

Advanced Scraping Data		
Praktikan	Aslab	
Nama: xxxx	Annur Hangga Prihadi	065001800028
Nim: xxxx	Faiz Kumara	065001800003

PRAKTIKUM 4

DATA SAINS DAN ANALITIK

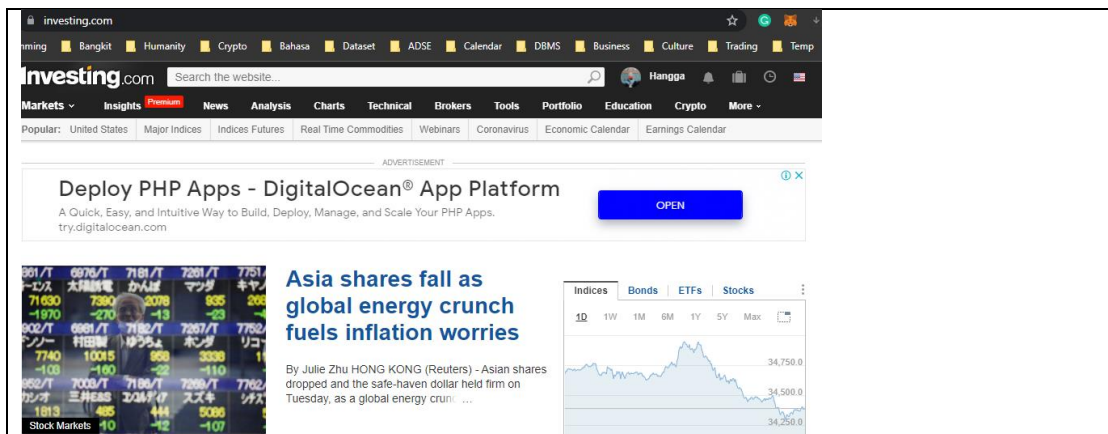
Topik pertemuan praktikum ke-empat adalah mengetahui mengambil historical data Bitcoin yang berada di portal website Investing.

SOURCE CODE 1: <https://github.com/hanggaa/PrakDSDA/blob/main/Prak4.ipynb>

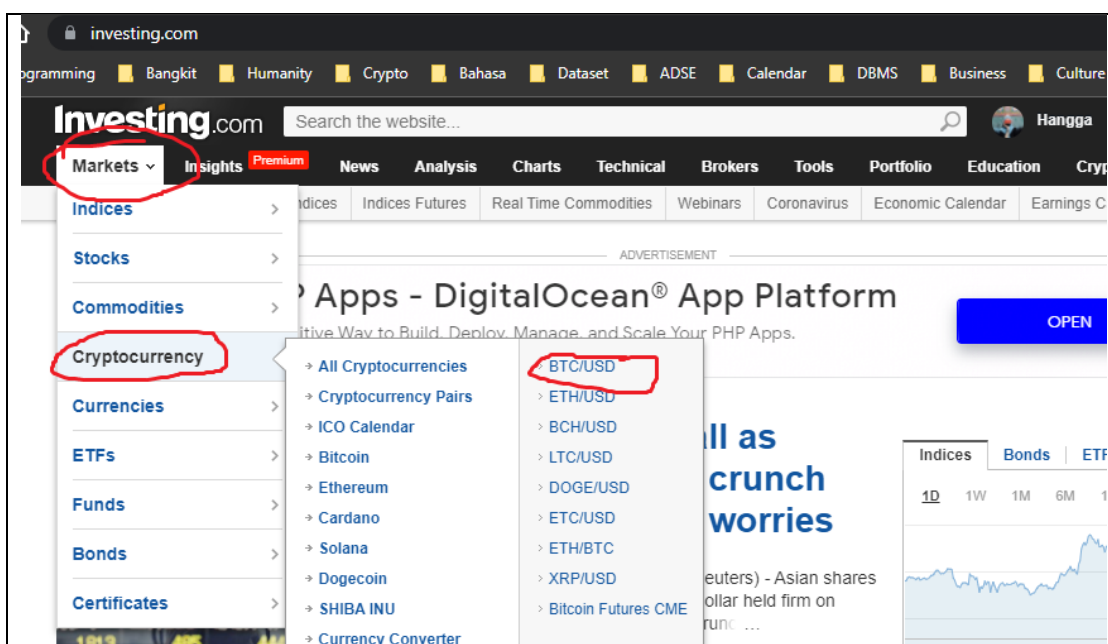
SOURCE CODE 2: <https://github.com/hanggaa/PrakDSDA/blob/main/Prak4.2.ipynb>

Latihan 1

1. Buka website Investing.com



2. Cari data Bitcoin dengan langkah berikut (Pastikan datanya adalah BTC/USD)



3. Pastikan data Bitcoin dari Exchange Bitfinex lalu klik Historical Data

BTC/USD - Bitcoin US Dollar

Bitfinex

↑ 56,507.0 +727.0 (+1.30%)

23:09:59 - Real-time Data. (Disclaimer)

Volume: 7,667 | Bid/Ask: 56,508.0 / 56,509.0 | Day's Range: 55,761.0 - 57,821.0

General Chart News & Analysis Forum Technical

Overview **Historical Data** Related Instruments Currency Converter

4. Klik kanan > Inspect Element pada browser

Dimensions: Responsive 1039 x 742 66% No throttling

Investing.com

General Chart News & Analysis Forum Technical

Overview Historical Data Related Instruments Currency Converter

BTC/USD Bitfinex Historical Data

Time Frame: Daily

Date	Price	Open	High	Low	Vol.	Change %
Oct 12, 2021	56,446.6	57,468.0	57,493.0	56,443.0	7.67K	-1.75%
Oct 11, 2021	57,468.0	54,701.0	57,802.5	54,430.0	7.51K	5.00%
Oct 10, 2021	54,701.0	54,978.0	56,414.0	54,173.0	5.84K	-0.50%
Oct 09, 2021	54,978.0	53,925.0	55,443.7	53,711.0	2.44K	1.95%
Oct 08, 2021	53,925.0	53,790.0	56,000.0	53,649.0	5.49K	0.25%
Oct 07, 2021	53,790.0	55,391.0	55,399.0	53,444.0	4.89K	-2.79%
Oct 06, 2021	55,391.0	51,508.0	55,724.0	50,448.0	10.52K	7.42%
Oct 05, 2021	51,508.0	49,232.0	51,891.1	49,065.0	6.68K	4.95%
Oct 04, 2021	49,232.0	48,243.0	49,503.0	46,961.0	6.19K	2.05%
Oct 03, 2021	48,243.0	47,875.0	49,187.0	47,124.0	3.80K	1.19%
Oct 02, 2021	47,875.0	48,181.0	48,330.0	47,459.0	2.06K	-1.00%
Oct 01, 2021	48,181.0	43,830.0	48,454.0	43,311.0	8.32K	9.87%
Sep 30, 2021	43,830.0	41,551.0	44,107.0	41,432.0	5.11K	5.49%
Sep 29, 2021	41,551.0	41,568.0	42,581.0	40,811.0	4.39K	1.19%
Sep 28, 2021	41,568.0	40,191.0	42,777.0	40,928.0	6.15K	-2.67%
Sep 27, 2021	40,191.0	43,220.5	44,354.0	42,144.0	4.26K	-2.29%
Sep 26, 2021	43,220.5	42,711.0	43,931.0	40,830.0	5.47K	1.10%
Sep 25, 2021	42,711.0	42,454.0	43,677.0	41,731.0	2.96K	-0.54%

5. Cari lalu pilih tab Network pada bagian Inspect

Elements Console Sources

Network

Performance

Memory

Application

Security

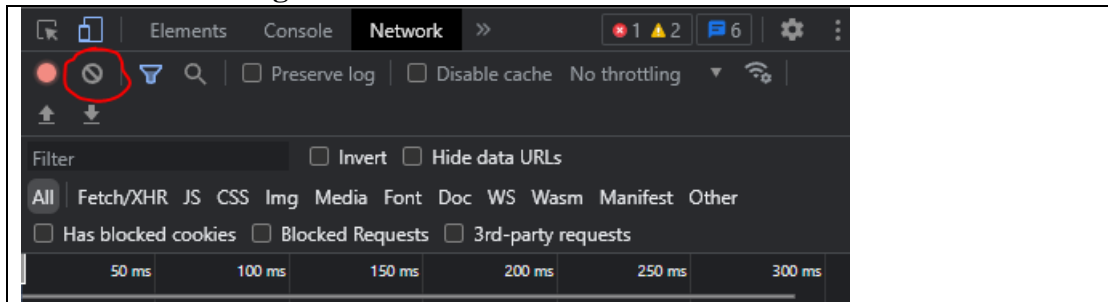
Lighthouse

... section#leftColumn div#column-content.historicalDataTimeRange div.text_align_b

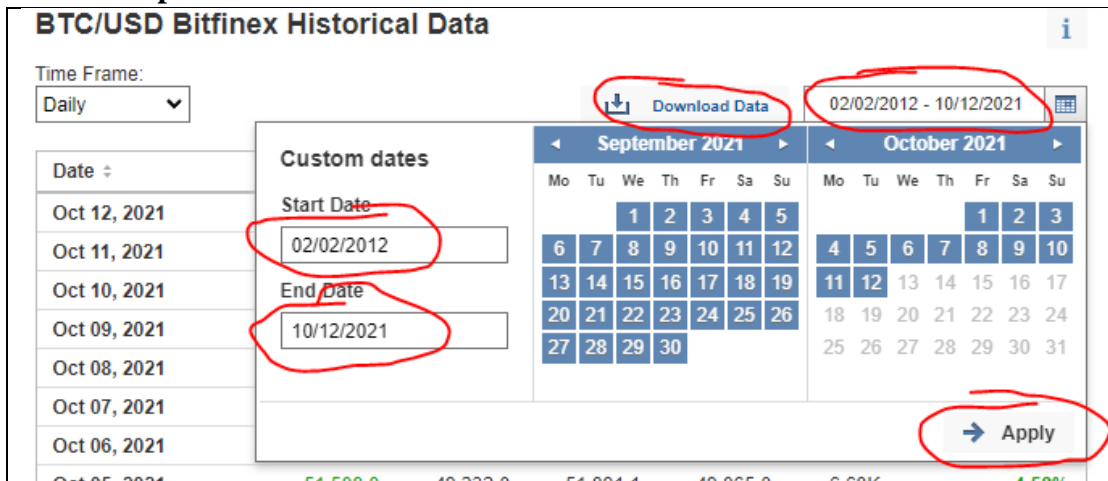
Styles Computed Layout Event Listeners DOM Breakpoints Properties Ac

Filter :hov :cls

6. Bersihkan data dengan klik button clear



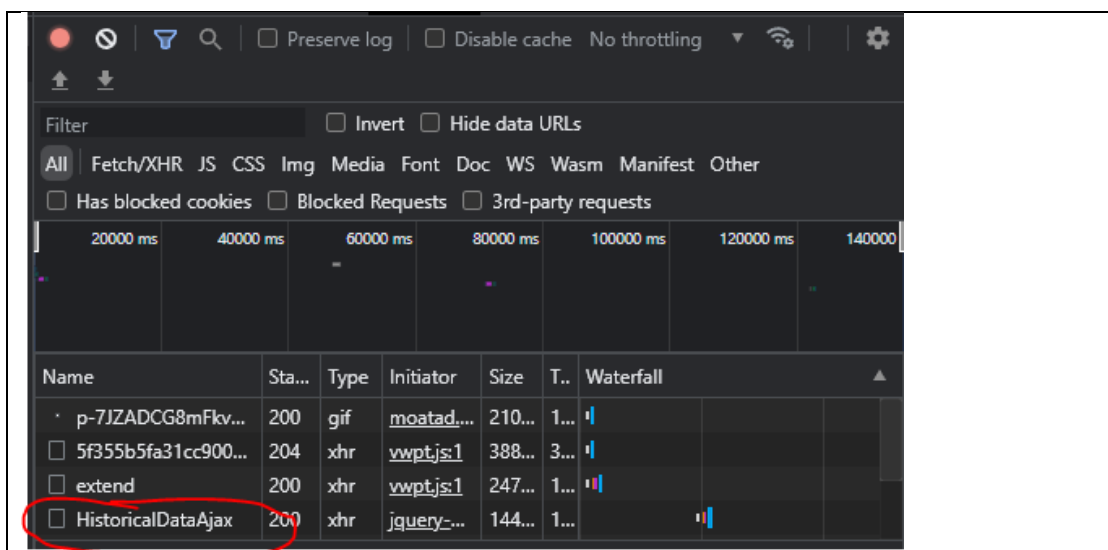
7. Ubah data pada Historical Data Bitcoin lalu download data



8. Pastikan data sudah terunduh



9. Kembali ke browser lalu cari file HistoricalDataAjax kemudian klik file tersebut



10. Catat beberapa komponen penting seperti

- a. Request URL
- b. User-agent
- c. X-requested-with
- d. Curr_id, smlID, header

Ke dalam Notepad

```
Request URL: https://www.investing.com/instruments/HistoricalDataAjax
x

user-agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Mobile Safari/537.36
x-requested-with: XMLHttpRequest

curr_id: 945629
smlID: 145284
header: BTC/USD Bitfinex Historical Data

Request URL: https://www.investing.com/instruments/HistoricalDataAjax
user-agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Mobile Safari/537.36
x-requested-with: XMLHttpRequest
curr_id: 945629
smlID: 145284
header: BTC/USD Bitfinex Historical Data
```

11. Masukkan elemen nomor 10 ke dalam source code kalian

```
user-agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Mobile Safari/537.36
x-requested-with: XMLHttpRequest

urlheader = {
    "User-Agent": "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Mobile Safari/537.36",
    "X-Requested-With": "XMLHttpRequest"
}

Request URL: https://www.investing.com/instruments/HistoricalDataAjax
x

filename=['D:\BTC_USD Bitfinex Historical Data.csv'] #Pastikan sesuai
url = "https://www.investing.com/instruments/HistoricalDataAjax"
```

```
curr_id: 945629
smlID: 145284
header: BTC/USD Bitfinex Historical Data

if (each == 'D:\BTC_USD Bitfinex Historical Data.csv'): #Pastik
    header="BTC/USD Bitfinex Historical Data"
    curr_id="945629"
    smlID= "145284"
    column_name='BTC Price'
```

12. Install library yang dibutuhkan

```
In [1]: import sys
!{sys.executable} -m pip install bs4

Requirement already satisfied: bs4 in c:\users\hangg\anacond
Requirement already satisfied: beautifulsoup4 in c:\users\ha
Requirement already satisfied: soupsieve>1.2 in c:\users\han
```

13. Jalankan source code web scrape

```
Source Code Web Scrape

In [2]: import requests
from bs4 import BeautifulSoup
import pandas as pd
import csv
from datetime import date
import datetime
import numpy as np
from matplotlib import pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import statsmodels.stats.api
import statsmodels.formula.api

today = date.today()
today=date.strftime(today,'%d/%m/%Y')
urlheader = {
    "User-Agent": "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.91 Mobile Safari/537.36"
    "X-Requested-With": "XMLHttpRequest"
}
```

14. Menampilkan n data sementara

```
In [3]: df.head(4)

Out[3]:
```

	Date	Price	Open	High	Low	Vol.	Change %
0	12-Oct-21	56,602.0	57,468.0	57,493.0	56,443.0	7.44K	-1.51%
1	11-Oct-21	57,468.0	54,701.0	57,802.5	54,430.0	7.51K	5.06%
2	10-Oct-21	54,701.0	54,978.0	56,414.0	54,173.0	5.84K	-0.50%
3	09-Oct-21	54,978.0	53,925.0	55,443.7	53,711.0	2.44K	1.95%

15. Menampilkan deksripsi data frame sementara

```
In [4]: df.describe()
```

```
Out[4]:
```

	Date	Price	Open	High	Low	Vol.	Change %
count	3537	3537	3537	3537	3537	3537	3537
unique	3537	3055	3044	3003	3019	1922	1323
top	31-Jan-20	5.1	5.1	5.1	5.1	-	0.00%
freq	1	26	26	26	26	660	135

16. Menampilkan dimensi data frame sementara

```
In [5]: df.shape
```

```
Out[5]: (3537, 7)
```

17. Menampilkan tipe data kolom data frame sementara

```
In [6]: df.dtypes
```

```
Out[6]: Date      object
Price      object
Open       object
High       object
Low        object
Vol.       object
Change %   object
dtype: object
```

18. Menampilkan informasi kolom data frame sementara

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3537 entries, 0 to 3536
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        3537 non-null  object
1   Price       3537 non-null  object
2   Open        3537 non-null  object
3   High        3537 non-null  object
4   Low         3537 non-null  object
5   Vol.        3537 non-null  object
6   Change %    3537 non-null  object
dtypes: object(7)
memory usage: 221.1+ KB
```

19. Ubah tipe data kolom yang diperlukan

```
In [8]: df['Date'] = pd.to_datetime(df['Date'])
df['Price'] = df['Price'].str.replace(',', '').astype(float)
df['Open'] = df['Open'].str.replace(',', '').astype(float)
df['High'] = df['High'].str.replace(',', '').astype(float)
df['Low'] = df['Low'].str.replace(',', '').astype(float)
```

20. Menampilkan n data sebenarnya

```
In [9]: df.head(4)
```

Out[9]:

	Date	Price	Open	High	Low	Vol.	Change %
0	2021-10-12	56602.0	57468.0	57493.0	56443.0	7.44K	-1.51%
1	2021-10-11	57468.0	54701.0	57802.5	54430.0	7.51K	5.06%
2	2021-10-10	54701.0	54978.0	56414.0	54173.0	5.84K	-0.50%
3	2021-10-09	54978.0	53925.0	55443.7	53711.0	2.44K	1.95%

21. Menampilkan informasi data frame yang sebenarnya

```
In [10]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3537 entries, 0 to 3536
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        3537 non-null   datetime64[ns]
1   Price       3537 non-null   float64
2   Open       3537 non-null   float64
3   High       3537 non-null   float64
4   Low        3537 non-null   float64
5   Vol.       3537 non-null   object
6   Change %   3537 non-null   object
dtypes: datetime64[ns](1), float64(4), object(2)
memory usage: 221.1+ KB
```

22. Menampilkan deskripsi data frame sebenarnya

```
In [11]: df.describe()
```

Out[11]:

	Price	Open	High	Low
count	3537.000000	3537.000000	3537.000000	3537.000000
mean	6882.641165	6867.310913	7060.097653	6666.455160
std	12351.245934	12323.964251	12664.668783	11974.408705
min	4.200000	4.200000	4.200000	4.200000
25%	273.300000	273.300000	279.600000	267.300000
50%	872.600000	871.400000	901.500000	833.700000
75%	8230.100000	8230.000000	8423.500000	8019.700000
max	63518.000000	63518.000000	64374.000000	62348.000000

23. Membuat variabel berdasarkan kelompok tanggal

```
In [12]: avgp = df.groupby('Date')['Price'].mean()
```

24. Memecah variabel untuk menentukan index dan values yang digunakan

```
In [13]: x = avgp.index  
y = avgp.values
```

25. Membuat grafik pergerakan harga Bitcoin

```
In [14]: plt.figure(figsize=(16,8))  
plt.scatter(x,y)  
plt.plot(x,y)  
plt.xlabel('Tahun')  
plt.ylabel('Pergerakan Harga')  
plt.title('Pergerakan Harga Bitcoin')  
  
Out[14]: Text(0.5, 1.0, 'Pergerakan Harga Bitcoin')
```



26. Mencari prediksi rata-rata harga tahun 2021 menggunakan regresi linier

```
In [15]: linreg = LinearRegression()  
x=np.array(x).reshape(-1,1)  
linreg.fit(x,y)  
  
Out[15]: LinearRegression()  
  
In [16]: P_2021 = np.array(2021).reshape(-1,1)  
pred_P = linreg.predict(P_2021)  
  
In [17]: print('Prediksi rata-rata harga tahun 2021 = ', pred_P.item())  
  
Prediksi rata-rata harga tahun 2021 = -134180.54578409265
```

27. Mencari ikhtisar regresi berganda dari pengaruh Harga tinggi dan Harga buka terhadap Harga

```
In [18]: regB = statsmodels.formula.api.ols('Price~High+Open', data = df).fit()  
print(regB.summary())
```

```
OLS Regression Results  
=====
```

Dep. Variable:	Price	R-squared:	0.999
Model:	OLS	Adj. R-squared:	0.999
Method:	Least Squares	F-statistic:	1.904e+06
Date:	Tue, 12 Oct 2021	Prob (F-statistic):	0.00
Time:	11:32:26	Log-Likelihood:	-25992.
No. Observations:	3537	AIC:	5.199e+04
Df Residuals:	3534	BIC:	5.201e+04
Df Model:	2		
Covariance Type:	nonrobust		

```
=====
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.7297	7.242	0.101	0.920	-13.470	14.929
High	1.1914	0.016	74.292	0.000	1.160	1.223
Open	-0.2227	0.016	-13.514	0.000	-0.255	-0.190

```
=====
```


NOTE

28. Jika menemui error seperti gambar, silakan buka source code kedua

```
ValueError: time data '12-Oct-21' does not match format '%b %d, %Y'
```

Latihan 2

1. Cari model peluang regresi berganda dari pengaruh harga tinggi dan harga rendah terhadap harga
2. Cari model peluang regresi berganda dari pengaruh harga rendah dan harga buka terhadap harga
3. Cari model peluang regresi berganda dari pengaruh harga tinggi dan harga terhadap harga buka
4. Cari model peluang regresi berganda dari pengaruh harga rendah dan harga terhadap harga buka

Lampiran Screenshot hasil 1, 2, 3, dan 4

[Input screenshot disini](#)

Makna dari masing-masing hasil di atas!

[Ketik makna disini](#)