

# Web Scraping

By  
Hanif Muhammad Mufid

---

Data Analytics Capstone

---

# Latar Belakang

Level : Hard

Website Data Pekerjaan di Indonesia pada halaman <https://www.kalibrr.id/id-ID/job-board/te/data/1>

Mengambil data `title pekerjaan`, `lokasi pekerjaan`, `tanggal pekerjaan di post dan dealine submit permohonan`, dan `perusahaan` dari 15 halaman pertama.

Webscraping ini dilakukan untuk mendapatkan informasi terkait pekerjaan-pekerjaan yang ada dan informasi detail lainnya. Kemudian data yang didapat akan dilakukan process EDA dan Data Wrangling. Hasil akhir dari data ini akan dijadikan visualisasi dan akan diimplementasikan dengan menggunakan library Flask.

# Requesting the Data and Creating a BeautifulSoup

```
1 import requests  
2  
3 url_get = requests.get('https://www.kalibrr.id/id-ID/job-board/te/data/1')
```

Pada tahap ini dilakukan requesting web dari website dengan get method

```
1 from bs4 import BeautifulSoup  
2  
3 soup = BeautifulSoup(url_get.content,"html.parser")
```

Kemudian, membuat Beautiful Soup object agar dapat mengeksplor dan mengambil data dari website

# Finding the right key to scrap the data & Extracting the right information

```
halaman = 15

title = []
lokasi = []
post_dl = []
perusahaan = []
```

Membuat variabel list, yang nantinya akan dimasukan data-data yang akan diambil berdasarkan kategorinya.

# Finding the right key to scrap the data & Extracting the right information

```
for i in range(1, halaman + 1):
    print(f'Proses Halaman {i}')
    url_get = requests.get(f"https://www.kalibrr.id/id-ID/job-board/te/data/{i}")
    soup = BeautifulSoup(url_get.content, "html.parser")
    title_0 = soup.find_all('h2', attrs = {'class' : 'k-text-xl k-font-medium'})
    title_1 = BeautifulSoup(str(title_0), 'html.parser')
    for item in title_1.find_all('a', attrs={'class':'k-text-primary-color'}):
        title.append(item.text)
    perusahaan_0 = soup.find_all('span', attrs = {'class' : 'k-inline-flex k-items-center k-mb-1'})
    perusahaan_1 = BeautifulSoup(str(perusahaan_0), 'html.parser')
    for item in perusahaan_1.find_all('a', attrs={'class':'k-text-subdued'}):
        perusahaan.append(item.text)
    for item in soup.find_all('a', attrs={'class':'k-text-subdued k-block'}):
        lokasi.append(item.text)
    for item in soup.find_all('span', attrs={'class':'k-block k-mb-1'}):
        post_dl.append(item.text)
print('DONE')
```

Proses Halaman 1  
Proses Halaman 2  
Proses Halaman 3  
Proses Halaman 4  
Proses Halaman 5  
Proses Halaman 6  
Proses Halaman 7  
Proses Halaman 8  
Proses Halaman 9  
Proses Halaman 10  
Proses Halaman 11  
Proses Halaman 12  
Proses Halaman 13  
Proses Halaman 14  
Proses Halaman 15  
DONE

Disini, sudah mencoba menemukan kunci yang tepat untuk setiap data nya, dan akan dilakukan pengulangan sebanyak 15 kali, untuk mendapatkan data pada 15 halaman pertama.

# Finding the right key to scrap the data & Extracting the right information

```
1 len(title), len(lokasi), len(post_dl), len(perusahaan)  
(225, 225, 225, 225)
```

Memastikan apakah data yang didapat sudah sesuai, yaitu 15 data pekerjaan setiap halaman dan halaman yang diambil berjumlah 15. Maka data seharusnya berjumlah  $15 \times 15 = 225$  data setiap kolom nya.

# Creating data frame & Data wrangling

```
1 import pandas as pd  
2  
3 df = pd.DataFrame({"Title Pekerjaan":title,"Lokasi": lokasi, "Post dan Deadline":post_dl, "Perusahaan": perusahaan})  
4 df.head()
```

	Title Pekerjaan	Lokasi	Post dan Deadline	Perusahaan
0	DevOps and Data Engineer	Tangerang Selatan, Indonesia	Posted 8 days ago • Apply before 13 May	Mobius Digital
1	Data Quality Analyst	Central Jakarta City, Indonesia	Posted 9 days ago • Apply before 1 May	Astra Financial
2	Project Manager	Jakarta, Indonesia	Posted 9 days ago • Apply before 29 Jun	PGI Data
3	Network Security Engineer	Jakarta, Indonesia	Posted 7 days ago • Apply before 19 Apr	PGI Data
4	iOS Developer	Jakarta Selatan, Indonesia	Posted 2 days ago • Apply before 9 May	PhinCon

Kemudian memasukan variabel list yang sudah berisi data tadi kedalam DataFrame

# Creating data frame & Data wrangling

```
1 df.isna().sum()
```

Title Pekerjaan	0
Lokasi	0
Post dan Deadline	0
Perusahaan	0
<b>dtype:</b> int64	

```
1 df.dtypes
```

Title Pekerjaan	object
Lokasi	object
Post dan Deadline	object
Perusahaan	object
<b>dtype:</b> object	

```
1 df.nunique()
```

Title Pekerjaan	145
Lokasi	35
Post dan Deadline	113
Perusahaan	87
<b>dtype:</b> int64	

Mengecek apakah terdapat data yang kosong, dan juga mengecek tipe data setiap kolomnya.

# Creating data frame & Data wrangling

1 df['Lokasi'].value_counts()	
South Jakarta, Indonesia	36
Jakarta, Indonesia	19
Tangerang, Indonesia	19
Jakarta Selatan, Indonesia	18
Jakarta Pusat, Indonesia	17
Central Jakarta, Indonesia	14
North Jakarta, Indonesia	11
West Jakarta, Indonesia	10
South Tangerang, Indonesia	9
Kota Jakarta Selatan, Indonesia	9
Jakarta Barat, Indonesia	8
Surabaya, Indonesia	7
Central Jakarta City, Indonesia	5
Sleman, Indonesia	4
Bandung Kota, Indonesia	3
Kota Jakarta Pusat, Indonesia	3
Tangerang Selatan, Indonesia	3

Malang Kota, Indonesia	3
Yogyakarta, Indonesia	2
Jakarta Utara, Indonesia	2
South Jakarta City, Indonesia	2
Makassar, Indonesia	2
Banyuwangi, Indonesia	2
Bandung Kabupaten, Indonesia	2
Tangerang Kabupaten, Indonesia	2
Banjarmasin, Indonesia	2
East Jakarta, Indonesia	2
Tangerang Kota, Indonesia	2
Kota Jakarta Barat, Indonesia	1
Depok City, Indonesia	1
Jember, Indonesia	1
Tabanan, Indonesia	1
West Lombok, Indonesia	1
Jembrana, Indonesia	1
Sukabumi City, Indonesia	1

Pada kolom lokasi, kota perusahaan berada. Terlihat banyak kota yang sama namun memiliki redaksi yang berbeda.

# Creating data frame & Data wrangling

```
1 df['Lokasi'] = df['Lokasi'].replace(', Indonesia', '', regex = True)
```

```
df['Lokasi'] = df['Lokasi'].replace(['West Jakarta', 'Kota Jakarta Barat'], 'Jakarta Barat', regex = True)
df['Lokasi'] = df['Lokasi'].replace(['Central Jakarta', 'Central Jakarta City', 'Kota Jakarta Pusat', 'Jakarta Pusat City'],
                                    'Jakarta Pusat', regex = True)
df['Lokasi'] = df['Lokasi'].replace(['North Jakarta'], 'Jakarta Utara', regex = True)
df['Lokasi'] = df['Lokasi'].replace(['East Jakarta'], 'Jakarta Timur', regex = True)
df['Lokasi'] = df['Lokasi'].replace(['South Jakarta', 'South Jakarta City', 'Kota Jakarta Selatan', 'Jakarta Selatan City'],
                                    'Jakarta Selatan', regex = True)
df['Lokasi'] = df['Lokasi'].replace(['South Tangerang'], 'Tangerang Selatan', regex = True)
df['Lokasi'] = df['Lokasi'].replace(['Tangerang Kota'], 'Tangerang', regex = True)
```

Karena semua perusahaan berada di Indonesia, maka akan dihilangkan untuk mempermudah membaca data.  
Kemudian melakukan penyamaan redaksi terhadap lokasi-lokasi yang sama

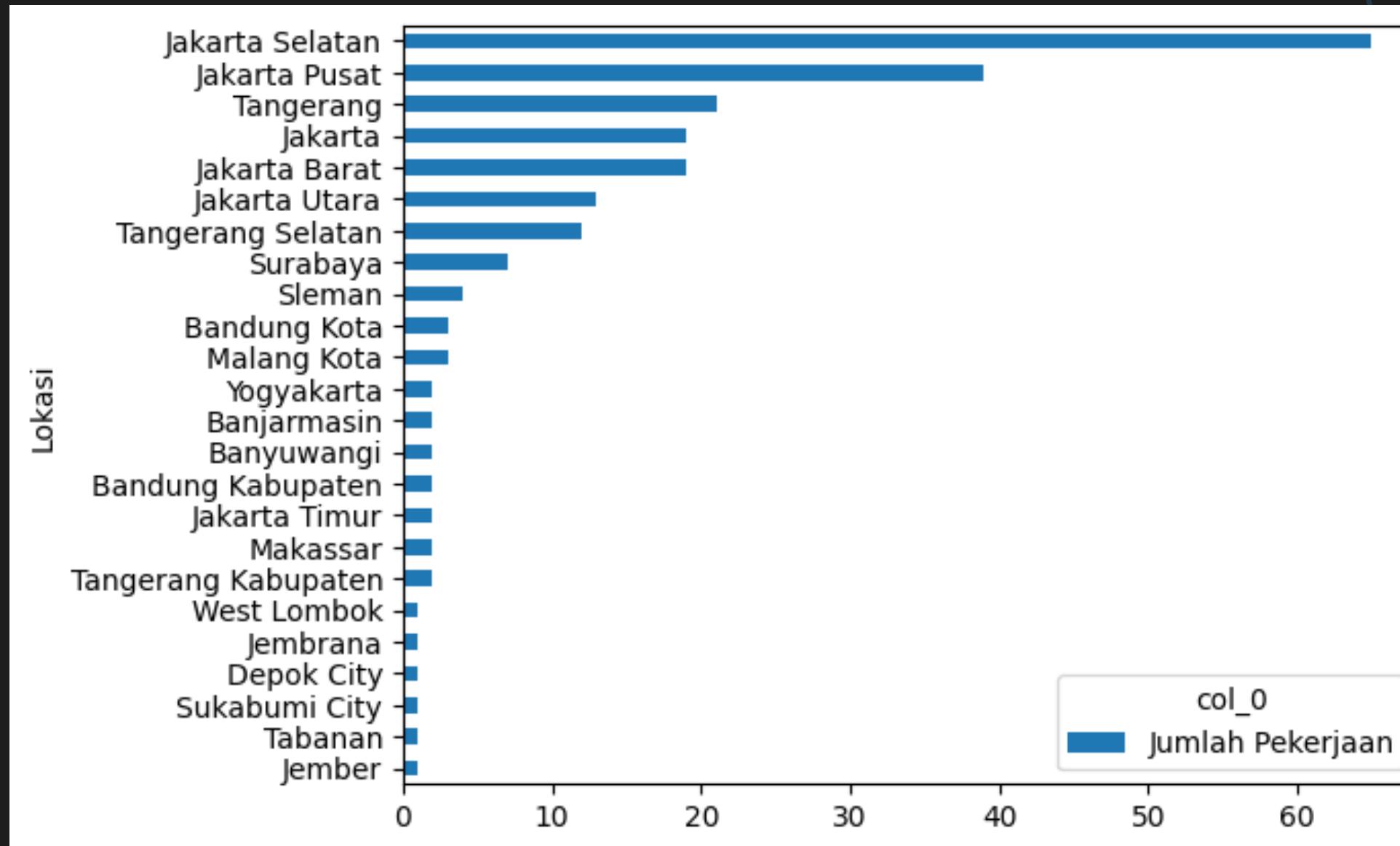
# Creating data frame & Data wrangling

```
1 df.head()
```

	Title Pekerjaan	Lokasi	Post dan Deadline	Perusahaan
0	DevOps and Data Engineer	Tangerang Selatan	Posted 8 days ago • Apply before 13 May	Mobius Digital
1	Data Quality Analyst	Jakarta Pusat	Posted 9 days ago • Apply before 1 May	Astra Financial
2	Project Manager	Jakarta	Posted 9 days ago • Apply before 29 Jun	PGI Data
3	Network Security Engineer	Jakarta	Posted 7 days ago • Apply before 19 Apr	PGI Data
4	iOS Developer	Jakarta Selatan	Posted 2 days ago • Apply before 9 May	PhinCon

Tampak Hasil akhir dari data wrangling

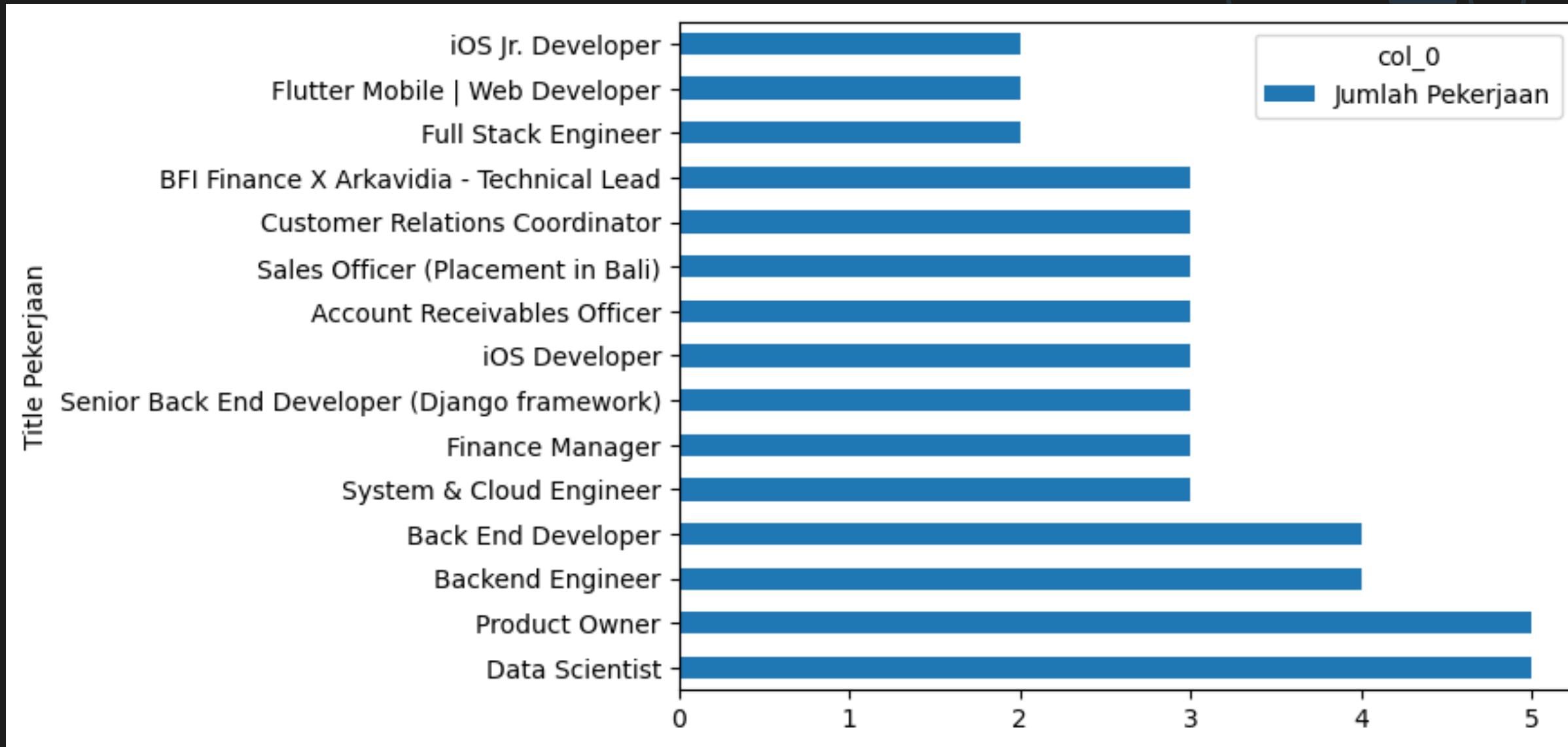
# Data Visualisation



Insight :

- Kota Jakarta Selatan mempunyai jumlah pekerjaan paling banyak dari kota yang lain.
- 5 kota dengan jumlah pekerjaan paling banyak didominasi oleh kota di provinsi DKI Jakarta.

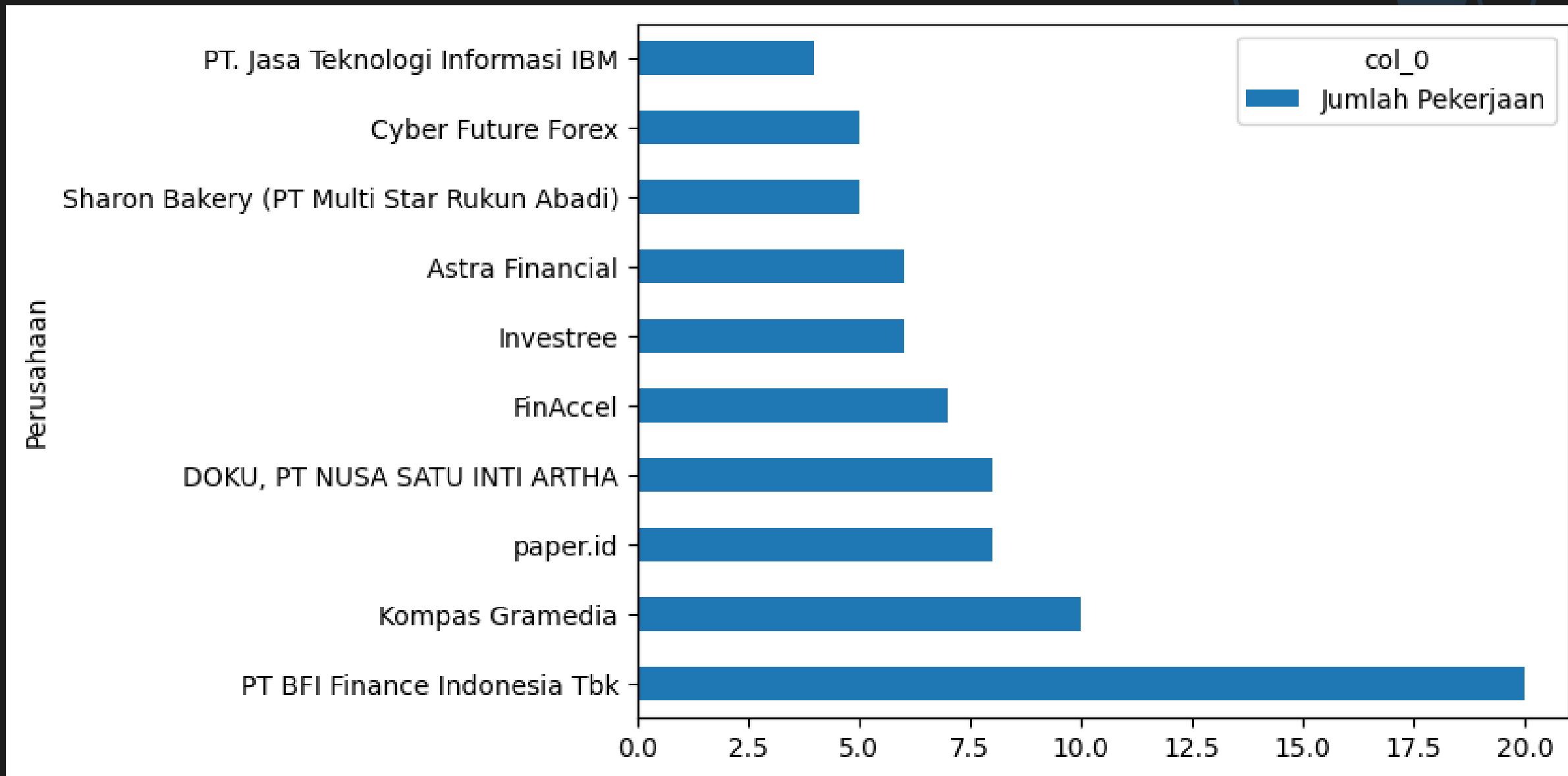
# Data Visualisation



Insight :

- Pekerjaan paling dicari adalah Data Scientist dan Product Owner
- 15 pekerjaan paling banyak didominasi oleh pekerjaan yang berkaitan dengan IT (Web, Database, Programming)

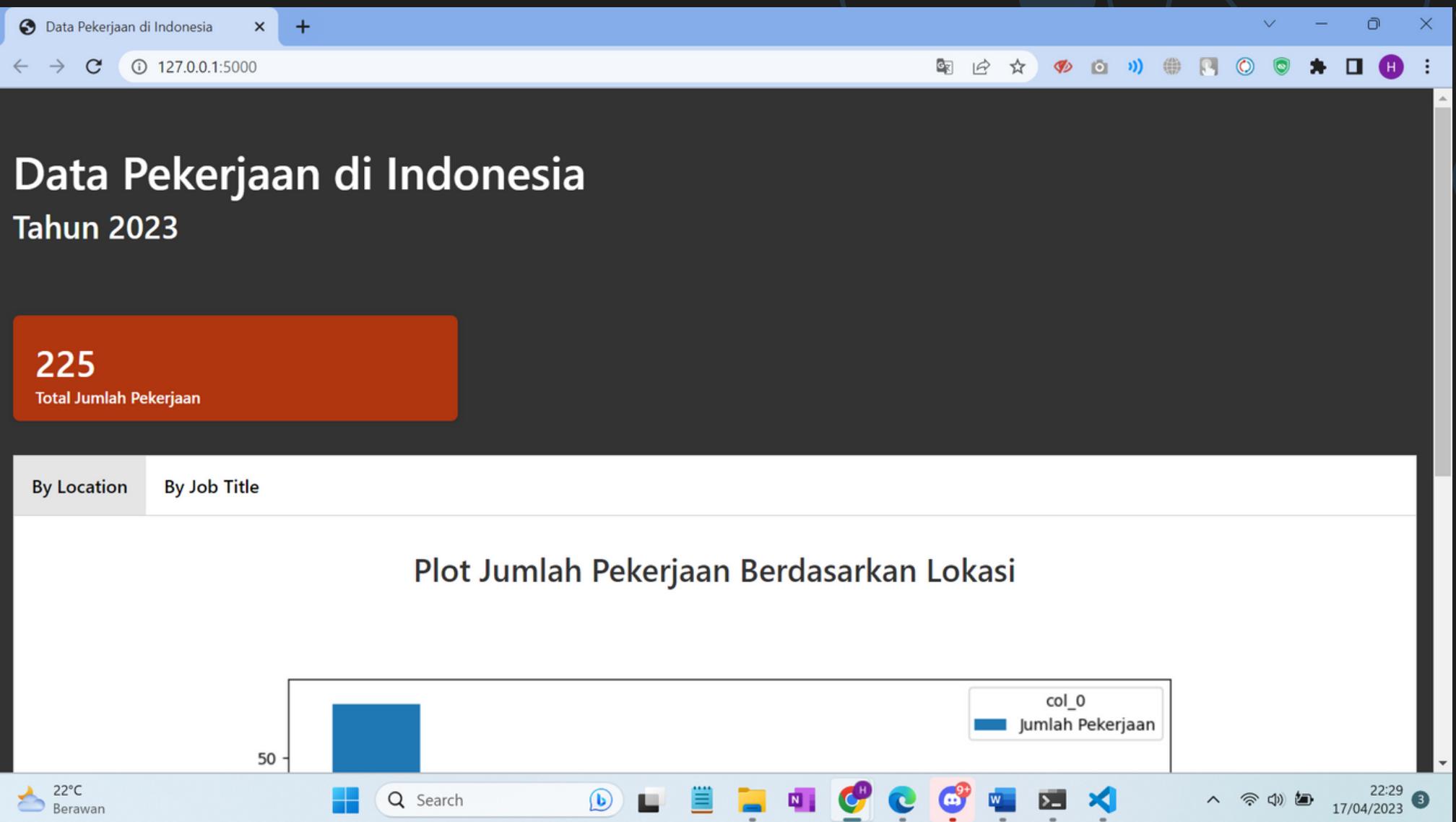
# Data Visualisation



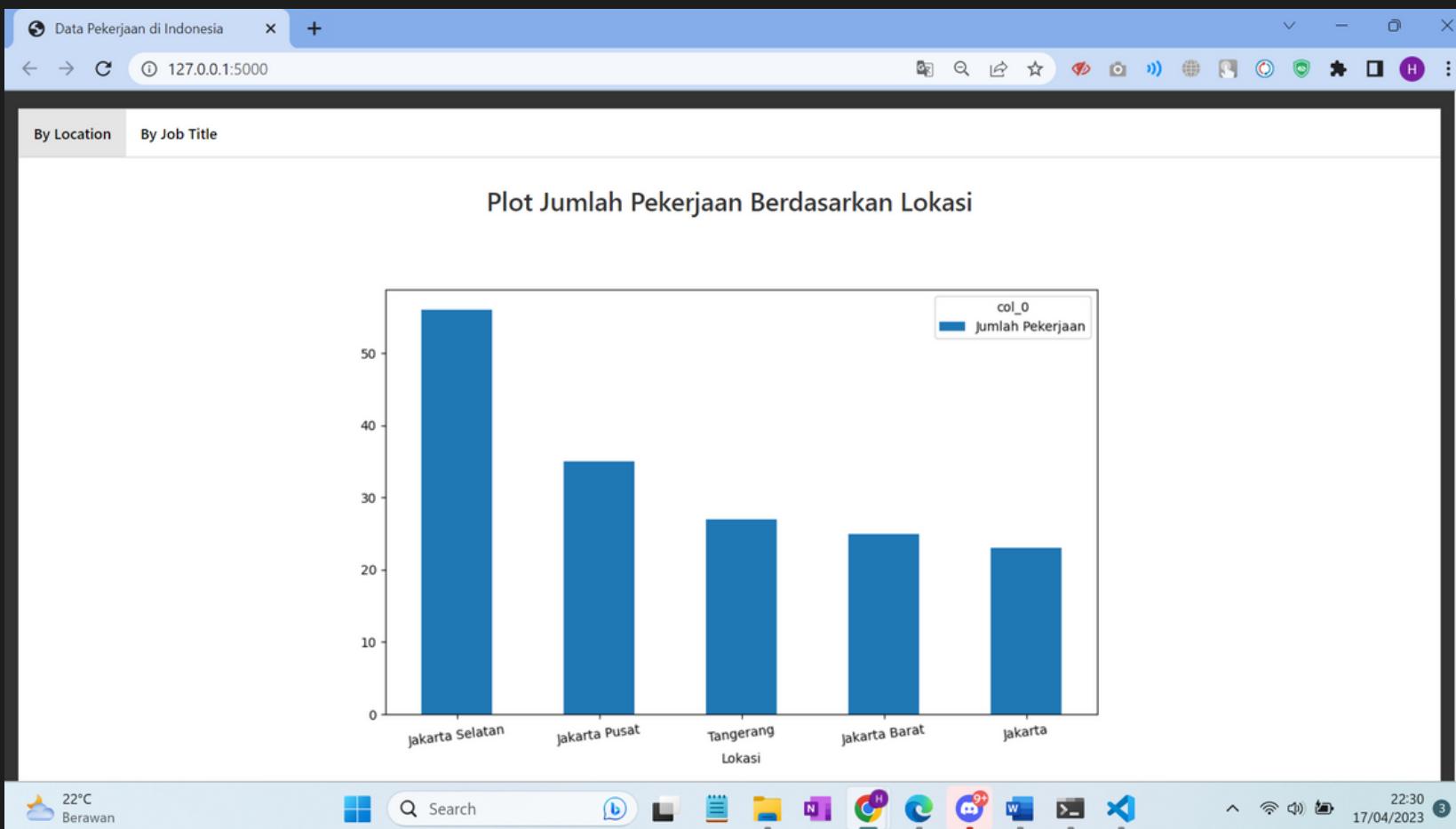
Insight :

- PT BFI Finance Indonesia Tbk menjadi perusahaan dengan jumlah pekerjaan terbanyak
- Antara PT BFI Finance Indonesia Tbk dan perusahaan lain mempunyai perbedaan signifikan terkait jumlah lowongan pekerjaan

# Implement It at the Webapps



# Implement It at the Webapps



# TERIMA KASIH