Hujan

Gus Mus Bicara Esensi Jumatana hingga Kegiatan Politik Berbungkus Ibadah

Teluk Ada Alas dan Bekas Kapal yang Berupa di Ujung



10. Klik kanan pada halaman tersebut, kemudian View Source



```
1 <!DOCTYPE html>
2 <html lang="id-ID">
3   <head>
4     <link rel="dns-prefetch" href="https://cdn.detik.net.id"/><link rel="dns-prefetch" href="https://www.detik.com" />
5     <title>detikcom - Informasi Berita Terkini dan Terbaru Hari Ini</title>
6     <meta name="description" content="Indeks berita terkini dan terbaru hari ini dari detikcom" />
7     <meta charset="utf-8">
8     <meta http-equiv="X-UA-Compatible" content="text/html; charset=UTF-8" />
9     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
10    <link href="https://plus.google.com/+detikcom" rel="publisher" />
11    <meta property="og:type" content="article" />
12    <meta property="og:site_name" content="detikcom" />
13    <meta property="og:title" content="detikcom - Informasi Berita Terkini dan Terbaru Hari Ini" />
14    <meta property="og:image" content="https://cdn.detik.net.id/detik2/images/logo.jpg" />
15    <meta property="og:description" content="Indeks berita terkini dan terbaru hari ini dari detikcom" />
16    <meta property="og:url" content="https://www.detik.com" />
17    <meta property="fb:app_id" content="187960271237149" />
18    <meta property="fb:admins" content="100000607566694" />
19    <meta property="og:image:type" content="image/jpeg" />
20    <meta property="og:image:width" content="650" />
21    <meta property="og:image:height" content="366" />
22    <meta name="copyright" content="" itemprop="dateline" />
23    <meta name="p:domain_verify" content="2057b86bf61e5a346e22a380c6fecf89" />
24    <meta name="robots" content="index, follow" />
25    <meta name="googlebot" content="index, follow" />
26    <meta name="googlebot-news" content="index, follow" />
27
28    <meta content="Indeks berita terkini dan terbaru hari ini dari peristiwa, kecelakaan, kriminalitas, kesehatan, ekonomi, politik, olahraga, hiburan, teknologi, lingkungan, dan lainnya" />
29    <meta name="keywords" content="berita hari ini, berita terkini, berita terbaru, berita detikcom" />
30    <meta name="thumbnailUrl" content="https://cdn.detik.net.id/detik2/images/logo.jpg" />
31    <meta name="dtk:acctype" content="acc-detikcom" />
32    <meta name="dtk:kanalid" content="-" />
33    <meta name="dtk:articleid" content="-" />
34    <meta name="dtk:createddate" content="-1" />
35    <meta name="dtk:platform" content="desktop" />
36
37    <meta name="contenttype" content="wpkanal" />
38    <meta name="kanalid" content="2" />
39    <meta name="platform" content="desktop" />
40    <meta content="https://www.detik.com" itemprop="url" />
41    <link rel="canonical" href="https://www.detik.com" />
```

Uraian :

Ketika view source pada laman <https://www.detik.com> , maka akan menampilkan source code seperti hasil output diatas.

11. Buatlah file baru dan ketikkan perintah berikut ini :

```
In [42]: from bs4 import BeautifulSoup
import requests

url = "https://www.detik.com"
#ambil isi url, masukkan ke variabel web
web = requests.get(url).text
#parsing isi variabel web
s = BeautifulSoup(web, "lxml")
#ambil class box cb-mostpop, karena hanya ada 1, maka cukup pakai f
b = s.find('div', attrs={'class' : 'box cb-mostpop'})
#didalam class tbs, temukan semua class 'media_title'
judul = b.find_all('h3', attrs={'class' : 'media_title'})
#baca tiap bagian dari judul, dan tampilan
print("Terpopuler dari Detik.com")
for j in judul :
    print('Judul : ',j.text)
```

Terpopuler dari Detik.com

Uraian :

Pada source code diatas akan menampilkan tiap bagian dari judul dan tampilan yang ingin dikeluarkan (output) adalah terpopuler dari detik.com .

12. Gunakan file tabel.html lalu buat file baru, kemudian ketikkan script berikut ini

```
In [19]: from bs4 import BeautifulSoup as bs
html = open ("tabel.html", "r")
s = bs(html, "lxml")
tb = s.find("table")
tr = s.find("table")

tr = tb.find_all("tr")
for row in tr :
    #ambil semua td
    td = row.find_all("td")
    if td :
        #baris pertama tr berisi th, jadi nilai td=false
        #maka perlu ada if td, agar yang ditampilkan
        #hanya yang ada isinya
        #tampilkan jabatan dan tunjangan saja
        print("Jabatan : ", td[2].text, "Tunjangan : ", td[3].text)
```

-----

**TypeError** Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel\_9844\3017242305.py in <module>

```
1 from bs4 import BeautifulSoup as bs
----> 2 html =open ("tabel.html", "r")
      3 s = bs(html, "lxml")
      4 tb = s.find("table")
      5 tr = s.find("table")
```

Uraian :

Pada saat dijalankan, terdapat kendala pada tabel.html, dimana data html tersebut tidak dapat diakses (baca) dalam source code sehingga proses pengambilan data tidak dapat dilakukan.

```

In [97]: from bs4 import BeautifulSoup as bs
        html = '''
        <!DOCTYPE html>
        <html>
        <head>
        </head>
        <body>
        <h1>Web Scraping</h1>
        <a href = "tabel.html"></a>
        </body>
        </html>
        '''

        s = BeautifulSoup(html, "lxml")
        tb = BeautifulSoup("table")
        tr = BeautifulSoup("table")

        tr = tb.find_all("tr")
        for row in tr :
            #ambil semua td
            td = row.find_all("td")
            if td :
                #baris pertama tr berisi th, jadi nilai td=false
                #maka perlu ada if td, agar yang ditampilkan
                #hanya yang ada isinya
                #tampilkan jabatan dan tunjangan saja
                print("Jabatan : ", td[2].text, "Tunjangan : ", td[3].text)

```

```

In [85]: #import library used to query a website

```

Uraian :

Source code diatas merupakan hasil modifikasi pada langkah 12 sebelumnya, dimana dalam hal ini akan mengambil data berupa jabatan dan tunjangan yang ada pada tabel.html, namun ketika program dijalankan tidak menampilkan hasil apapun.

## Scrapping data Online Shop

1. Kita buka contoh online shop <https://www.frankana.de/>
2. Ketikkan perintah berikut ini :

```
In [21]: #import library used to query a website
import urllib3
import xlwt
result = xlwt.Workbook()
sheet = result.add_sheet("product info")
sheet.write(0,1,"Product Name")
sheet.write(0,2, "Price")
#specify the url
wiki = "https://www.frankana.de/de/multimedia.html"
#import the beautiful soup functions to parse the data returned from
from bs4 import BeautifulSoup
page = urllib3.urlopen(wiki)
#parse the html in the 'page' variable , and store it in Beautiful
soup = BeautifulSoup(page, "html,parser")
all_products = soup.find_all("li",{"class" : "item last"})
index = 0
for product in all_products :
    productname = product.find("h2", {"class" : "product-name"}).fir
    index = index + 1
    sheet.write(index, 1, productname)
    price = product.find("div", {"class" : "price box"}).find_all("
    sheet.write(index, 2, price)
result.save("product list.xls")
```

```
-----
-----
NameError                                Traceback (most recent
call last)
~\AppData\Local\Temp\ipykernel_9844\1719671756.py in <module>
    10 #import the beautiful soup functions to parse the data re
turned from the website
    11 from bs4 import BeautifulSoup as bs
--> 12 urllib3 = urlopen('wiki')
    13 #parse the html in the 'page' variable , and store it in
Beautiful Soup format
    14 soup = BeautifulSoup(page, "html,parser")

NameError: name 'urlopen' is not defined
```

Hasil akan disimpan dengan nama file product list dalam bentuk excel. File disimpan dalam satu folder sama dengan nama file python kita.

Uraian :

Pada saat proses scraping pada laman [www.frankana.de](https://www.frankana.de) proses untuk pemanggilan `urllib3.urlopen(wiki)` tidak dapat dijalankan, sehingga proses scraping data tersebut tidak berhasil.

## Latihan

1. Jelaskan perbedaan web scrapping dan web crawling. Berikan contoh coding web crawling

Uraian :

Pengertian :

- a. Web crawling adalah teknik pengumpulan data yang digunakan untuk mengindeks informasi pada halaman menggunakan URL (Uniform Resource Locator) dengan menyertakan API (Application Programming Interface) untuk melakukan penambahan dataset yang lebih besar. Data yang dikumpulkan dapat berupa text, audio, video, dan gambar.
- b. Web scraping adalah proses pengambilan data atau esktraksi dari sebuah website, lalu data tersebut umumnya disimpan dalam sebuah format tertentu.

Perbedaannya :

- a. Web scraping mengacu pada ekstraksi data dari situs web atau halaman web yang biasanya data ini diekstraksi ke dalam format file yang baru misalnya data dari situs web dapat diekstraksi ke dalam spreadsheet excel, ataupun csv. Web scraping juga dapat dilakukan secara manual dengan cara melakukan parsing menggunakan HTML atau XML, meskipun dalam banyak kasus automation tools dapat digunakan untuk mengekstrak data. Akan tetapi jika, kamu ingin mendapatkan data dengan pendekatan terfokus untuk analisis lebih lanjut kamu dapat melakukannya dengan cara manual tersebut. Misalnya perusahaan mungkin mengekstrak detail produk dari pada salah satu situs e-commerce untuk mengetahui bagaimana mereka memposisikan produk mereka di pasar.
- b. Sementara web crawling mengacu pada proses penggunaan BOT atau spider untuk membaca dan menyimpan semua konten di situs web untuk tujuan pengarsipan dan pengindeksan mesin pencari seperti bing atau google menggunakan web crawling untuk mengekstrak semua informasi dari situs web dan mengindeksnya di situs mereka. Selain itu, web crawling biasanya dapat menggunakan API tanpa harus melakukan parsing HTML. Jadi, meskipun web scraping dan web crawling memiliki istilah yang mengacu pada ekstraksi data tapi, mereka memiliki perbedaan tujuan serta aplikasi-aplikasi untuk web scraping dan web crawling juga sangat berbeda.

Contoh coding :

a. Web scrapping

```
In [1]: from bs4 import BeautifulSoup
htmltxt = '''
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>Web Scrapping</h1>
<a href = "webku.html">Link ke Webku</a>
</body>
</html>
'''

soup = BeautifulSoup (htmltxt,'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>
```

Activate Windows  
Go to Settings to activate Windows.

b. Web crawling

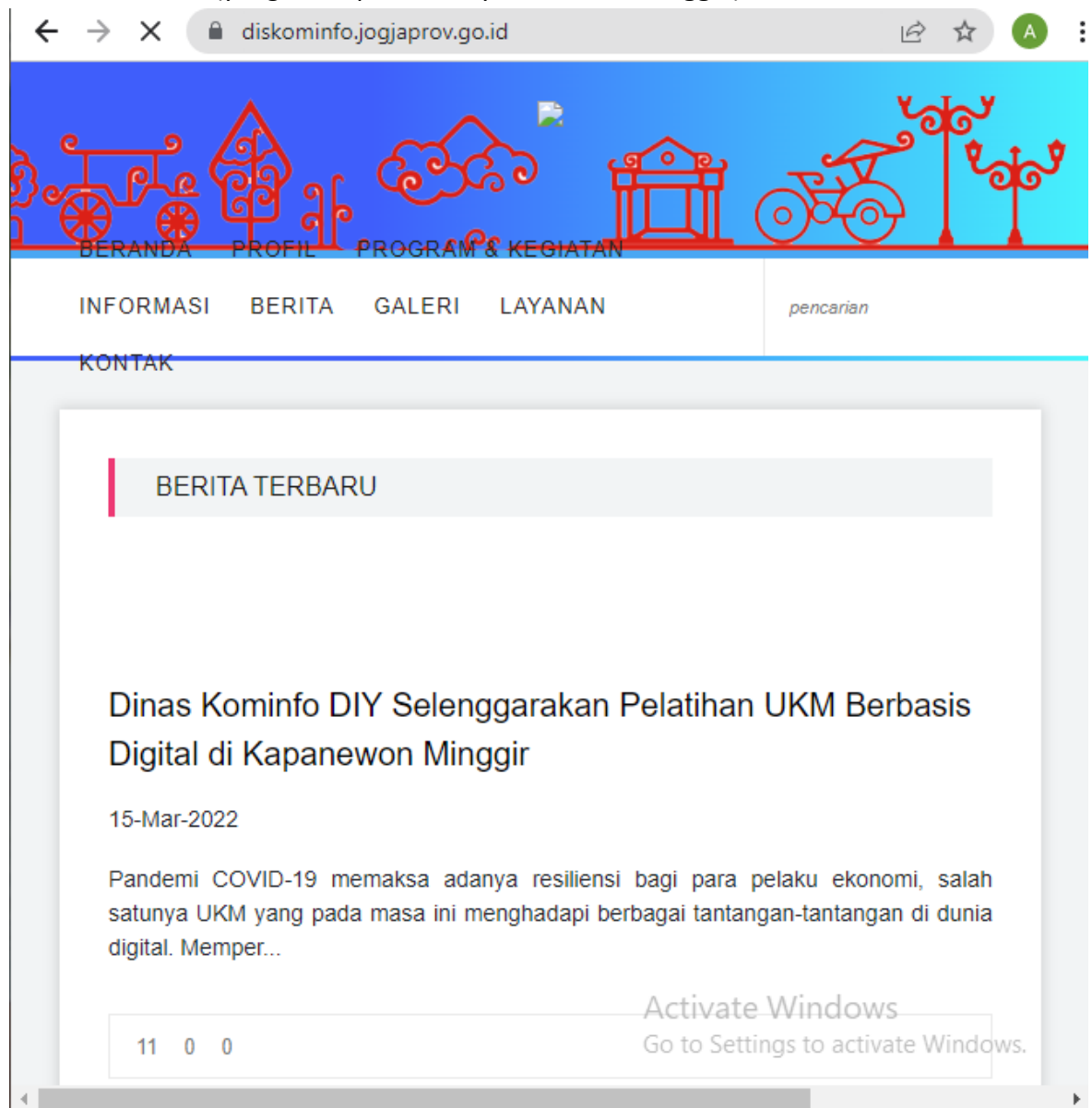
```
In [42]: from bs4 import BeautifulSoup
import requests

url = "https://www.detik.com"
#ambil isi url, masukkan ke variabel web
web = requests.get(url).text
#parsing isi variabel web
s = BeautifulSoup(web, "lxml")
#ambil class box cb-mostpop, karena hanya ada 1, maka cukup pakai f
b = s.find('div', attrs={'class' : 'box cb-mostpop'})
#didalam class tbs, temukan semua class 'media_title'
judul = b.find_all('h3', attrs={'class' : 'media_title'})
#baca tiap bagian dari judul, dan tampilan
print("Terpopuler dari Detik.com")
for j in judul :
    print('Judul : ',j.text)
```

Terpopuler dari Detik.com

## Tugas

1. Bukalah Web <https://diskominfo.jogjaprov.go.id/>, kemudian lakukan scraping untuk “Berita Terbaru” (yang ditampilkan hanya **Judul** dan **Tanggal**).



```

In [33]: from bs4 import BeautifulSoup
dokumen = '''
<!DOCTYPE html>
<html>
<head>
    <title>Tugas Praktikum 05</title>
</head>
<body>
<h1>Scrapping Diskominfo Prof Jogja</h1>
<p class='judul'>Berita Terbaru</p>
<p class='tanggal'>20 Maret 2022</p>
<a href = 'https://diskominfo.jogjaprov.go.id/'>Diskominfo Prof Jog
</body>
</html>
'''

html_soup = BeautifulSoup(dokumen, 'html.parser')
judul = html_soup.find('p', class_='judul')
tanggal = html_soup.find('p', class_='tanggal')
print(judul)
print(tanggal)

all_paragraf = html_soup.find_all('p')
print(all_paragraf)

```

<p class="judul">Berita Terbaru</p>  
 <p class="tanggal">20 Maret 2022</p>  
 [

Uraian :

Pada langkah ini sama halnya dengan langkah 5 dimana akan mengambil data pada suatu tag <html> dengan ketentuan tertentu sehingga hasil yang ditampilkan lebih spesifik. Dalam source code tersebut data yang akan diambil pada laman diskominfo.jogjaprov adalah data berita berdasarkan urutan judul dan tanggal.

- Soup.a berfungsi untuk mengambil data yang memiliki tag a sedangkan soup.a.text berfungsi untuk mengambil data text yang berada didalam tag <a>.
- Soup\_find ('p') berfungsi untuk digunakan untuk mengambil semua data yang memiliki tag <p>.
- Kemudian soup.find\_all('p') berfungsi untuk mencetak semua data yang ada pada ketentuan perintah dari soup\_find('p').



## **BAB II**

### **KESIMPULAN**

Pada praktikum ini, dapat saya simpulkan bahwa penggunaan web scrapping maupun web crawling sangat memudahkan kita pada saat ingin mengambil/menganalisa data yang ada pada internet maupun sosial media, tanpa kita perlu membaca berulang kali dan menganalisis data tersebut secara manual. Dan juga bahasa pemograman yang digunakan untuk scrapping website sangat mudah dimengerti.

# **BAB III**

## **DAFTAR PUSTAKA**

<https://ngalup.co/articles/pengertian-teknik-manfaat-kendala-web-scraping/>

<https://www.dqlab.id/teknik-pengumpulan-data-sekunder-dengan-web-crawling>