LAPORAN TUGAS BESAR 2 PENGENALAN KOMPUTASI ANALISIS DATA

Disusun untuk Memenuhi Tugas Besar 2 Mata Kuliah Pengenalan Komputasi KU1102
Dosen Pengampu:
Dr. Fadhil Hidayat, S.Kom., M.T.

Kelompok 13
K-19 STEI

1. Karol Yangqian Poetracahya (19623206)
2. Nayaka Ghana Subrata (19623031)
3. Julian Benedict (16523178)
4. Dimas Anggiat (16523052)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
DESEMBER 2023

DAFTAR ISI

BAB I
DESKRIPSI DATA DAN *FILE*

Dataset yang digunakan dalam tugas ini bernama Salaries dengan nama *file* "salaries.csv". *Dataset* ini berisi mengenai pendapatan global per tahun dari orang yang bekerja di bidang *artificial intelligence* (AI), *machine learning* (ML), dan *data science* (DS). Selain itu, di dalamnya juga disajikan data tahun gaji tersebut diberikan, tingkat pengalaman, tipe pekerjaan, jabatan, mata uang pendapatan, dan berbagai atribut mengenai lokasi pekerjaan tersebut. Dengan rincian atribut tersebut, informasi yang ingin diketahui dari *dataset* ini adalah hubungan pekerjaan dengan gaji per tahun dari masing-masing pekerjaan di bidang AI, ML, dan DS.

Data ini memiliki format *comma-separated values* (CSV). Seperti namanya, CSV adalah format *file* yang menyimpan data dengan pemisah atau *delimiter* yakni sebuah koma (,) untuk membedakan kolom atau atribut tiap data. Baris pertama merupakan judul kolom dan kolom pertama menunjukkan index. Seluruh baris dan kolom berikutnya merupakan data yang berpadanan dengan index dan atribut yang dimilikinya. *Dataset* ini didapat dari situs web "ai-jobs.net" yang dirujuk melalui Kaggle, sebuah situs web yang menyajikan pembelajaran mengenai *data science* dan *machine learning* serta memberikan wadah bagi komunitas dari seluruh dunia untuk belajar bersama mengenai bidang ini. Data berukuran 496 kb (kilobyte) ini memiliki dimensi yakni 8805 baris dan 11 kolom.

Tugas ini dikerjakan dengan Google Colab, sebuah aplikasi web untuk mengedit dan menjalankan *file* Jupyter Notebook (ditandai dengan ekstensi ".ipynb") yang menggunakan bahasa pemrograman Python. Pertama-tama, *directory* yang berbasis Google Drive diatur terlebih dahulu dengan *library* "google.colab.drive". Kemudian, untuk melakukan analisis data, digunakan *library* Pandas dan Matplotlib. Pandas digunakan untuk membaca *file* "salaries.csv" dan menyimpannya sebagai sebuah DataFrame. Dalam bahasa pemrograman, aktivitas di atas ditulis seperti pada gambar 1.1.

```
[ ]  # Set-up google colab

     from google.colab import drive
```

```
[ ]  # Set-up dataframe

     drive.mount('/content/drive')

     Mounted at /content/drive
```

```
[ ]  # Import pandas and matplotlib library

     import pandas as pd
     import matplotlib.pyplot as plt
     import matplotlib as mpl

     # Load data menggunakan pandas (program akan membaca data sesuai dengan path yang ditargetkan)
     df = pd.read_csv('/content/drive/MyDrive/Tubes/salaries.csv')

     print("Data: ")
     df
```

Gambar 1.1 *Set-up library* dan *loading* data sebagai DataFrame

Ada beberapa informasi terkait data yang dapat diakses menggunakan *tools* yang telah disebutkan. Informasi tersebut meliputi format *file* (Gambar 1.2), ukuran *file* (Gambar 1.3), sumber data atau *path* (Gambar 1.4), jumlah baris dan kolom (Gambar 1.5), informasi tambahan (Gambar 1.6), dan tabel data (Gambar 1.7).

Format File

```
# import os library
import os

# Mencari format file dengan method .path.splitext()
path, format = os.path.splitext('/content/drive/MyDrive/Tubes/salaries.csv')

# Output
print(f"format file: {format}")
```

```
format file: .csv
```

Gambar 1.2 Format *file*

Ukuran File

```
# Import os library
import os

# Mencari ukuran file dengan method .path.getsize()
size = os.path.getsize('/content/drive/MyDrive/Tubes/salaries.csv')

# Output
print(f"File size: {size} bytes")
```

```
File size: 496758 bytes
```

Gambar 1.3 Ukuran *file*

Sumber Data (Path)

```
[ ]  # import os library
     import os

     # Mencari format file dengan method .path.splitext()
     path, format = os.path.splitext('/content/drive/MyDrive/Tubes/salaries.csv')

     # Output
     print(f"Sumber data: {path}")
```

```
Sumber data: /content/drive/MyDrive/Tubes/salaries
```

Gambar 1.4 Sumber data (*path*)

Mencari Jumlah Baris dan Kolom

```
[ ]  # Metode pertama, menggunakan method .axes[]

     baris = len(df.axes[0])
     kolom = len(df.axes[1])

     print(f"Jumlah baris dari data: {baris} baris")
     print(f"Jumlah kolom dari data: {kolom} kolom")
```

```
Jumlah baris dari data: 8805 baris
Jumlah kolom dari data: 11 kolom
```

```
[ ]  # Metode kedua, menggunakan method .shape

     baris, kolom = df.shape

     print(f"Jumlah baris dari data: {baris} baris")
     print(f"Jumlah kolom dari data: {kolom} kolom")
```

```
Jumlah baris dari data: 8805 baris
Jumlah kolom dari data: 11 kolom
```

Gambar 1.5 Jumlah baris dan kolom data

Informasi Tambahan Terkait Data

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8805 entries, 0 to 8804
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   work_year           8805 non-null   int64 
 1   experience_level    8805 non-null   object
 2   employment_type     8805 non-null   object
 3   job_title           8805 non-null   object
 4   salary              8805 non-null   int64 
 5   salary_currency     8805 non-null   object
 6   salary_in_usd       8805 non-null   int64 
 7   employee_residence  8805 non-null   object
 8   remote_ratio        8805 non-null   int64 
 9   company_location    8805 non-null   object
 10  company_size        8805 non-null   object
dtypes: int64(4), object(7)
memory usage: 756.8+ KB
```

Gambar 1.6 Informasi tambahan

Tabel Data

```
[ ] print("Tabel data: ")
    df
```

Tabel data:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	EX	FT	Data Science Director	212000	USD	212000	US	0	US	M
1	2023	EX	FT	Data Science Director	190000	USD	190000	US	0	US	M
2	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
3	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
4	2023	SE	FT	Machine Learning Engineer	245700	USD	245700	US	0	US	M
...	...	...	...	...	...	...	...	...	...	...	...
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L
8801	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	US	L
8802	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	US	S
8803	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	US	L
8804	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	L

8805 rows × 11 columns

Gambar 1.7 Tabel data

BAB II
KARAKTERISTIK DATA

Data yang kami ambil dan proses mengandung beberapa atribut sebagai berikut.

1. Work year: Berisi data waktu dalam satuan tahun (Categorical-Nominal)
2. Experience level: Berisi data pengalaman kerja (Categorical-Nominal)
3. Employment type: Berisi data jenis pekerjaan (Categorical-Nominal)
4. Job title: Berisi data nama pekerjaan (Categorical-Nominal)
5. Salary: Berisi pendapatan mata uang asal dalam masing-masing mata uang (Quantitative-Discrete)
6. Salary currency: Berisi nilai mata uang (Categorical-Nominal)
7. Salary in USD: Berisi pendapatan dalam mata uang US (Quantitative-Discrete)
8. Employee residence: Berisi daerah tempat tinggal pekerja (Categorical-Nominal)
9. Remote ratio: Berisi persentase dari tingkat 'remote' pekerjaan (Quantitative-Continuous)
10. Company location: Berisi lokasi perusahaan (Categorical-Nominal)
11. Company size: Berisi ukuran perusahaan (Categorical-Nominal)

2.1. Nilai Atribut dan Tipe Data

Pada tahap pertama ini adalah menentukan nilai atribut dan tipe data, tiap masing-masing data tersebut. Pada tahap ini juga hampir semua atribut memilikinya, namun ada beberapa atribut yang berubah dari nilai atribut dan tipe data menjadi range atribut dan tipe data seperti pada "Salary", "Salary in USD", "Remote ratio". Contoh nilai atribut dan tipe data pada beberapa atribut yang telah ada. Pada bagian ini dengan *method* "pd.series" untuk nilai atribut, dan *method* "pd.Categorical" untuk tipe data.

Nilai Atribut dan Tipe Data

```
[ ]  # Membuat data menjadi kategori
     df['experience_level'] = pd.Categorical(df.experience_level)
     el = pd.Series(df['experience_level'].cat.categories)

     # Output
     print("Nilai atribut 'experience_level': ")
     print(el.to_string())
     print(f"\n")

     print("Tipe data: ", df['experience_level'].dtypes)
```

```
Nilai atribut 'experience_level':
0    EN
1    EX
2    MI
3    SE


Tipe data:  category
```

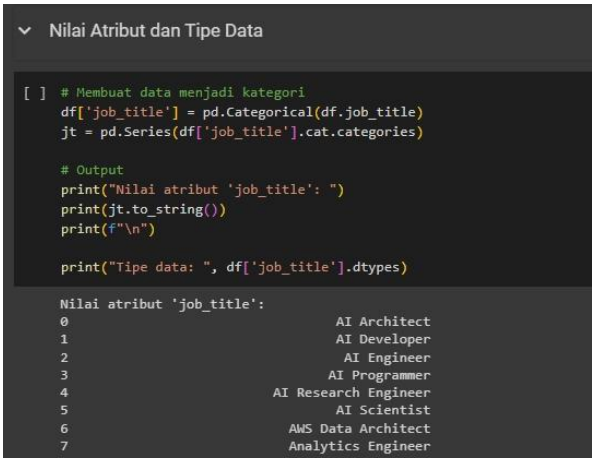
(a)

Nilai Atribut dan Tipe Data

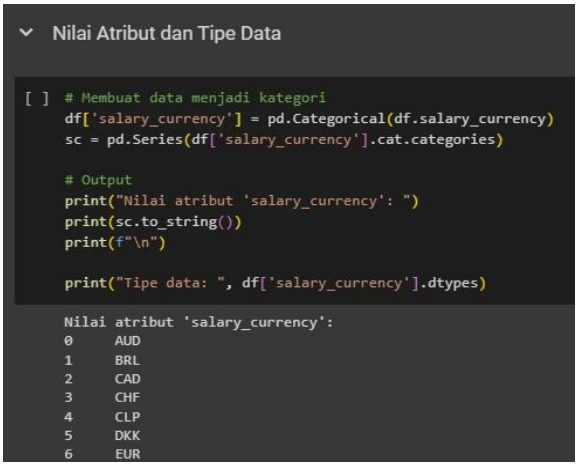
```
[ ]  # Membuat data menjadi kategori
     df['employment_type'] = pd.Categorical(df.employment_type)
     et = pd.Series(df['employment_type'].cat.categories)

     # Output
     print("Nilai atribut 'employment_type': ")
     print(et.to_string())
     print(f"\n")

     print("Tipe data: ", df['employment_type'].dtypes)
```

```
Nilai atribut 'employment_type':
0    CT
1    FL
2    FT
3    PT
```

(b)



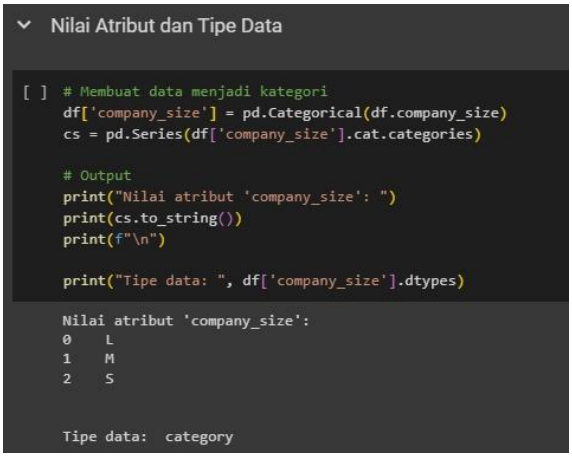
(c)



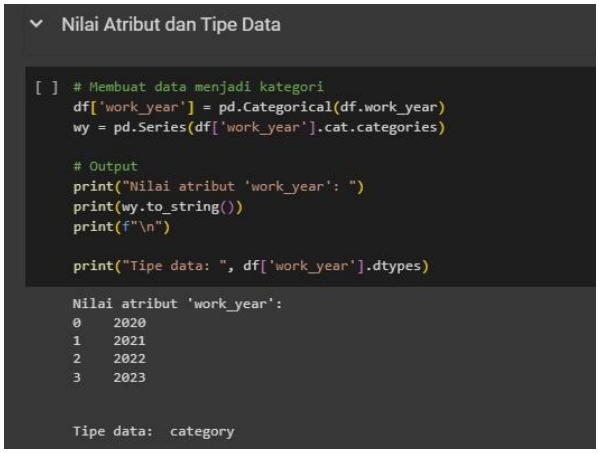
(d)

(e)

(f)



(g)



(h)

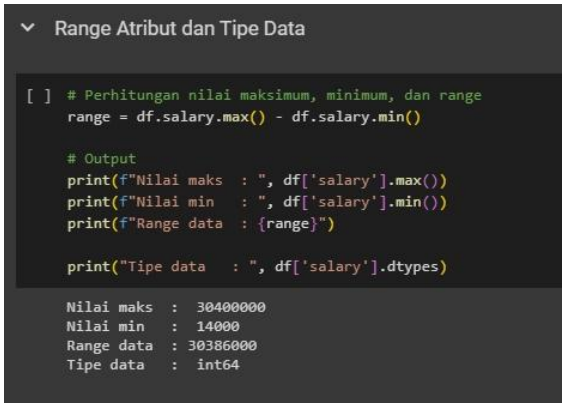
Gambar 2.1 Nilai atribut dan tipe data

Pada Gambar 2.1, terdapat macam-macam nilai atribut dan tipe data dari masing-masing atribut seperti pada gambar:

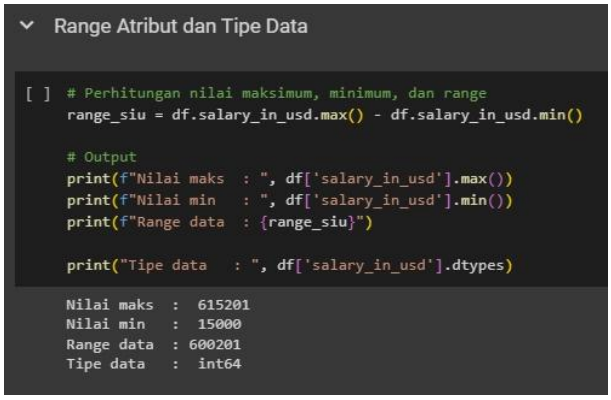
a. menunjukkan atribut experience level,
b. menunjukkan atribut employment type,
c. menunjukkan atribut job title,
d. menunjukkan atribut salary currency,
e. menunjukkan atribut employee residence,
f. menunjukkan atribut company location,
g. menunjukkan atribut company size, dan
h. menunjukkan atribut work year.

2.2. Range Atribut dan Tipe Data

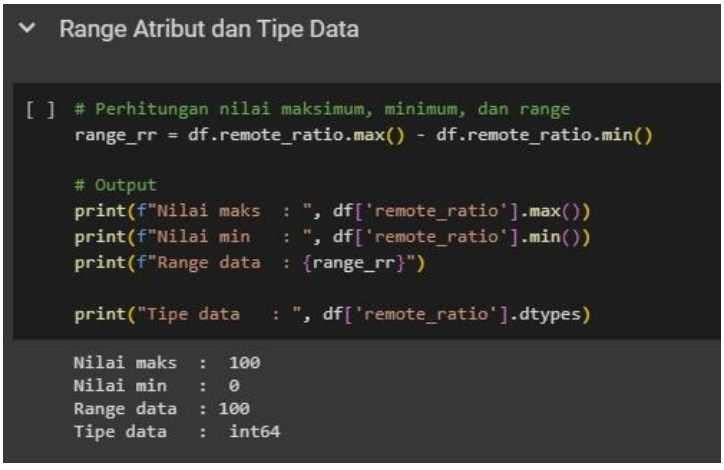
Range dan tipe data yang dimiliki setiap atribut dalam *dataset* ditampilkan dalam program seperti pada Gambar 2.2. Atribut yang dapat dicari *range*-nya hanyalah atribut dengan data kuantitatif yakni "salary", "salary_in_usd", dan "remote_ratio".



(a)



(b)



(c)

Gambar 2.2 Range atribut dan tipe data

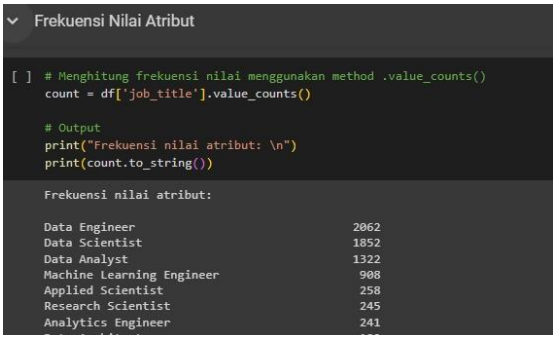
2.3. Frekuensi Data

Frekuensi data dihitung menggunakan *method* ".value_counts()" sehingga akan ditampilkan jumlah frekuensi data seluruh atribut tiap tahunnya, tiap kota atau sebagainya. Pada bagian ini, semua atribut memiliki frekuensi datanya masing-masing seperti pada Gambar 2.3.

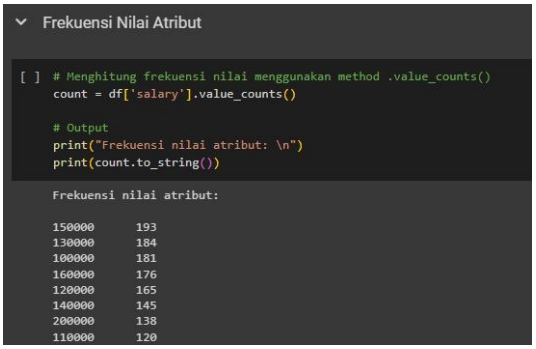
(a)

(b)

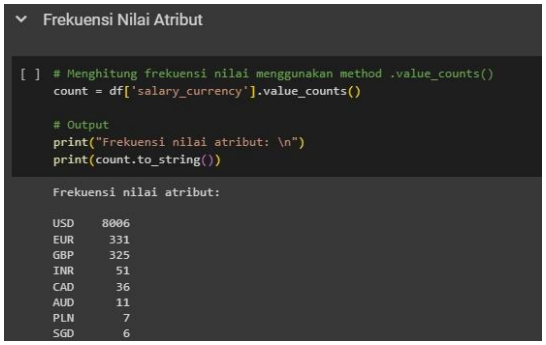
(c)



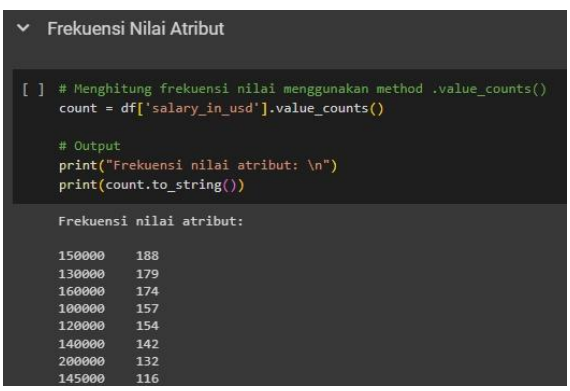
(d)



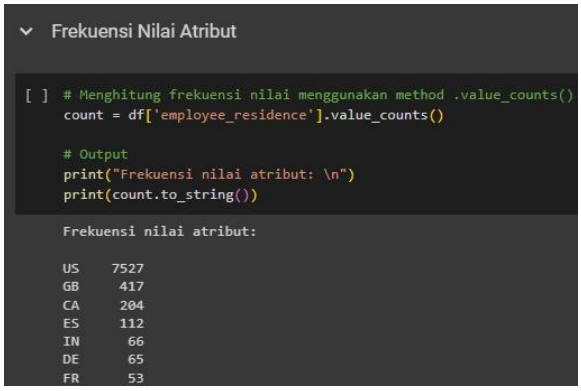
(e)



(f)

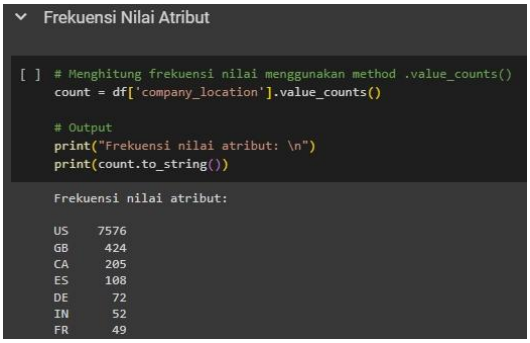


(g)

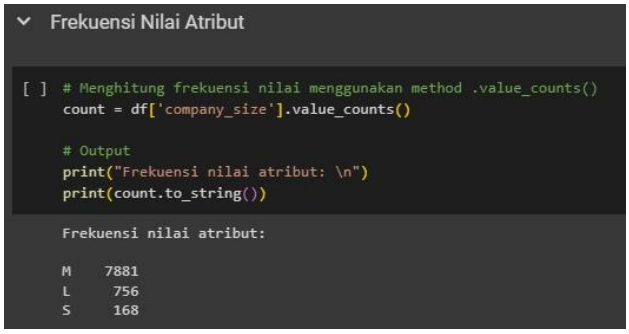


(h)

(i)



(j)



(k)

Gambar 2.3 Frekuensi nilai

Gambar 2.3 adalah macam-macam frekuensi nilai dari masing-masing atribut untuk gambar:

a. menunjukkan atribut work year,
b. menunjukkan atribut experience level,
c. menunjukkan atribut employment type,
d. menunjukkan atribut job title,
e. menunjukkan atribut salary,
f. menunjukkan atribut salary currency,
g. menunjukkan atribut salary in USD,
h. menunjukkan atribut employee residence,

i. menunjukkan atribut remote ratio,
j. menunjukkan atribut company location, dan
k. menunjukkan atribut company size.

2.4. Persentase Kekotoran

Persentase kekotoran digunakan untuk mengecek kekosongan data. Penjumlahan data kosong dilakukan dengan method ".isnull()" dan "sum()", kemudian akan dibagi dengan jumlah seluruh data lalu dikalikan dengan seratus untuk menampilkan jumlah data kosong dalam bentuk persentase.

Persentase Kekotoran

```
[ ]  # Perhitungan tingkat kekotoran data
     kotor = df['work_year'].isnull().sum()
     Data_sum = len(df)
     persenKotor = (kotor/Data_sum) * 100

     # Output
     print("Persentase data kosong: ", int(persenKotor), "%")
```

```
Persentase data kosong:  0 %
```

(a)

Persentase Kekotoran

```
[ ]  # Perhitungan tingkat kekotoran data
     kotor_el = df['experience_level'].isnull().sum()
     Data_sum_el = len(df)
     persenKotor_el = (kotor_el/Data_sum_el) * 100

     # Output
     print("Persentase data kosong: ", int(persenKotor_el), "%")
```

```
Persentase data kosong:  0 %
```

(b)

Persentase Kekotoran

```
[ ]  # Perhitungan tingkat kekotoran data
     kotor_et = df['employment_type'].isnull().sum()
     Data_sum_et = len(df)
     persenKotor_et = (kotor_et/Data_sum_et) * 100

     # Output
     print("Persentase data kosong: ", int(persenKotor_et), "%")
```

```
Persentase data kosong:  0 %
```

(c)

Persentase Kekotoran

```
[ ]  # Perhitungan tingkat kekotoran data
     kotor_jt = df['job_title'].isnull().sum()
     Data_sum_jt = len(df)
     persenKotor_jt = (kotor_jt/Data_sum_jt) * 100

     # Output
     print("Persentase data kosong: ", int(persenKotor_jt), "%")
```

```
Persentase data kosong:  0 %
```

(d)

Persentase Kekotoran

```
[ ]  # Perhitungan tingkat kekotoran data
     kotor_s = df['salary'].isnull().sum()
     Data_sum_s = len(df)
     persenKotor_s = (kotor_s/Data_sum_s) * 100

     # Output
     print("Persentase data kosong: ", int(persenKotor_s), "%")
```

```
Persentase data kosong:  0 %
```

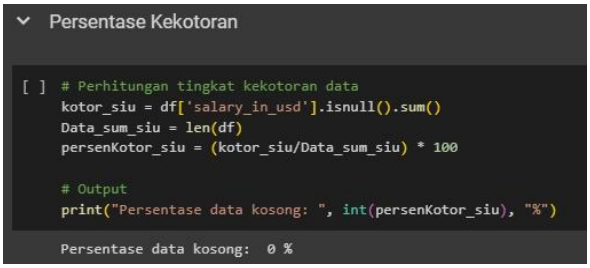
(e)

Persentase Kekotoran

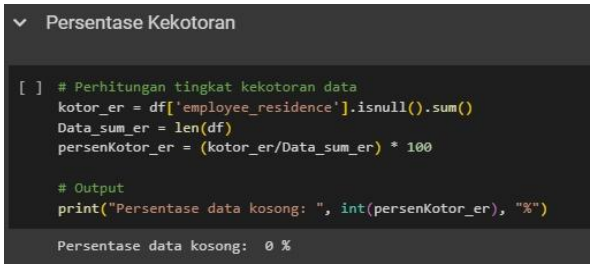
```
[ ]  # Perhitungan tingkat kekotoran data
     kotor_sc = df['salary_currency'].isnull().sum()
     Data_sum_sc = len(df)
     persenKotor_sc = (kotor_sc/Data_sum_sc) * 100

     # Output
     print("Persentase data kosong: ", int(persenKotor_sc), "%")
```

```
Persentase data kosong:  0 %
```

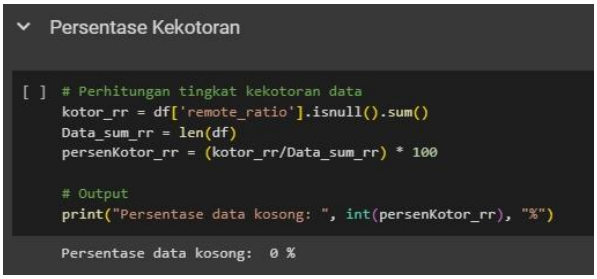
(f)



(g)

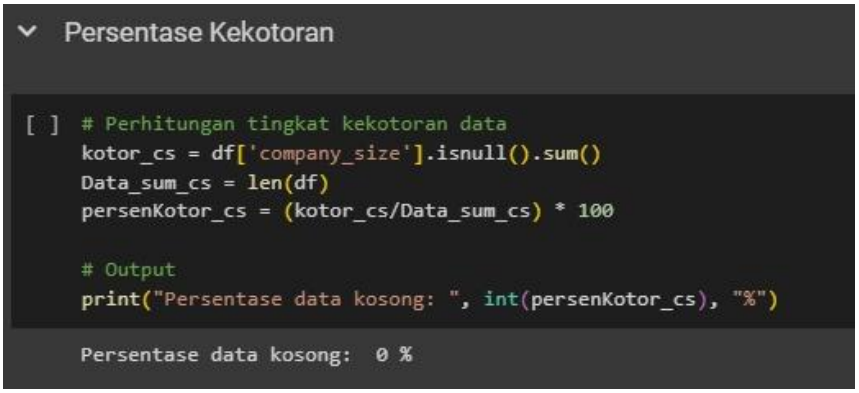


(h)



(i)

(j)



(k)

Gambar 2.4 Persentase kekotoran

Pada Gambar 2.4, ditunjukkan macam-macam persentase kekotoran dari masing-masing atribut untuk gambar:

a. Menunjukkan atribut work year
b. Menunjukkan atribut experience level
c. Menunjukkan atribut employment type
d. Menunjukkan atribut job title
e. Menunjukkan atribut salary
f. Menunjukkan atribut salary currency
g. Menunjukkan atribut salary in USD
h. Menunjukkan atribut employee residence
i. Menunjukkan atribut remote ratio
j. Menunjukkan atribut company location
k. Menunjukkan atribut company size

2.5. Keunikan Data

Untuk mengecek apakah data pada atribut unik digunakan *method* ".is_unique()" dan untuk menghitung jumlah data unik digunakan ".nunique()". Pada bagian ini juga ada beberapa atribut yang tidak perlu dicek karena data sudah berjenis kuantitatif seperti atribut "salary", "salary_in_usd", dan "remote_ratio".

Gambar 2.5 Keunikan data

2.6. Deskripsi Atribut

Pada bagian deskripsi atribut, ditampilkan beberapa data yang telah diproses sebelumnya menggunakan method ".describe()". Data yang dimunculkan sebagai meliputi jumlah data secara keseluruhan, jumlah data yang unik, dan jumlah data terbanyak serta tahunnya.

Deskripsi Atribut Secara Umum

```
[ ] # Membuat data menjadi kategori
    df['work_year'] = pd.Categorical(df.work_year)
    wy = pd.Series(df['work_year'].cat.categories)

    # Deskripsi menggunakan method .describe()
    wy2 = df['work_year'].describe()
    print(wy2.to_string())
```

```
count     8805
unique       4
top       2023
freq      6861
```

(a)

Deskripsi Atribut Secara Umum

```
[ ] # Membuat data menjadi kategori
    df['experience_level'] = pd.Categorical(df.experience_level)
    el = pd.Series(df['experience_level'].cat.categories)

    # Deskripsi menggunakan method .describe()
    el2 = df['experience_level'].describe()
    print(el2.to_string())
```

```
count     8805
unique       4
top         SE
freq      6336
```

(b)

Deskripsi Atribut Secara Umum

```
[ ] # Membuat data menjadi kategori
    df['employment_type'] = pd.Categorical(df.employment_type)
    et = pd.Series(df['employment_type'].cat.categories)

    # Deskripsi menggunakan method .describe()
    et2 = df['employment_type'].describe()
    print(et2.to_string())
```

```
count     8805
unique       4
top         FT
freq      8762
```

(c)

Deskripsi Atribut Secara Umum

```
[ ] # Membuat data menjadi kategori
    df['job_title'] = pd.Categorical(df.job_title)
    jt = pd.Series(df['job_title'].cat.categories)

    # Deskripsi menggunakan method .describe()
    jt2 = df['job_title'].describe()
    print(jt2.to_string())
```

```
count              8805
unique              124
top       Data Engineer
freq               2062
```

(d)

Deskripsi Atribut Secara Umum

```
# Deskripsi menggunakan method .describe()
s2 = df['salary'].describe()
print(s2.to_string())
```

```
count    8.805000e+03
mean     1.747287e+05
std      4.560690e+05
min      1.400000e+04
25%      1.055000e+05
50%      1.441000e+05
75%      1.900000e+05
max      3.040000e+07
```

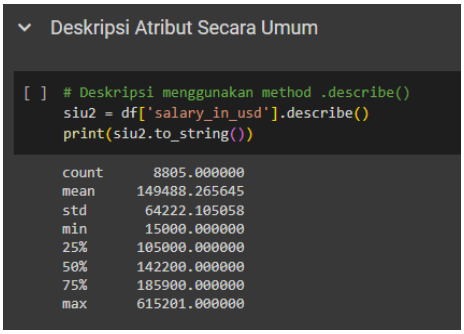
(e)

Deskripsi Atribut Secara Umum

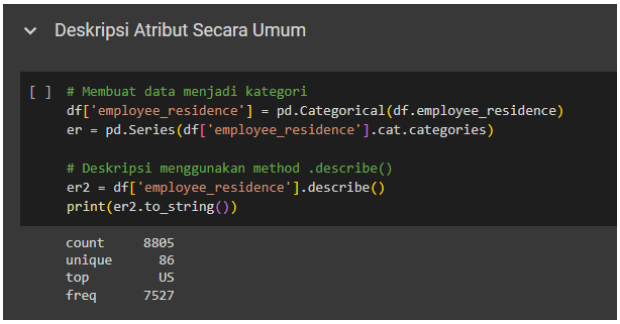
```
[ ] # Membuat data menjadi kategori
    df['salary_currency'] = pd.Categorical(df.salary_currency)
    sc = pd.Series(df['salary_currency'].cat.categories)

    # Deskripsi menggunakan method .describe()
    sc2 = df['salary_currency'].describe()
    print(sc2.to_string())
```

```
count     8805
unique      22
top        USD
freq      8006
```

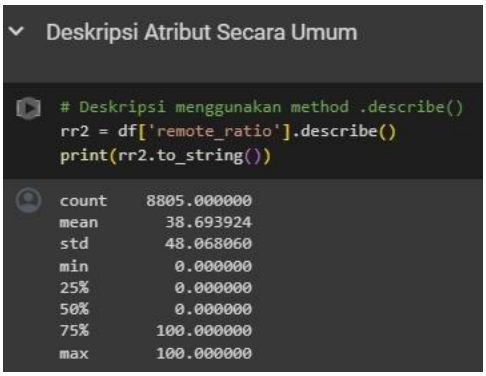
(f)



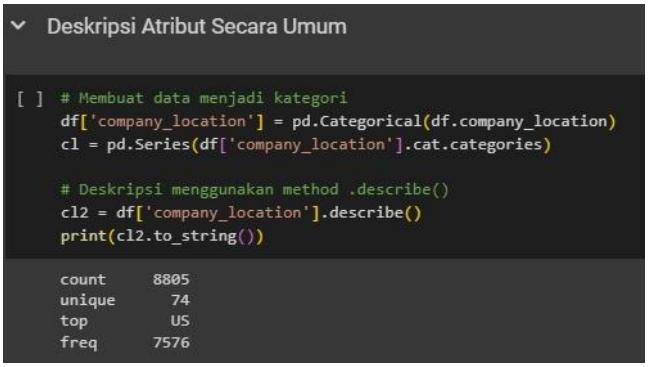
(g)



(h)



(i)



(j)

(k)

Gambar 2.6 Deskripsi atribut secara umum

BAB III
STATISTIK

3.1. Sampel Data

Sampel data yang dapat ditunjukkan dari *dataset* ini meliputi beberapa data baris awal dan akhir, data dalam *range* indeks tertentu, data maksimum dan minimum, serta data dengan suatu atribut. Dalam pengambilan sampel data, *library* yang digunakan adalah Pandas yang memiliki berbagai *methods* untuk mengakses berbagai jenis sampel data.

3.1.1. Memunculkan Beberapa Data Baris Awal

Data di beberapa baris awal dapat dimunculkan dengan dua metode, yakni menggunakan *method* ".head()" (Gambar 3.1) dan menggunakan *slicing* (Gambar 3.2). Pada Gambar 3.1 (a) ditampilkan 5 data pertama pada *dataset*. Ini dilakukan secara bawaan oleh *method* ".head()". *Method* ini dapat diberi argumen jumlah baris pertama yang ingin ditampilkan. Pada Gambar 3.1 (b), ditampilkan 10 data pertama pada *dataset*. Gambar 3.2 juga menampilkan 10 data pertama pada *dataset* tapi menggunakan metode *slicing*. Perlu diperhatikan bahwa indeks terakhir pada subset data ini merupakan kurang satu dari jumlah data pertama yang ditampilkan.

```
[ ] df.head()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	EX	FT	Data Science Director	212000	USD	212000	US	0	US	M
1	2023	EX	FT	Data Science Director	190000	USD	190000	US	0	US	M
2	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
3	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
4	2023	SE	FT	Machine Learning Engineer	245700	USD	245700	US	0	US	M

(a)

```
[ ] # 10 data awal
    df.head(10)
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	EX	FT	Data Science Director	212000	USD	212000	US	0	US	M
1	2023	EX	FT	Data Science Director	190000	USD	190000	US	0	US	M
2	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
3	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
4	2023	SE	FT	Machine Learning Engineer	245700	USD	245700	US	0	US	M
5	2023	SE	FT	Machine Learning Engineer	132300	USD	132300	US	0	US	M
6	2023	MI	FT	Data Specialist	90000	USD	90000	US	0	US	M
7	2023	MI	FT	Data Specialist	80000	USD	80000	US	0	US	M
8	2023	SE	FT	Machine Learning Engineer	212000	USD	212000	US	0	US	M
9	2023	SE	FT	Machine Learning Engineer	93300	USD	93300	US	0	US	M

(b)

Gambar 3.1 Memunculkan beberapa data baris awal dengan *method* ".head()" (a) tanpa argumen dan (b) dengan argumen jumlah baris pertama yang ingin ditampilkan

```
[ ] # 10 data pertama

    df[:10]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	EX	FT	Data Science Director	212000	USD	212000	US	0	US	M
1	2023	EX	FT	Data Science Director	190000	USD	190000	US	0	US	M
2	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
3	2023	MI	FT	Business Intelligence Engineer	35000	GBP	43064	GB	0	GB	M
4	2023	SE	FT	Machine Learning Engineer	245700	USD	245700	US	0	US	M
5	2023	SE	FT	Machine Learning Engineer	132300	USD	132300	US	0	US	M
6	2023	MI	FT	Data Specialist	90000	USD	90000	US	0	US	M
7	2023	MI	FT	Data Specialist	80000	USD	80000	US	0	US	M
8	2023	SE	FT	Machine Learning Engineer	212000	USD	212000	US	0	US	M
9	2023	SE	FT	Machine Learning Engineer	93300	USD	93300	US	0	US	M

Gambar 3.2 Memunculkan beberapa data baris awal dengan *slicing*

3.1.2. Memunculkan Beberapa Data Baris Akhir

Data di beberapa baris akhir dapat dimunculkan dengan dua metode, yakni menggunakan *method* ".tail()" (Gambar 3.3) dan menggunakan *slicing* (Gambar 3.4). Secara bawaan, ".tail()" menampilkan 5 data terakhir seperti pada Gambar 3.3 (a). *Method* ini juga bisa diberi argumen jumlah data terakhir yang ingin ditampilkan seperti Gambar 3.3 (b) yang menampilkan 10 data terakhir. Gambar 3.4 juga menampilkan 10 data terakhir tetapi menggunakan metode *slicing*.

```
[ ] df.tail()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L
8801	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	US	L
8802	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	US	S
8803	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	US	L
8804	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	L

(a)

```
[ ] # 10 data terakhir

    df.tail(10)
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8795	2021	SE	FT	Director of Data Science	168000	USD	168000	JP	0	JP	S
8796	2021	MI	FT	Data Scientist	160000	SGD	119059	SG	100	IL	M
8797	2021	MI	FT	Applied Machine Learning Scientist	423000	USD	423000	US	50	US	L
8798	2021	MI	FT	Data Engineer	24000	EUR	28369	MT	50	MT	L
8799	2021	SE	FT	Data Specialist	165000	USD	165000	US	100	US	L
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L
8801	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	US	L
8802	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	US	S
8803	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	US	L
8804	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	L

(b)

Gambar 3.3 Memunculkan beberapa data baris akhir dengan *method* ".tail()" (a) tanpa argumen dan (b) dengan argumen jumlah baris terakhir yang ingin ditampilkan

```
[ ]  # 10 data terakhir

     akhir = len(df)
     df[akhir - 10:akhir]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8795	2021	SE	FT	Director of Data Science	168000	USD	168000	JP	0	JP	S
8796	2021	MI	FT	Data Scientist	160000	SGD	119059	SG	100	IL	M
8797	2021	MI	FT	Applied Machine Learning Scientist	423000	USD	423000	US	50	US	L
8798	2021	MI	FT	Data Engineer	24000	EUR	28369	MT	50	MT	L
8799	2021	SE	FT	Data Specialist	165000	USD	165000	US	100	US	L
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L
8801	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	US	L
8802	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	US	S
8803	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	US	L
8804	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	L

Gambar 3.4 Memunculkan beberapa data baris akhir dengan *slicing*

3.1.3. Memunculkan Data dalam *Range* Indeks Tertentu (*Slicing Method*)

Slicing adalah sebuah metode untuk memunculkan data dalam *range* indeks tertentu. Argumen yang diberikan adalah indeks awal dan indeks akhir ditambah satu. Kedua argumen ini dipisah dengan titik dua (:). Apabila indeks awal *slicing* adalah indeks pertama dalam *dataset*, maka argumen indeks awal tidak perlu ditulis. Hal tersebut juga dapat diterapkan untuk indeks akhir *slicing* yang merupakan indeks terakhir dalam *dataset*. Gambar 3.5 (a) menampilkan data ke-5 hingga ke-16. Sedangkan, Gambar 3.5 (b) menampilkan data ke-8800 hingga terakhir.

```
[ ]  # Data ke-5 hingga ke-16

     df[4:16]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
4	2023	SE	FT	Machine Learning Engineer	245700	USD	245700	US	0	US	M
5	2023	SE	FT	Machine Learning Engineer	132300	USD	132300	US	0	US	M
6	2023	MI	FT	Data Specialist	90000	USD	90000	US	0	US	M
7	2023	MI	FT	Data Specialist	80000	USD	80000	US	0	US	M
8	2023	SE	FT	Machine Learning Engineer	212000	USD	212000	US	0	US	M
9	2023	SE	FT	Machine Learning Engineer	93300	USD	93300	US	0	US	M
10	2023	MI	FT	Data Scientist	212000	USD	212000	US	0	US	M
11	2023	MI	FT	Data Scientist	93300	USD	93300	US	0	US	M
12	2023	SE	FT	ML Engineer	276000	USD	276000	US	0	US	M
13	2023	SE	FT	ML Engineer	174000	USD	174000	US	0	US	M
14	2023	MI	FT	Data Engineer	133000	USD	133000	US	0	US	M
15	2023	MI	FT	Data Engineer	58300	USD	58300	US	0	US	M

(a)

```
[ ]  # Data ke-8800 hingga ke-8805

     df[8799:]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8799	2021	SE	FT	Data Specialist	165000	USD	165000	US	100	US	L
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L
8801	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	US	L
8802	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	US	S
8803	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	US	L
8804	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	L

(b)

Gambar 3.5 Memunculkan data dalam *range* indeks tertentu menggunakan *slicing*

3.1.4. Data Bernilai Maksimum

Beberapa data terbesar dapat dimunculkan dengan *method* ".sort_values()" dengan argumen pertama adalah nama atribut yang ingin diurutkan dalam bentuk List data String dan argumen kedua yakni "ascending = False" atau "ascending = 0". Nilai argumen "ascending =" dapat berupa sebuah List data boolean dengan urutan elemen sesuai dengan List nama atribut. Gambar 3.6 mengurutkan nilai pada atribut "salary" dari yang tertinggi lalu menampilkan 10 data dengan nilai "salary" terbesar. Atribut ini sebenarnya merupakan nominal gaji dalam mata uang negara asal sehingga memberikan informasi yang tidak terlalu berarti. Gambar 3.7 memaparkan 20 pendapatan dalam USD tertinggi. Dari data ini dapat diperoleh informasi perkiraan pekerjaan dari negara apa saja yang memiliki taraf hidup yang paling tinggi. Perlu diperhatikan bahwa data tahun masih belum terurut.

```
[ ] df.sort_values(['salary', 'job_title'], ascending = [0,1])[:10]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8726	2021	MI	FT	Data Scientist	30400000	CLP	40038	CL	100	CL	L
8635	2021	MI	FT	BI Data Analyst	11000000	HUF	36259	HU	50	US	L
8705	2020	MI	FT	Data Scientist	11000000	HUF	35735	HU	50	HU	L
8542	2021	MI	FT	ML Engineer	8500000	JPY	77364	JP	50	JP	S
8039	2022	SE	FT	Lead Machine Learning Engineer	7500000	INR	95386	IN	50	IN	L
8804	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	L
8543	2021	MI	FT	ML Engineer	7000000	JPY	63711	JP	50	JP	S
7441	2022	EN	FT	Data Scientist	6600000	HUF	17684	HU	100	HU	M
8260	2022	EX	FT	Head of Machine Learning	6000000	INR	76309	IN	50	IN	L
4873	2022	EN	FT	Research Engineer	5500000	JPY	41809	JP	50	JP	L

Gambar 3.6 Data 10 pendapatan dengan nominal mata uang negara asal tertinggi

```
[ ] df.sort_values(['salary_in_usd', 'job_title'], ascending = [0,1])[:20]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
44	2023	EN	FL	Business Intelligence Consultant	500000	GBP	615201	IN	100	IN	S
8587	2020	MI	FT	Research Scientist	450000	USD	450000	US	0	US	M
7099	2022	MI	FT	Data Analyst	350000	GBP	430967	GB	0	GB	M
5099	2023	MI	FT	Analytics Engineer	350000	GBP	430640	GB	0	GB	M
8797	2021	MI	FT	Applied Machine Learning Scientist	423000	USD	423000	US	50	US	L
5627	2023	SE	FT	AI Scientist	1500000	ILS	417937	IL	0	IL	L
8732	2021	EX	CT	Principal Data Scientist	416000	USD	416000	US	100	US	S
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L
8530	2022	SE	FT	Data Analytics Lead	405000	USD	405000	US	100	US	L
1079	2023	MI	FT	Research Scientist	405000	USD	405000	US	0	US	L
4329	2023	SE	FT	Analytics Engineering Manager	325000	GBP	399880	GB	50	GB	L
3657	2023	SE	FT	Machine Learning Engineer	392000	USD	392000	US	0	US	M
6383	2023	SE	FT	Data Analyst	385000	USD	385000	US	0	US	M
1245	2023	SE	FT	Data Engineer	385000	USD	385000	US	0	US	M
1295	2023	SE	FT	Data Engineer	385000	USD	385000	US	0	US	M
1876	2023	SE	FT	Data Infrastructure Engineer	385000	USD	385000	US	0	US	M
2094	2023	SE	FT	ML Engineer	385000	USD	385000	US	100	US	M
1289	2023	SE	FT	Research Engineer	385000	USD	385000	US	0	US	M
3490	2023	SE	FT	ML Engineer	383910	USD	383910	US	0	US	M
8535	2022	SE	FT	Applied Data Scientist	380000	USD	380000	US	100	US	L

Gambar 3.7 Data 20 besar pendapatan dalam USD tertinggi

Data dapat disaring terlebih dahulu berdasarkan syarat tertentu. Syarat dapat berupa nilai tertentu dari sebuah atribut kategorikal maupun sebuah

range dari sebuah atribut kuantitatif. Ini dilakukan dengan membuat sebuah DataFrame baru menggunakan *method* ".loc[]" dengan argumen yakni syarat yang ingin diper. Syarat lebih dari satu dapat diberikan dengan pemisah logika yakni "&" sebagai operator logika "dan" dan "|" sebagai operator logika "atau". Setiap syarat harus dibuka dan ditutup dengan tanda kurung seperti pada gambar 3.9 dan 3.10.

Gambar 3.8 mengurutkan nilai "salary" dari yang tertinggi lalu menampilkan 15 data teratas. Data ini juga memberikan informasi yang tidak begitu berarti karena nilai "salary" merupakan mata uang asal yang nilai konversinya berbeda-beda. Sedangkan, Gambar 3.9 memberikan informasi yang lebih spesifik yakni 12 pendapatan dalam USD tertinggi dengan pengalaman *executive_level* dan pekerjaan sepenuhnya jarak jauh. Perbandingan dalam data ini lebih relevan karena nilai pendapatan telah dikonversi menjadi USD. Gambar 3.10 juga lebih berarti karena memaparkan nilai pendapatan yang seluruhnya dalam EUR. Perlu diperhatikan bahwa data tahun belum dispesifikasikan sehingga data tertinggi merupakan yang tertinggi sepanjang masa yang terdata.

```
[ ] dfAP = df.loc[df['job_title'] == 'Applied Machine Learning Scientist']

    dfAP.sort_values(['salary', 'job_title'], ascending = [0,1])[:15]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
4720	2023	EN	FT	Applied Machine Learning Scientist	4000000	INR	48644	IN	100	DE	L
8145	2022	MI	FL	Applied Machine Learning Scientist	2400000	INR	30523	IN	100	IN	S
8797	2021	MI	FT	Applied Machine Learning Scientist	423000	USD	423000	US	50	US	L
8043	2022	MI	FT	Applied Machine Learning Scientist	173000	USD	173000	US	50	US	M
6929	2022	SE	FT	Applied Machine Learning Scientist	150000	USD	150000	US	100	US	M
7894	2022	SE	FT	Applied Machine Learning Scientist	108000	USD	108000	US	0	US	L
2495	2022	MI	CT	Applied Machine Learning Scientist	93000	EUR	97712	IT	100	NL	L
5758	2023	SE	FT	Applied Machine Learning Scientist	90000	USD	90000	US	100	US	L
8006	2022	MI	FT	Applied Machine Learning Scientist	75000	USD	75000	BO	100	US	M
8495	2022	MI	FT	Applied Machine Learning Scientist	75000	USD	75000	BO	100	US	L
8143	2022	SE	FT	Applied Machine Learning Scientist	73400	EUR	77119	FR	100	GB	L
5672	2023	EN	FT	Applied Machine Learning Scientist	40000	EUR	43187	DE	50	DE	M
8552	2021	MI	FT	Applied Machine Learning Scientist	38400	USD	38400	VN	100	US	M
8523	2022	EN	CT	Applied Machine Learning Scientist	29000	EUR	30469	TN	100	CZ	M

Gambar 3.8 Data 15 nominal pendapatan dalam mata uang negara asal tertinggi untuk pekerjaan Applied Machine Learning Scientist

```
[ ] dfEX = df.loc[(df['experience_level'] == 'EX') & (df['remote_ratio'] >= 50)]

    dfEX.sort_values(['salary_in_usd', 'job_title'], ascending = [0,1])[:12]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8732	2021	EX	CT	Principal Data Scientist	416000	USD	416000	US	100	US	S
2006	2023	EX	FT	Director of Data Science	375500	USD	375500	US	100	US	M
1804	2023	EX	FT	Head of Data	329500	USD	329500	US	100	US	M
2159	2023	EX	FT	Head of Data	329500	USD	329500	US	100	US	M
8751	2020	EX	FT	Director of Data Science	325000	USD	325000	US	100	US	L
8478	2022	EX	FT	Data Engineer	324000	USD	324000	US	100	US	M
5958	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
6104	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
6250	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
6521	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
7020	2022	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
7362	2022	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M

Gambar 3.9 Data 12 pendapatan dalam USD terbesar dari pengalaman EX dan remote ratio lebih dari sama dengan 50%

```
[ ] dfEUR = df.loc[((df['job_title'] == 'Data Engineer') | (df['job_title'] == 'Head of Data')) & (df['salary_currency'] == 'EUR')]

    dfEUR.sort_values(['salary', 'job_title'], ascending = [0,1])[:14]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
7198	2022	MI	FT	Data Engineer	105120	EUR	110446	LT	0	LT	M
2931	2023	SE	FT	Data Engineer	100000	EUR	107968	DE	100	DE	M
6354	2023	SE	FT	Data Engineer	95000	EUR	102569	IE	100	IE	M
8503	2021	SE	FT	Head of Data	87000	EUR	102839	SI	100	SI	L
2932	2023	SE	FT	Data Engineer	83913	EUR	90599	DE	100	DE	M
3127	2023	SE	FT	Data Engineer	80000	EUR	86374	DE	100	SE	L
7962	2022	MI	FT	Data Engineer	80000	EUR	84053	GR	100	GR	M
7966	2022	MI	FT	Data Engineer	80000	EUR	84053	ES	100	ES	M
8268	2022	MI	FT	Data Engineer	80000	EUR	84053	ES	100	ES	M
8272	2022	MI	FT	Data Engineer	80000	EUR	84053	GR	100	GR	M
7199	2022	MI	FT	Data Engineer	75360	EUR	79178	LT	0	LT	M
1119	2023	MI	FT	Data Engineer	75000	EUR	80976	FR	50	FR	M
4484	2023	MI	FT	Data Engineer	75000	EUR	80976	ES	100	ES	M
2984	2023	SE	FT	Data Engineer	72000	EUR	77737	ES	100	ES	M

Gambar 3.10 Data 14 pendapatan terbesar dari mata uang EUR dan pekerjaan Data Engineer atau Head of Data

3.1.5. Data Bernilai Minimum

Sama seperti menampilkan beberapa data terbesar, mencari beberapa data terkecil juga dapat ditampilkan menggunakan *method* ".sort_values()". Hal yang membedakan adalah pada argumen "ascending =" yang untuk kasus ini diisi boolean "True" atau "1". Syarat terhadap kelompok data yang ingin ditampilkan juga dapat diberikan dengan cara membuat DataFrame baru yang menggunakan *method* ".loc[]".

Gambar 3.12 dan 3.13 juga merupakan contoh sampel data yang lebih berarti karena tidak seperti Gambar 3.11, kedua sampel data tersebut sudah dikonversi ke mata uang tertentu sehingga terdapat perbandingan yang jelas. Gambar 3.12 menunjukkan gaji terendah untuk pengalaman menengah sepanjang masa yang terdata adalah pekerjaan *business intelligence developer* dengan indeks 3071. Pada Gambar 3.13, gaji *data scientist* dalam INR di India terendah adalah pada indeks 8699.

```
[ ] df.sort_values(['salary', 'job_title'], ascending = [1,0])[:5]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8637	2020	EN	PT	ML Engineer	14000	EUR	15966	DE	100	DE	S
5282	2020	EX	FT	Staff Data Analyst	15000	USD	15000	NG	0	CA	M
7910	2021	EN	FT	Machine Learning Developer	15000	USD	15000	TH	100	TH	L
8207	2022	EN	FT	Data Analyst	15000	USD	15000	ID	0	ID	L
3071	2022	MI	FT	Business Intelligence Developer	15000	USD	15000	GH	100	GH	M

Gambar 3.11 Data 5 pendapatan dalam mata uang negara asal terendah

```
[ ] dfMI = df.loc[df['experience_level'] == 'MI']

    dfMI.sort_values(['salary_in_usd', 'job_title'], ascending = [1,0])[:12]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
3071	2022	MI	FT	Business Intelligence Developer	15000	USD	15000	GH	100	GH	M
3332	2023	MI	FT	Data Analyst	866000	PHP	15680	PH	50	PH	L
6687	2022	MI	FT	Computer Vision Engineer	1250000	INR	15897	IN	100	IN	M
6355	2023	MI	FT	Product Data Analyst	1350000	INR	16417	IN	100	IN	L
8699	2021	MI	FT	Data Scientist	1250000	INR	16904	IN	100	IN	S
7119	2021	MI	FT	Data Analyst	1250000	INR	16904	IN	50	IN	L
1655	2023	MI	FT	Business Data Analyst	17000	USD	17000	AM	100	RU	L
5833	2023	MI	FT	Data Scientist	1400000	INR	17025	IN	100	IN	L
6640	2023	MI	FT	Data Analytics Lead	1440000	INR	17511	IN	50	SG	M
8490	2022	MI	FT	Business Data Analyst	1400000	INR	17805	IN	100	IN	M
8771	2021	MI	FT	Big Data Engineer	18000	USD	18000	MD	0	MD	S
5836	2023	MI	FT	Lead Data Analyst	1500000	INR	18241	IN	50	IN	L

Gambar 3.12 Data 12 pendapatan dalam USD terendah untuk pengalaman kerja MI

```
[ ] dfINR = df.loc[(df['salary_currency'] == 'INR') & (df['job_title'] == 'Data Scientist') & (df['company_location'] == 'IN')]

    dfINR.sort_values(['salary', 'job_title'], ascending = [1,0])[:14]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
8699	2021	MI	FT	Data Scientist	1250000	INR	16904	IN	100	IN	S
5833	2023	MI	FT	Data Scientist	1400000	INR	17025	IN	100	IN	L
8494	2022	EN	FT	Data Scientist	1400000	INR	17805	IN	100	IN	M
7863	2022	EN	FT	Data Scientist	1800000	INR	22892	IN	50	IN	M
8650	2021	EN	FT	Data Scientist	2100000	INR	28399	IN	100	IN	M
8642	2021	EN	FT	Data Scientist	2200000	INR	29751	IN	50	IN	L
8491	2022	MI	FT	Data Scientist	2400000	INR	30523	IN	100	IN	L
8735	2021	MI	FT	Data Scientist	2500000	INR	33808	IN	0	IN	M
8717	2020	MI	FT	Data Scientist	3000000	INR	40481	IN	0	IN	L
6829	2021	SE	FT	Data Scientist	4000000	INR	54094	IN	100	IN	L

Gambar 3.13 Data 14 pendapatan dalam mata uang asal terendah dengan nilai mata uang INR, pekerjaan Data Scientist, dan lokasi perusahaan di IN

3.1.6. Penyaringan Data berdasarkan Atribut

Sebuah DataFrame dapat disaring dengan syarat tertentu menggunakan *method* ".loc[]" yang diisi argumen berupa kondisi yang diperlakukan terhadap atribut DataFrame yang disaring. Argumen dapat berupa kombinasi dari ekspresi yang dinyatakan dengan operator komparasi maupun logika. Perlu diperhatikan bahwa setiap ekspresi yang dipisah dengan operator logika harus dibuka dan ditutup tanda kurung.

Kasus 1: Data pendapatan kotor pada jenis pekerjaan PT

```
[ ] df.loc[df['employment_type'] == 'PT']
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
4161	2021	MI	PT	Business Data Analyst	56000	USD	56000	GH	100	US	M
6069	2022	EN	PT	Data Analyst	34320	USD	34320	US	100	US	S
6262	2023	EN	PT	Data Analyst	78000	PLN	18160	PL	100	IN	L
6796	2022	EN	PT	Data Analyst	24000	EUR	25216	ES	100	US	L
7582	2022	EN	PT	Data Analyst	125404	USD	125404	CN	50	US	S
7997	2021	EN	PT	Computer Vision Software Engineer	120000	DKK	19073	DK	50	DK	L
8042	2022	EN	PT	Data Scientist	110000	USD	110000	DO	100	FR	M
8493	2022	MI	PT	Data Engineer	50000	EUR	52533	DE	50	DE	L
8548	2022	EN	PT	Data Scientist	100000	USD	100000	DZ	50	DZ	M
8637	2020	EN	PT	ML Engineer	14000	EUR	15966	DE	100	DE	S
8679	2021	MI	PT	Data Engineer	59000	EUR	69741	NL	100	NL	L
8742	2021	EN	PT	Computer Vision Engineer	180000	DKK	28609	DK	50	DK	S
8762	2020	EN	PT	Data Scientist	19000	EUR	21669	IT	50	IT	S

Gambar 3.14 Data pendapatan kotor pada jenis pekerjaan PT

```
[ ] df.loc[(df['work_year'] == 2022) & (df['job_title'] == 'Data Analyst')]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
6069	2022	EN	PT	Data Analyst	34320	USD	34320	US	100	US	S
6796	2022	EN	PT	Data Analyst	24000	EUR	25216	ES	100	US	L
6930	2022	MI	FT	Data Analyst	150000	USD	150000	US	0	US	M
6931	2022	MI	FT	Data Analyst	100000	USD	100000	US	0	US	M
6950	2022	MI	FT	Data Analyst	150000	USD	150000	US	0	US	M
...	...	...	...	...	...	...	...	...	...	...	...
8462	2022	SE	FT	Data Analyst	170000	USD	170000	US	100	US	M
8463	2022	SE	FT	Data Analyst	135000	USD	135000	US	100	US	M
8466	2022	MI	FT	Data Analyst	135000	USD	135000	US	100	US	M
8467	2022	MI	FT	Data Analyst	50000	USD	50000	US	100	US	M
8502	2022	MI	FT	Data Analyst	20000	USD	20000	GR	100	GR	S

272 rows × 11 columns

Gambar 3.15 Data pendapatan pada tahun 2022 untuk pekerjaan Data Analyst

```
[ ] df.loc[(df['company_location'] == 'US') & (df['company_size'] == 'M') & ((df['experience_level'] == 'EN') | (df['experience_level'] == 'EX'))]
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	EX	FT	Data Science Director	212000	USD	212000	US	0	US	M
1	2023	EX	FT	Data Science Director	190000	USD	190000	US	0	US	M
92	2023	EX	FT	Data Analyst	125000	USD	125000	US	0	US	M
93	2023	EX	FT	Data Analyst	87500	USD	87500	US	0	US	M
120	2023	EN	FT	Data Analyst	109900	USD	109900	US	0	US	M
...	...	...	...	...	...	...	...	...	...	...	...
8560	2022	EN	FT	Computer Vision Engineer	125000	USD	125000	US	0	US	M
8564	2021	EN	FT	Data Analyst	50000	USD	50000	US	100	US	M
8614	2021	EN	FT	Data Analyst	80000	USD	80000	US	100	US	M
8639	2021	EN	FT	Computer Vision Software Engineer	70000	USD	70000	US	100	US	M
8686	2021	EN	FT	Data Scientist	100000	USD	100000	US	100	US	M

460 rows × 11 columns

Gambar 3.16 Data dengan lokasi perusahaan di US, ukuran perusahaan M, dan pengalaman kerja EN atau EX

Data pada gambar 3.14 hanya disaring berdasarkan atribut "employment_type" yang bernilai "PT" yang berarti *part time*. Terlihat bahwa hanya terdapat 13 data dengan tipe pekerjaan *part time*. Pada gambar 3.15, terdapat 272 data pendapatan pada tahun 2022 dengan jabatan pekerjaan Data Analyst. Pada gambar

3.16, diperlakukan penyaringan dengan 3 kriteria utama, yakni lokasi perusahaan di US, ukuran perusahaan sedang (M), dan kriteria terakhir yang merupakan gabungan 2 kriteria, yakni pengalaman kerja karyawan *entry-level* (EN) atau *executive-level* (EX). Kriteria terakhir berarti DataFrame baru menampilkan semua data dengan atribut *experience_level* EN maupun EX. Jumlah data pada penyaringan tersebut adalah 460 data.

3.2. Informasi Statistik

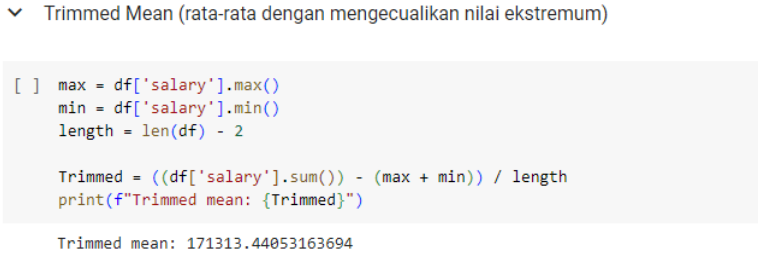
Beberapa informasi statistik yang dapat diperoleh adalah sebagai berikut.

1. Mean (rata-rata): Ukuran pemusatan dari suatu sebaran probabilitas
2. *Trimmed Mean*: mereduksi nilai rata-rata yang terlalu menyimpang karena nilai ekstremum
3. Median (nilai tengah): nilai yang berada di tengah data
4. Modus (nilai yang sering muncul): nilai yang sering muncul
5. *Variance* (variansi): menggambarkan varians kuantitas acak sebagai fungsi meannya
6. Standar Deviasi: menentukan seberapa dekat data dari sampel statistik dengan data rata-rata data tersebut
7. *Mean Absolute Deviation*: mengukur kesalahan perkiraan dalam unit ukuran yang sama seperti data aslinya
8. Persentil: nilai yang berada pada suatu bagian yang terbagi menjadi beberapa partisi persentil
9. Ekstremum: nilai terbesar dan nilai terkecil
10. *Range* (jangkauan): selisih nilai terbesar dan nilai terkecil
11. *Interquartile Range*: Selisih persentil ke-75 dengan persentil ke-25
12. *Outlier*: penyimpangan jauh dari beberapa data

Rincian informasi di atas dicari untuk atribut “salary”, “salary_in_usd”, dan “remote_ratio” karena ketiga atribut tersebut memiliki data kuantitatif yang dapat diproses secara statistika.

3.2.1. Atribut “salary”

(a)



(b)

Median (nilai tengah)

```
[ ]  print(f"Nilai tengah: {int(df['salary'].median())}")

     Nilai tengah: 144100
```

(c)

Modus (nilai yang sering muncul)

```
[ ]  Modus = df['salary'].mode().values[0]

     print(f"Nilai yang sering muncul: {Modus}")

     Nilai yang sering muncul: 150000
```

(d)

Variance (varians)

```
[ ]  print(f"Varians: {df['salary'].var()}")

     Varians: 207998959253.87048
```

(e)

Standar Deviasi

```
[ ]  print(f"Standar deviasi: {df['salary'].std()}")

     Standar deviasi: 456069.0290448042
```

(f)

Mean Absolute Deviation (deviasi rata-rata absolut)

```
[ ]  print("Nilai deviasi rata-rata absolut: ", (df['salary']-df['salary'].mean()).abs().mean())

     Nilai deviasi rata-rata absolut:  80079.9422655743
```

(g)

Percentile

```
[ ]  print(f"Persentil 10 %: {int(df['salary'].quantile(0.1))}")
     print(f"Persentil 25 %: {int(df['salary'].quantile(0.25))}")
     print(f"Persentil 50 %: {int(df['salary'].quantile(0.5))}")
     print(f"Persentil 61 %: {int(df['salary'].quantile(0.61))}")
     print(f"Persentil 75 %: {int(df['salary'].quantile(0.75))}")
     print(f"Persentil 90 %: {int(df['salary'].quantile(0.9))}")

     Persentil 10 %: 70000
     Persentil 25 %: 105500
     Persentil 50 %: 144100
     Persentil 61 %: 160000
     Persentil 75 %: 190000
     Persentil 90 %: 239748
```

(h)

Ekstremum

```
[ ]  print(f"Nilai maksimum: {df['salary'].max()}")
     print(f"Nilai minimum : {df['salary'].min()}")
```

```
Nilai maksimum: 30400000
Nilai minimum : 14000
```

(i)

Range (jangkauan)

```
[ ]  # Perhitungan range
     range = df.salary.max() - df.salary.min()

     # Output
     print(f"Jangkauan data: {range}")
```

```
Jangkauan data: 30386000
```

(j)

Interquartile Range (jangkauan interkuartil)

```
[ ]  dfn = df.sort_values(['salary', 'job_title'], ascending = [0,1])

     Q1 = df['salary'].quantile(0.25)
     Q3 = df['salary'].quantile(0.75)

     IQR = Q3 - Q1
     print(f"Jangkauan interkuartil: {int(IQR)}")
```

```
Jangkauan interkuartil: 84500
```

(k)

Outlier (pencilan)

```
[ ]  # outlier < Q1 - 1.5(Q3-Q1)
     # outlier > Q3 + 1.5(Q3-Q1)

     Q1 = df['salary'].quantile(0.25)
     Q3 = df['salary'].quantile(0.75)

     batasBawah = Q1 - 1.5*(Q3-Q1)
     batasAtas = Q3 + 1.5*(Q3-Q1)

     # Outlier
     print("Outlier data: \n")
     df.loc[(df['salary'] < batasBawah) | (df['salary'] > batasAtas)]
```

Outlier data:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
44	2023	EN	FL	Business Intelligence Consultant	500000	GBP	615201	IN	100	IN	S
192	2023	SE	FT	Research Scientist	370000	USD	370000	US	0	US	M
266	2023	SE	FT	Machine Learning Engineer	333500	USD	333500	US	0	US	M
270	2023	SE	FT	Data Architect	354200	USD	354200	US	100	US	M
349	2023	SE	FT	Machine Learning Engineer	328400	USD	328400	US	100	US	M
...	...	...	...	...	...	...	...	...	...	...	...
8780	2021	EN	FT	AI Scientist	1335000	INR	18053	IN	100	AS	S
8785	2021	MI	FT	Lead Data Analyst	1450000	INR	19609	IN	100	IN	L
8797	2021	MI	FT	Applied Machine Learning Scientist	423000	USD	423000	US	50	US	L
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L
8804	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	IN	L

183 rows × 11 columns

(l)

Distribusi Frekuensi Nilai

```
[ ]  freq_table = pd.crosstab(df['salary'], 'frequency')

     freq_table
```

col_0	frequency
salary	
14000	1
15000	4
15662	1
16000	1
17000	1
...	...
7000000	2
7500000	1
8500000	1
11000000	2
30400000	1

1485 rows × 1 columns

(m)

Gambar 3.17 *Syntax* program untuk menampilkan (a) mean, (b) *trimmed mean*, (c) median, (d) modus, (e) variansi, (f) standar deviasi, (g) *mean absolute deviation*, (h) persentil, (i) ekstremum, (j) jangkauan, (k) *interquartile range*, (l) *outlier*, dan (m) distribusi frekuensi nilai atribut "salary"

3.2.2. Atribut "salary_in_usd"

Mean (rata-rata)

```
[ ]  print(f"Nilai rata-rata: {df['salary_in_usd'].mean()}")

     Nilai rata-rata: 149488.26564452017
```

(a)

Trimmed Mean (rata-rata dengan mengecualikan nilai ekstremum)

```
[ ]  max = df['salary_in_usd'].max()
     min = df['salary_in_usd'].min()
     length = len(df) - 2

     Trimmed = ((df['salary_in_usd'].sum()) - (max + min)) / length
     print(f"Trimmed mean: {Trimmed}")

     Trimmed mean: 149450.639327502
```

(b)

Median (nilai tengah)

```
[ ]  print(f"Nilai tengah: {int(df['salary_in_usd'].median())}")

     Nilai tengah: 142200
```

(c)

Modus (nilai yang sering muncul)

```
[ ]  Modus = df['salary_in_usd'].mode().values[0]

     print(f"Nilai yang sering muncul: {Modus}")

     Nilai yang sering muncul: 150000
```

(d)

Variance (varians)

```
[ ]  print(f"Varians: {df['salary_in_usd'].var()}")

     Varians: 4124478778.1310377
```

(e)

Standar deviasi

```
[ ]  print(f"Standar deviasi: {df['salary_in_usd'].std()}")

     Standar deviasi: 64222.10505839121
```

(f)

Mean Absolute Deviation (deviasi rata-rata absolut)

```
[ ]  print("Nilai deviasi rata-rata absolut: ", (df['salary_in_usd']-df['salary_in_usd'].mean()).abs().mean())

     Nilai deviasi rata-rata absolut:  50200.51534964293
```

(g)

Percentile

```
[ ]  print(f"Persentil 10 %: {int(df['salary_in_usd'].quantile(0.1))}")
     print(f"Persentil 25 %: {int(df['salary_in_usd'].quantile(0.25))}")
     print(f"Persentil 50 %: {int(df['salary_in_usd'].quantile(0.5))}")
     print(f"Persentil 61 %: {int(df['salary_in_usd'].quantile(0.61))}")
     print(f"Persentil 75 %: {int(df['salary_in_usd'].quantile(0.75))}")
     print(f"Persentil 90 %: {int(df['salary_in_usd'].quantile(0.9))}")

     Persentil 10 %: 70659
     Persentil 25 %: 105000
     Persentil 50 %: 142200
     Persentil 61 %: 160000
     Persentil 75 %: 185900
     Persentil 90 %: 234000
```

(h)

Ekstremum

```
[ ]  print(f"Nilai maksimum: {df['salary_in_usd'].max()}")
     print(f"Nilai minimum : {df['salary_in_usd'].min()}")

     Nilai maksimum: 615201
     Nilai minimum : 15000
```

(i)

Range (jangkauan)

```
[ ]  # Perhitungan range
     range = df.salary_in_usd.max() - df.salary_in_usd.min()

     # Output
     print(f"Jangkauan data: {range}")
```

```
Jangkauan data: 600201
```

(j)

Interquartile Range (jangkauan interkuartil)

```
[ ]  Q1 = df['salary_in_usd'].quantile(0.25)
     Q3 = df['salary_in_usd'].quantile(0.75)

     IQR = Q3 - Q1
     print(f"Jangkauan interkuartil: {int(IQR)}")
```

```
Jangkauan interkuartil: 80900
```

(k)

Outlier (pencilan)

```
[ ]  # outlier < Q1 - 1.5(Q3-Q1)
     # outlier > Q3 + 1.5(Q3-Q1)

     Q1 = df['salary_in_usd'].quantile(0.25)
     Q3 = df['salary_in_usd'].quantile(0.75)

     batasBawah = Q1 - 1.5*(Q3-Q1)
     batasAtas = Q3 + 1.5*(Q3-Q1)

     # Outlier
     print("Outlier data: \n")
     df.loc[(df['salary_in_usd'] < batasBawah) | (df['salary_in_usd'] > batasAtas)]
```

```
Outlier data:
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
44	2023	EN	FL	Business Intelligence Consultant	500000	GBP	615201	IN	100	IN	S
192	2023	SE	FT	Research Scientist	370000	USD	370000	US	0	US	M
232	2023	SE	FT	Research Engineer	308000	USD	308000	US	0	US	M
266	2023	SE	FT	Machine Learning Engineer	333500	USD	333500	US	0	US	M
270	2023	SE	FT	Data Architect	354200	USD	354200	US	100	US	M
...	...	...	...	...	...	...	...	...	...	...	...
8587	2020	MI	FT	Research Scientist	450000	USD	450000	US	0	US	M
8732	2021	EX	CT	Principal Data Scientist	416000	USD	416000	US	100	US	S
8751	2020	EX	FT	Director of Data Science	325000	USD	325000	US	100	US	L
8797	2021	MI	FT	Applied Machine Learning Scientist	423000	USD	423000	US	50	US	L
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L

159 rows × 11 columns

(l)

Distribusi Frekuensi Nilai

```
[ ]  freq_table = pd.crosstab(df['salary_in_usd'], 'frequency')

     freq_table
```

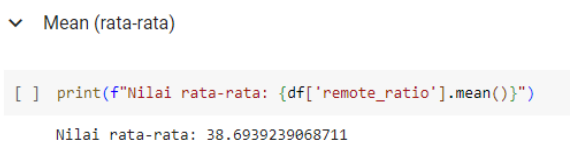
salary_in_usd	col_0 frequency
15000	4
15680	1
15809	1
15897	1
15966	1
...	...
423000	1
430640	1
430967	1
450000	1
615201	1

1768 rows × 1 columns

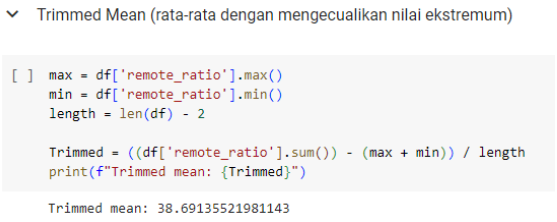
(m)

Gambar 3.18 *Syntax* program untuk menampilkan (a) mean, (b) *trimmed mean*, (c) median, (d) modus, (e) variansi, (f) standar deviasi, (g) *mean absolute deviation*, (h) persentil, (i) ekstremum, (j) jangkauan, (k) *interquartile range*, (l) *outlier*, dan (m) distribusi frekuensi nilai atribut "salary_in_usd"

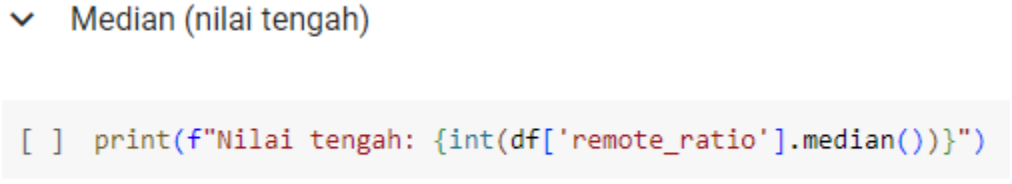
3.2.3. Atribut "remote_ratio"



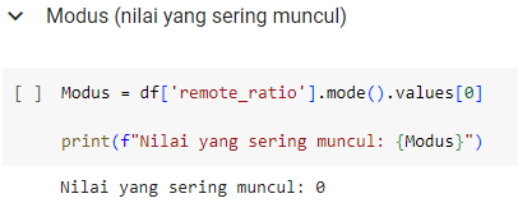
(a)



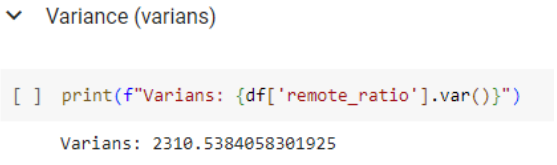
(b)



(c)



(d)



(e)

(f)

Mean Absolute Deviation (deviasi rata-rata absolut)

```
[ ] print("Nilai deviasi rata-rata absolut: ", (df['remote_ratio']-df['remote_ratio'].mean()).abs().mean())
```

```
Nilai deviasi rata-rata absolut:  46.485443167164384
```

(g)

Percentile

```
[ ] print(f"Persentil 10 %: {int(df['remote_ratio'].quantile(0.1))}")
    print(f"Persentil 25 %: {int(df['remote_ratio'].quantile(0.25))}")
    print(f"Persentil 50 %: {int(df['remote_ratio'].quantile(0.5))}")
    print(f"Persentil 61 %: {int(df['remote_ratio'].quantile(0.61))}")
    print(f"Persentil 75 %: {int(df['remote_ratio'].quantile(0.75))}")
    print(f"Persentil 90 %: {int(df['remote_ratio'].quantile(0.9))}")
```

```
Persentil 10 %: 0
Persentil 25 %: 0
Persentil 50 %: 0
Persentil 61 %: 50
Persentil 75 %: 100
Persentil 90 %: 100
```

(h)

Ekstremum

```
[ ] print(f"Nilai maksimum: {df['remote_ratio'].max()}")
    print(f"Nilai minimum : {df['remote_ratio'].min()}")
```

```
Nilai maksimum: 100
Nilai minimum : 0
```

(i)

Range (jangkauan)

```
[ ] # Perhitungan range
    range = df.remote_ratio.max() - df.remote_ratio.min()

    # Output
    print(f"Jangkauan data: {range}")
```

```
Jangkauan data: 100
```

(j)

Interquartile Range (jangkauan interkuartil)

```
[ ] Q1 = df['remote_ratio'].quantile(0.25)
    Q3 = df['remote_ratio'].quantile(0.75)

    IQR = Q3 - Q1
    print(f"Jangkauan interkuartil: {int(IQR)}")
```

```
Jangkauan interkuartil: 100
```

(k)

Outlier (pencilan)

```
[ ] # outlier < Q1 - 1.5(Q3-Q1)
    # outlier > Q3 + 1.5(Q3-Q1)

    Q1 = df['remote_ratio'].quantile(0.25)
    Q3 = df['remote_ratio'].quantile(0.75)

    batasBawah = Q1 - 1.5*(Q3-Q1)
    batasAtas = Q3 + 1.5*(Q3-Q1)

    # Outlier
    print("Outlier data: \n")
    df.loc[(df['remote_ratio'] < batasBawah) | (df['remote_ratio'] > batasAtas)]
```

```
Outlier data:
```

work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size

```
Tidak terdapat outlier data
```

(l)

Distribusi Frekuensi Nilai

```
[ ]  freq_table = pd.crosstab(df['remote_ratio'], 'frequency')

     freq_table
```

remote_ratio	col_0 frequency
0	5289
50	218
100	3298

(m)

Gambar 3.19 *Syntax* program untuk menampilkan (a) mean, (b) *trimmed mean*, (c) median, (d) modus, (e) variansi, (f) standar deviasi, (g) *mean absolute deviation*, (h) persentil, (i) ekstremum, (j) jangkauan, (k) *interquartile range*, (l) *outlier*, dan (m) distribusi frekuensi nilai atribut “remote_ratio”

3.2.4. Rekapitulasi Statistik

Informasi Statistik		Atribut		
		“salary”	**“salary_in_usd”**	**“remote_ratio”**
Mean		174728.7	149488.2	38.6
Trimmed mean		171313.4	149450.6	38.6
Median		144100	142200	0
Modus		150000	150000	0
Variansi		207998959253.8	4124478778.1	2310.5
Standar deviasi		456069.0	64222.1	48.0
Mean absolute deviation		80079.9	50200.5	46.4
Persentil	**10%**	70000	70659	0
	25%	105500	105000	0
	50%	144100	142200	0
	61%	160000	160000	50
	75%	190000	185900	100
	90%	239748	234000	100
Ekstremum	**Maksimum**	30400000	615201	100
	Minimum	14000	15000	0
Jangkauan		30386000	600201	100
Jangkauan interkuartil		84500	80900	100

Tabel 3.1 Rekapitulasi informasi statistik data kuantitatif

Sebelum melakukan analisis statistik, perlu diingat bahwa data “salary” adalah nominal pendapatan dalam mata uang negara asal dan pada “salary_in_usd” telah dikonversi menjadi USD. Hal ini memicu adanya perbedaan yang cukup mencolok dalam informasi statistik antara kedua atribut. Standar deviasi “salary” patut dipertanyakan sebab nilainya yang terlalu besar hingga tidak masuk akal. Jangkauan “salary” juga terlalu besar sehingga patut dicurigai. Atribut “salary” yang nilai-nilai dari datanya belum dikonversi ke sebuah mata uang yang sejenis menimbulkan kekeliruan saat

dilakukan analisis data. Bisa jadi terdapat sebuah nilai mata uang yang sangat kecil dibanding dengan USD sehingga menyebabkan nominal pendapatan dalam mata uang tersebut sangat besar dibandingkan dalam USD dan sebaliknya. Ini menyebabkan informasi statistik "salary" sangat fluktuatif dan mencurigakan. Data ini harus dikonversi terlebih dahulu menjadi mata uang yang sama sehingga lebih akurat saat dianalisis.

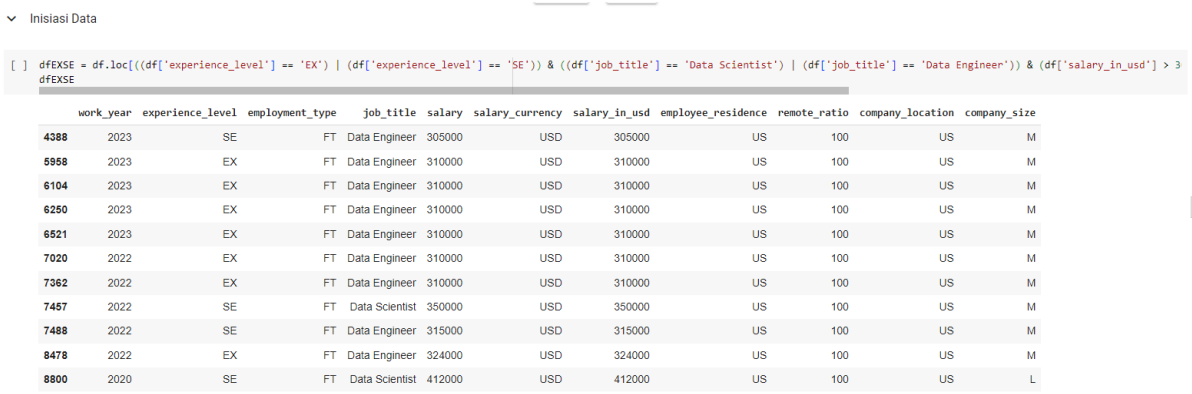
Data "salary_in_usd" sudah layak untuk dianalisis. Berdasarkan data ini, rata-rata pendapatan berada di sekitar 150 ribu USD dengan standar deviasi 64 ribu USD. Standar deviasi yang cukup besar ini menunjukkan besar pendapatan pekerjaan yang dikaji sangat bervariasi. Ini juga ditunjukkan dengan jangkauan yang cukup besar di sekitar 600 ribu USD. Modus data ini juga berada di sekitar nilai mean yang artinya data yang terekam memiliki persebaran yang cukup dekat dengan mean.

Data "remote_ratio" cukup unik karena walaupun datanya hanya ada 3 jenis, data tetap bisa diolah menjadi informasi statistik. Mean data ini yang berada di bawah atau sama dengan angka 38.6 berarti rata-rata pekerjaan cenderung memiliki sistem bekerja hybrid. Standar deviasi yang cukup besar juga menandakan data yang cukup fluktuatif, dengan kata lain, data tersebar di antara 3 kategori pada atribut ini, yakni pekerjaan tatap muka, hybrid, dan sepenuhnya jarak jauh.

3.3. Studi Kasus Statistik

Dalam studi kasus ini, dibuat statistik pendapatan mata uang asal berdasarkan beberapa ketentuan berikut:

1. experience_level: EX / SE
2. job_title: Data Scientist / Data Engineer
3. salary_in_usd > 300000
4. company_location: US
5. remote_ratio: 50 <= remote_ratio <= 100

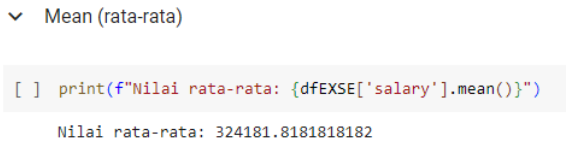


Inisiasi Data

```
[ ] dfEXSE = df.loc[((df['experience_level'] == 'EX') | (df['experience_level'] == 'SE')) & ((df['job_title'] == 'Data Scientist') | (df['job_title'] == 'Data Engineer')) & (df['salary_in_usd'] > 3
    dfEXSE
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
4388	2023	SE	FT	Data Engineer	305000	USD	305000	US	100	US	M
5958	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
6104	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
6250	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
6521	2023	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
7020	2022	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
7362	2022	EX	FT	Data Engineer	310000	USD	310000	US	100	US	M
7457	2022	SE	FT	Data Scientist	350000	USD	350000	US	100	US	M
7488	2022	SE	FT	Data Engineer	315000	USD	315000	US	100	US	M
8478	2022	EX	FT	Data Engineer	324000	USD	324000	US	100	US	M
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L

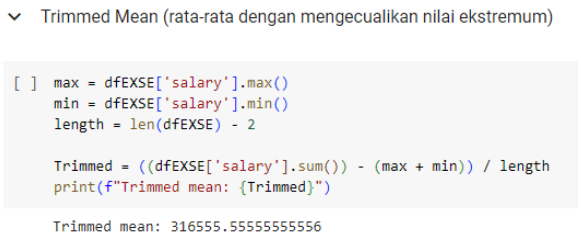
Gambar 3.20 Inisialisasi data studi kasus



Mean (rata-rata)

```
[ ] print(f"Nilai rata-rata: {dfEXSE['salary'].mean()}")

    Nilai rata-rata: 324181.8181818182
```

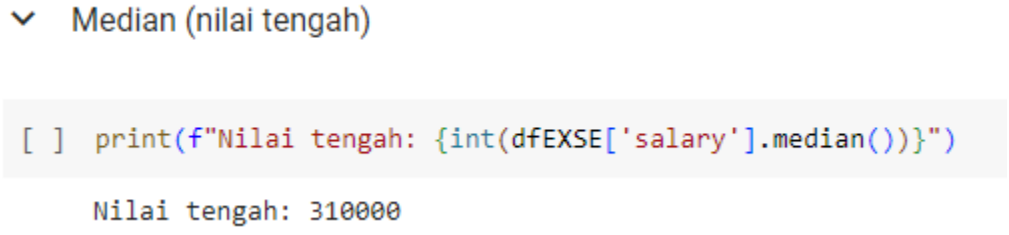
(a)



Trimmed Mean (rata-rata dengan mengecualikan nilai ekstremum)

```
[ ] max = dfEXSE['salary'].max()
    min = dfEXSE['salary'].min()
    length = len(dfEXSE) - 2

    Trimmed = ((dfEXSE['salary'].sum()) - (max + min)) / length
    print(f"Trimmed mean: {Trimmed}")

    Trimmed mean: 316555.55555555556
```

(b)



Median (nilai tengah)

```
[ ] print(f"Nilai tengah: {int(dfEXSE['salary'].median())}")

    Nilai tengah: 310000
```

(c)

Modus (nilai yang sering muncul)

```
[ ]  Modus = dfEXSE['salary'].mode().values[0]

     print(f"Nilai yang sering muncul: {Modus}")
```

```
Nilai yang sering muncul: 310000
```

(d)

Variance (Varians)

```
[ ]  print(f"Varians: {dfEXSE['salary'].var()}")
```

```
Varians: 1003763636.3636364
```

(e)

Standar Deviasi

```
[ ]  print(f"Standar deviasi: {dfEXSE['salary'].std()}")
```

```
Standar deviasi: 31682.229030856342
```

(f)

Mean Absolute Deviation (deviasi rata-rata absolut)

```
[ ]  print("Nilai deviasi rata-rata absolut: ", (dfEXSE['salary']-dfEXSE['salary'].mean()).abs().mean())
```

```
Nilai deviasi rata-rata absolut:  20661.157024793385
```

(g)

Percentile

```
[ ]  print(f"Persentil 10 %: {int(dfEXSE['salary'].quantile(0.1))}")
     print(f"Persentil 25 %: {int(dfEXSE['salary'].quantile(0.25))}")
     print(f"Persentil 50 %: {int(dfEXSE['salary'].quantile(0.5))}")
     print(f"Persentil 61 %: {int(dfEXSE['salary'].quantile(0.61))}")
     print(f"Persentil 75 %: {int(dfEXSE['salary'].quantile(0.75))}")
     print(f"Persentil 90 %: {int(dfEXSE['salary'].quantile(0.9))}")
```

```
Persentil 10 %: 310000
Persentil 25 %: 310000
Persentil 50 %: 310000
Persentil 61 %: 310500
Persentil 75 %: 319500
Persentil 90 %: 350000
```

(h)

Extremum

```
[ ]  print(f"Nilai maksimum: {dfEXSE['salary'].max()}")
     print(f"Nilai minimum : {dfEXSE['salary'].min()}")
```

```
Nilai maksimum: 412000
Nilai minimum : 305000
```

(i)

Range (jangkauan)

```
[ ]  # Perhitungan range
     range = dfEXSE.salary.max() - dfEXSE.salary.min()

     # Output
     print(f"Jangkauan data: {range}")
```

```
Jangkauan data: 107000
```

(j)

Interquartile Range (jangkauan interkuartil)

```
[ ]  Q1 = dfEXSE['salary'].quantile(0.25)
     Q3 = dfEXSE['salary'].quantile(0.75)

     IQR = Q3 - Q1
     print(f"Jangkauan interkuartil: {int(IQR)}")
```

```
Jangkauan interkuartil: 9500
```

(k)

Outlier (pencilan)

```
[ ]  # outlier < Q1 - 1.5(Q3-Q1)
     # outlier > Q3 + 1.5(Q3-Q1)

     Q1 = dfEXSE['salary'].quantile(0.25)
     Q3 = dfEXSE['salary'].quantile(0.75)

     batasBawah = Q1 - 1.5*(Q3-Q1)
     batasAtas = Q3 + 1.5*(Q3-Q1)

     # Outlier
     print("Outlier data: \n")
     dfEXSE.loc[(dfEXSE['salary'] < batasBawah) | (dfEXSE['salary'] > batasAtas)]
```

```
Outlier data:
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
7457	2022	SE	FT	Data Scientist	350000	USD	350000	US	100	US	M
8800	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	US	L

(l)

Distribusi Frekuensi Nilai

```
[ ]  freq_table = pd.crosstab(dfEXSE['salary'], 'frequency')

     freq_table
```

col_0	frequency
salary	
305000	1
310000	6
315000	1
324000	1
350000	1
412000	1

(m)

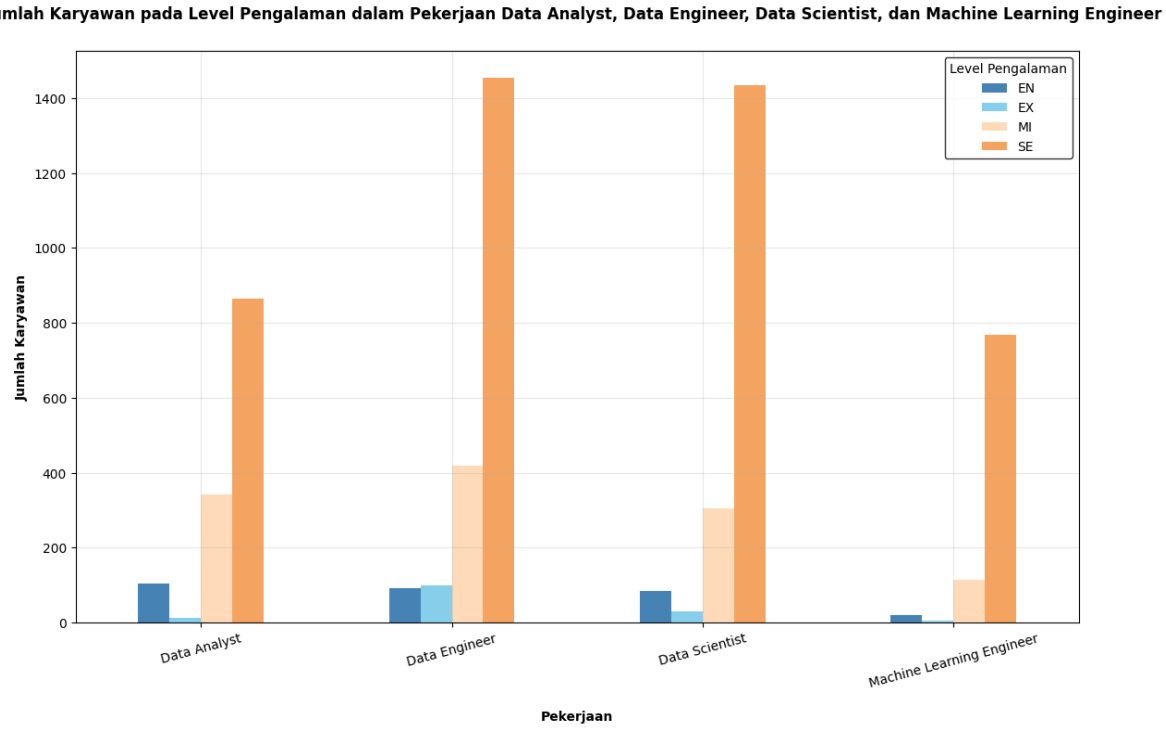
Gambar 3.21 *Syntax* program untuk menampilkan (a) mean, (b) *trimmed mean*, (c) median, (d) modus, (e) variansi, (f) standar deviasi, (g) *mean absolute deviation*, (h) persentil, (i) ekstremum, (j) jangkauan, (k) *interquartile range*, (l) *outlier*, dan (m) distribusi frekuensi nilai data untuk studi kasus

Dalam studi kasus ini, standar deviasi memiliki nilai yang lebih kecil lagi dibanding pada analisis berbagai atribut sebelumnya. Ini menjelaskan bahwa sebaran data menjadi lebih sempit. Ini terjadi karena telah diperlakukan penyaringan hingga 5 syarat yang mempersempit DataFrame. Walau sudah lebih spesifik, masih perlu diingat bahwa statistik ini didapatkan dari data "salary" yang merupakan pendapatan dalam mata uang asal yang belum diolah sehingga dapat menimbulkan kekeliruan.

BAB IV
VISUALISASI

4.1. Perbandingan Kategori

4.1.1. Grouped Bar Chart



Gambar 4.1 Visualisasi perbandingan antara jumlah pekerja pada pengalaman tertentu dalam pekerjaan Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer

Grafik bar diatas menampilkan perbandingan antara jumlah pekerja pada pengalaman tertentu dalam pekerjaan Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer. Pada grafik, sumbu x mewakili kategori pekerjaan dan sumbu y mewakili jumlah karyawan Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer.

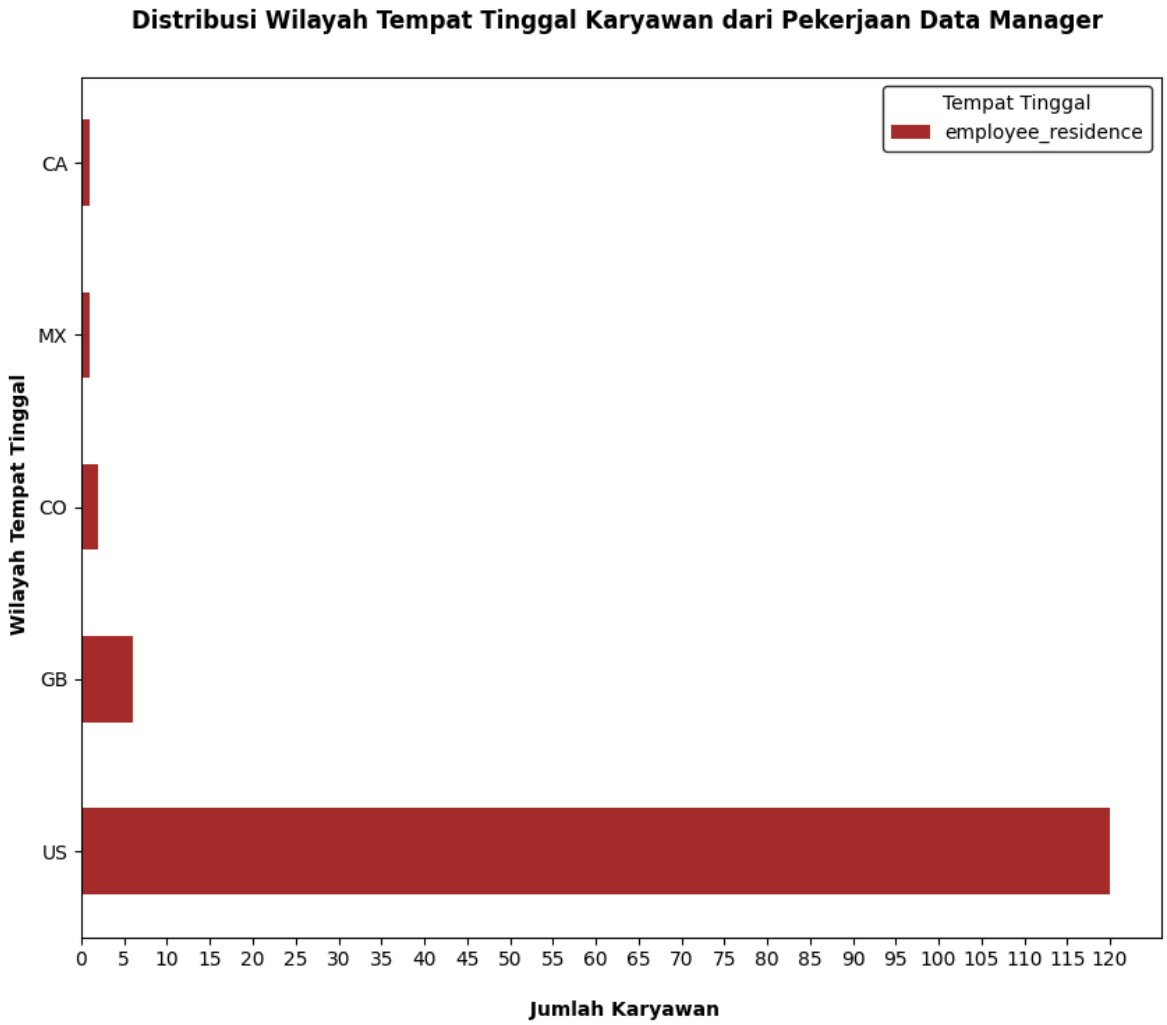
Arti pemilihan warna: Warna yang digunakan (dari kiri ke kanan) adalah warna dingin ke warna panas, yang menunjukkan tingkatan level pekerjaan dari EN (Entry-Level) sampai SE (Senior-Level)

Insight: Pada bar chart di atas, kita dapat melihat perbandingan antara jumlah pekerja pada pengalaman tertentu dalam pekerjaan Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer. Dapat disimpulkan, jumlah terbanyak berada pada level pengalaman 'SE' dan jumlah yang paling sedikit berada pada level pengalaman 'EX'

```
# Import libraries
from matplotlib import rcParams
import matplotlib as mpl
rcParams['figure.figsize'] = 14,8

# Grouping
dfwork = df.loc[(df['job_title'] == 'Data Scientist') | (df['job_title'] == 'Data Engineer') | (df['job_title'] == 'Data Analyst') | (df['job_title'] == 'Machine Learning Engineer')]

# Plotting
dfwork.groupby(['job_title', 'experience_level']).size().unstack().plot(kind='bar', color=['steelblue', 'skyblue', 'peachpuff', 'sandybrown'])

plt.style.use('default')
plt.grid(visible=True, alpha=0.3)
plt.title("Jumlah Karyawan pada Level Pengalaman dalam Pekerjaan Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer \n", loc='center', pad=10, fontsize=12, fontweight='bold', family='sans-serif');
plt.xlabel("\nPekerjaan", fontsize=10, fontweight='semibold', family='sans-serif')
plt.xticks(rotation=15)
plt.ylabel("\nJumlah Karyawan", fontsize=10, fontweight='semibold', family='sans-serif')
plt.legend(loc='upper right', title='Level Pengalaman', frameon=True, fancybox=True, draggable=True).get_frame().set_edgecolor('black')
```

Gambar 4.2 *Syntax* untuk menampilkan visualisasi perbandingan kategori antara jumlah pekerja pada pengalaman tertentu dalam pekerjaan Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer

4.1.2. Horizontal Bar Chart



Gambar 4.3 Visualisasi perbandingan antara distribusi wilayah tempat tinggal pekerja dari pekerjaan Data Manager

Grafik bar diatas menampilkan perbandingan antara distribusi wilayah tempat tinggal pekerja dari pekerjaan Data Manager. Pada sumbu x mewakili jumlah karyawan setiap wilayah dan sumbu y mewakili kategori wilayah tempat tinggal.

Arti pemilihan warna: Warna yang digunakan adalah warna coklat yang cukup kontras, untuk menegaskan bar chart dan juga memudahkan pembaca dalam menganalisis grafik.

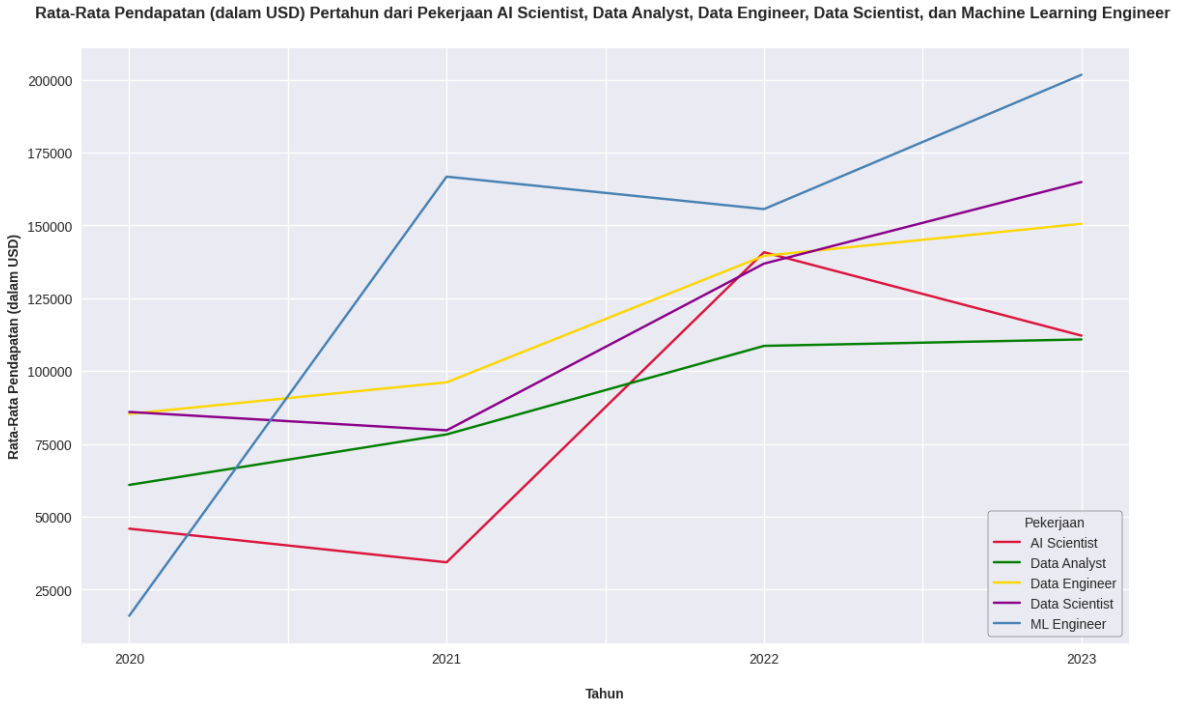
Insight: Pada horizontal bar chart di atas, dapat kita lihat distribusi wilayah tempat tinggal karyawan yang bekerja sebagai Data Manager. Jumlah karyawan terbanyak terletak di wilayah US, sedangkan jumlah karyawan yang paling sedikit terletak di dua daerah, yakni wilayah CA dan wilayah MX.

```
# Import Libraries
from matplotlib import rcParams
import matplotlib as mpl
rcParams['figure.figsize'] = 10,8

# Grouping
dfh = df.loc[(df['job_title'] == 'Data Manager')]
dfh['employee_residence'].value_counts().plot(kind='barh', color='brown')

# Plotting
plt.style.use('default')
plt.title("Distribusi Wilayah Tempat Tinggal Karyawan dari Pekerjaan Data Manager \n", loc='center', pad=10, fontsize=12, fontweight='bold', family='sans-serif');
plt.xlabel("\n Jumlah Karyawan", fontsize=10, fontweight='semibold', family='sans-serif')
plt.xticks([0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100,105,110,115,120])
plt.ylabel("\n Wilayah Tempat Tinggal", fontsize=10, fontweight='semibold', family='sans-serif')
plt.legend(loc='upper right', title='Tempat Tinggal', frameon=True).get_frame().set_edgecolor('black')
```

Gambar 4.4 *Syntax* untuk menampilkan visualisasi perbandingan antara distribusi wilayah tempat tinggal pekerja dari pekerjaan Data Manager

4.2. Penampilan Perubahan Terhadap Waktu

4.2.1. Line Chart



Gambar 4.5 Visualisasi perubahan rata-rata pendapatan pertahun

Grafik garis diatas menampilkan perubahan rata-rata pendapatan pertahun. Pada sumbu x mewakili tahun dan sumbu y mewakili jumlah rata-rata pendapatan.

Arti pemilihan warna :

1. Biru: Menunjukkan tren positif (kenaikan) secara tajam
2. Ungu: Menunjukkan trend positif (kenaikan), namun tidak konsisten (terdapat penurunan di suatu waktu)
3. Kuning: Menunjukkan trend yang cukup netral (memiliki sedikit kenaikan)
4. Hijau: Menunjukkan trend yang hampir netral (memiliki sangat sedikit kenaikan)
5. Merah: Menunjukkan trend negatif (penurunan) secara tajam

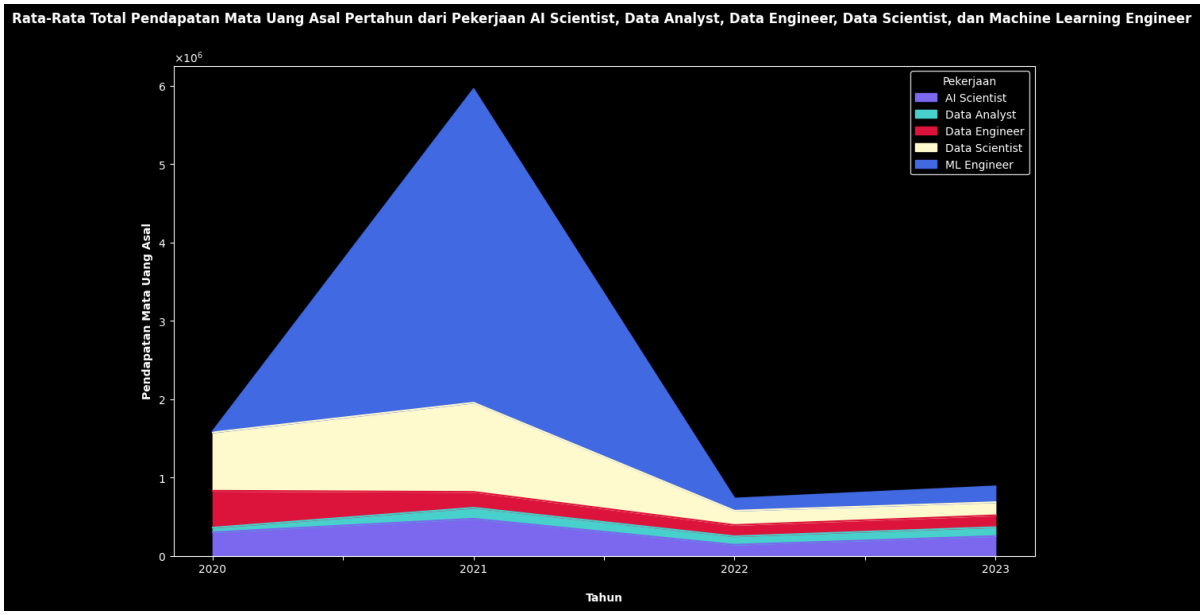
secara umum, warna biru menunjukkan adanya trend positif sedangkan warna merah menunjukkan adanya trend negatif, dengan warna di tengahnya menunjukkan suatu kondisi trend yang netral

Insight: Pada line chart di atas, dapat kita lihat perubahan nilai terhadap waktu dari rata-rata pendapatan mata uang asal pertahun dari pekerjaan AI Scientist, Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer. Kenaikan tajam pendapatan mata uang asal hingga pada tahun 2023 terjadi pada pekerjaan Machine Learning Engineer, sedangkan penurunan tajam pendapatan mata uang asal hingga pada tahun 2023 terjadi pada pekerjaan AI Scientist.

```
# Import Libraries
from matplotlib import rcParams
import matplotlib as mpl
rcParams['figure.figsize'] = 14,8

# Grouping
df['work_year'] = pd.Categorical(df.work_year)
dfline = df.loc[(df['job_title'] == 'Data Analyst') | (df['job_title'] == 'Data Scientist') | (df['job_title'] == 'Data Engineer') | (df['job_title'] == 'ML Engineer') | (df['job_title'] == 'AI Scientist')]

dfline.groupby(['work_year', 'job_title'])['salary_in_usd'].mean().unstack().plot(kind='line', color=['crimson', 'green', 'gold', 'darkmagenta', 'steelblue'], grid=True)

# Plotting
plt.style.use('seaborn-v0_8');
plt.title("Rata-Rata Pendapatan (dalam USD) Pertahun dari Pekerjaan AI Scientist, Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer \n", loc='center', pad=10, fontsize=12, fontweight='bold', family='sans-serif');
plt.xlabel("\nTahun", fontsize=10, fontweight='semibold', family='sans-serif');
plt.ylabel("\nRata-Rata Pendapatan (dalam USD)", fontsize=10, fontweight='semibold', family='sans-serif');
plt.legend(loc='lower right', title='Pekerjaan', frameon=True).get_frame().set_edgecolor('black');
```

Gambar 4.6 *Syntax* untuk menampilkan visualisasi perubahan rata-rata pendapatan pertahun

4.2.2. Stacked Area Chart



Gambar 4.7 Visualisasi perubahan rata rata total pendapatan mata uang asal pertahun

Grafik area diatas menampilkan perubahan rata rata total pendapatan mata uang asal pertahun. Pada sumbu x mewakili tahun dan sumbu y mewakili jumlah pendapatan mata uang asal.

Arti pemilihan warna : Warna yang digunakan adalah warna pastel yang cukup kontras. Warna-warna ini menunjukkan dan menekankan perbedaan level pengalaman pada setiap stacked area chart, dan juga memudahkan pembaca dalam menganalisis grafik karena warna yang ditampilkan mudah terlihat dan mudah diidentifikasi.

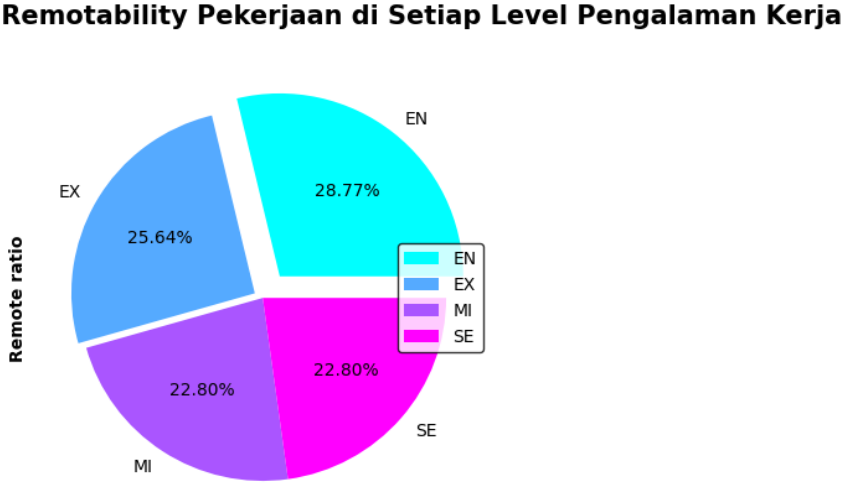
Insight: Pada area chart di atas, dapat kita lihat total nilai terhadap waktu dari rata-rata pendapatan mata uang asal dalam rentang tahun 2020-2023 dari pekerjaan AI Scientist, Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer. Pekerjaan Machine Learning Engineer memiliki pendapatan rata-rata total tertinggi dibandingkan dengan pekerjaan lain, sedangkan pekerjaan AI Scientist memiliki pendapatan rata-rata total terendah dibandingkan dengan pekerjaan lain.

```
# Import Libraries
from matplotlib import rcParams
import matplotlib as mpl
rcParams['figure.figsize'] = 14,8

# Grouping
df['work_year'] = pd.Categorical(df.work_year)
dfline = df.loc[(df['job_title'] == 'Data Analyst') | (df['job_title'] == 'Data Scientist') | (df['job_title'] == 'Data Engineer') | (df['job_title'] == 'ML Engineer') | (df['job_title'] == 'AI Scientist')]

dfline.groupby(['work_year', 'job_title'])['salary'].mean().unstack().plot(kind='area', color=['mediumslateblue', 'mediumturquoise', 'crimson', 'lemonchiffon', 'royalblue'], grid=False)

# Plotting
plt.style.use('dark_background');
plt.title("Rata-Rata Total Pendapatan Mata Uang Asal Pertahun dari Pekerjaan AI Scientist, Data Analyst, Data Engineer, Data Scientist, dan Machine Learning Engineer \n", loc='center', pad=10, fontsize=12, fontweight='bold', family='sans-serif');
plt.xlabel("\nTahun", fontsize=10, fontweight='semibold', family='sans-serif');
plt.ticklabel_format(axis='y', style='sci', scilimits=(6,6), useMathText=True);
plt.ylabel("\nRata-Rata Pendapatan Mata Uang Asal", fontsize=10, fontweight='semibold', family='sans-serif');
plt.legend(loc='upper right', title='Pekerjaan', frameon=True).get_frame().set_edgecolor('white');
```

Gambar 4.8 *Syntax* untuk menampilkan visualisasi perubahan rata rata total pendapatan mata uang asal pertahun

4.3. Penampilan Hierarki dan Hubungan Keseluruhan-Bagian

4.3.1. Pie Chart



Gambar 4.9 Visualisasi persentase rata-rata remotability pekerjaan pada pengalaman tertentu

Grafik pie di atas menampilkan persentase rata-rata remotability pekerjaan pada pengalaman tertentu.

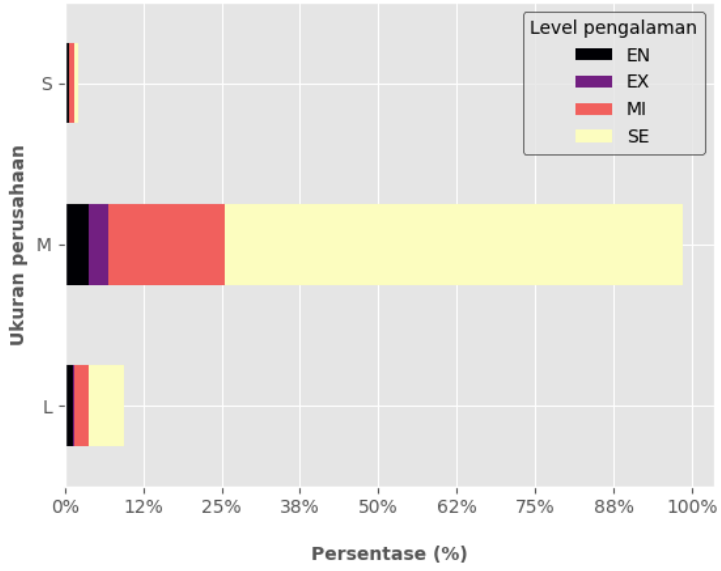
Arti pemilihan warna: Warna yang digunakan merupakan warna dari warna dingin ke warna panas, yang menyatakan level pengalaman kerja dari EN sampai SE (Entry-level - Senior-level).

Insight: Dapat kita lihat bahwa rata-rata dari remote ratio tertinggi berada pada pengalaman EN, yang ditegaskan juga dengan partisi pie nya yang terpisah dari pie chart, disusul dengan pengalaman EX yang ditegaskan dengan partisi pie nya yang terpisah cukup jauh dari pie chart. Karena rata-rata pada pengalaman MI dan SE sama besar, kedua partisi menyatu.

```
# Import Libraries
from matplotlib import rcParams
import matplotlib as mpl
rcParams['figure.figsize'] = 14,8

# Grouping
df.groupby('experience_level')['remote_ratio'].mean().plot(kind='pie', autopct='%.2f%%', explode=[.15, .05, 0, 0]
                                                           , cmap='cool')


# Plotting
plt.legend(loc='center right', frameon=True).get_frame().set_edgecolor('black')
plt.ylabel('Remote ratio', fontsize=15, fontweight='semibold', family='sans-serif')
plt.title("Persentase Rata-Rata Remotability Pekerjaan di Setiap Level Pengalaman Kerja \n", loc='center', pad=10, fontsize=15, fontweight='bold', family='sans-serif');
```

Gambar 4.10 *Syntax* untuk menampilkan visualisasi persentase rata-rata remotability pekerjaan pada pengalaman tertentu

4.3.2. Stacked Horizontal Bar Chart



Gambar 4.11 Visualisasi persentase jumlah level pengalaman kerja pada suatu ukuran perusahaan

Grafik diatas menunjukkan persentase jumlah level pengalaman kerja pada suatu ukuran perusahaan. Pada sumbu x mewakili besar persentase dan sumbu y mewakili ukuran perusahaan.

Arti pemilihan warna: Warna yang digunakan merupakan warna dari gelap ke terang, yang menyatakan level pengalaman dari EN sampai SE (Entry-level - Senior-level)

Insight : Pada stacked bar chart di atas, terlihat bar yang dibagi menjadi beberapa partisi, yakni 4 partisi. Masing-masing partisi memiliki warna yang berbeda (yakni dari warna gelap ke warna cerah) yang menunjukkan level pangalaman dari EN hingga SE.

Dapat dilihat, jumlah karyawan dengan level pengalaman SE terbanyak bekerja di perusahaan dengan ukuran M (97%-25%=72%), sedangkan jumlah karyawan dengan level pengalaman SE paling sedikit

bekerja di perusahaan dengan ukuran S. Analisis juga dapat dilakukan pada level pengalaman lain dengan melihat besar persentase dari setiap stacked bar.

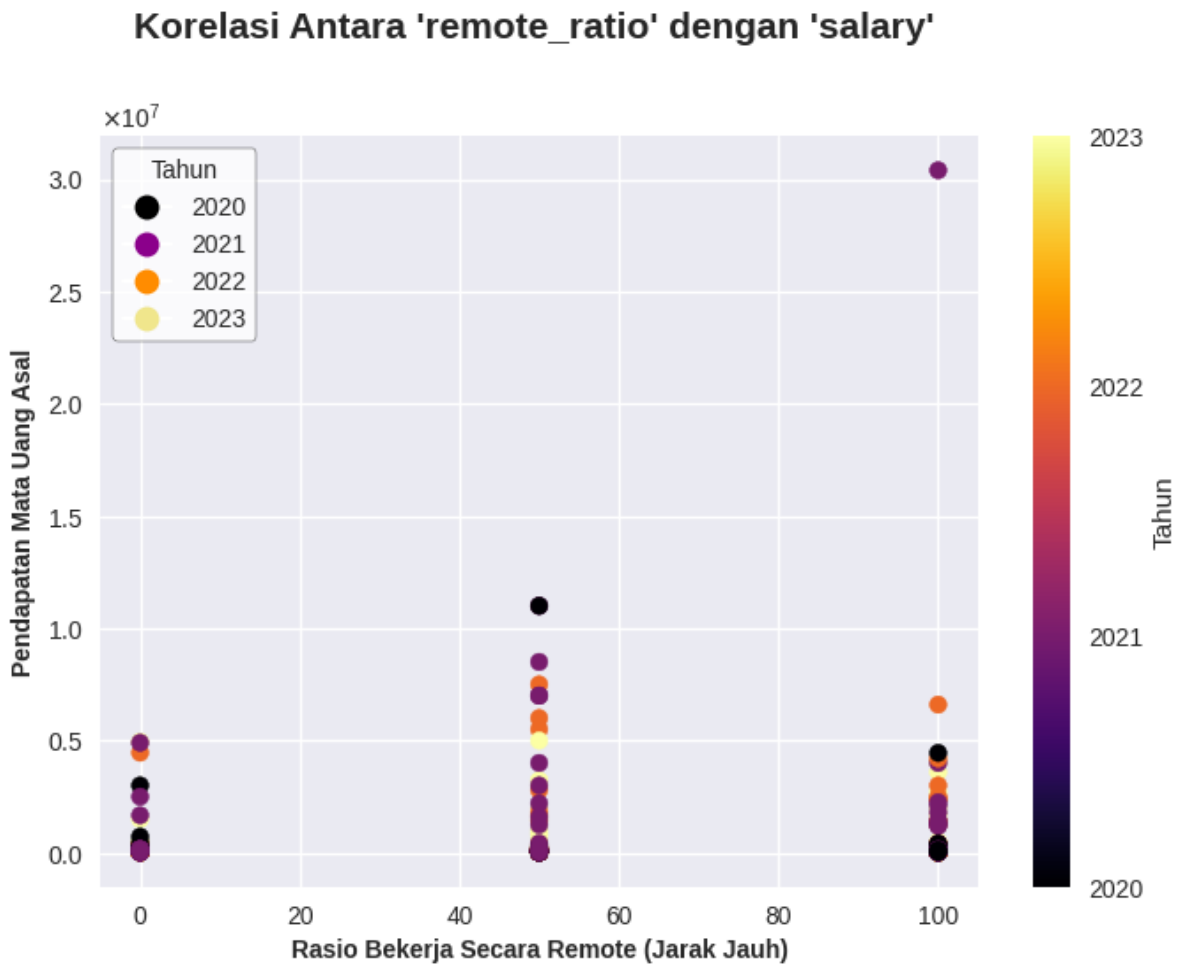
Untuk menghitung nilai persentase, kurangi batas atas dan batas bawah dari tiap partisi bar.

```
# Import Libraries
from matplotlib import rcParams
import matplotlib as mpl
import matplotlib.ticker as tck
rcParams['figure.figsize'] = 14,8

# Grouping
ax = df.groupby(['company_size', 'experience_level']).size().unstack().plot(kind='barh', stacked=True, cmap='magma')
ax.xaxis.set_major_formatter(tck.PercentFormatter(xmax=8000))

# Plotting
plt.style.use('ggplot')
plt.title("Persentase Jumlah Karyawan di Setiap Level Pengalaman Kerja pada Ukuran Perusahaan Kecil (S), Sedang(M), dan Besar(L) \n", loc='center', pad=10, fontsize=15, fontweight='bold', family='sans-serif');
plt.xlabel("\nPersentase (%)", fontsize=10, fontweight='semibold', family='sans-serif')
plt.ylabel("\nUkuran perusahaan", fontsize=10, fontweight='semibold', family='sans-serif')
plt.legend(loc='upper right', title='Level pengalaman', frameon=True).get_frame().set_edgecolor('black')
```

Gambar 4.12 *Syntax* untuk menampilkan visualisasi persentase jumlah level pengalaman kerja pada suatu ukuran perusahaan

4.4. *Plotting Relationship*

4.4.1. Scatter Plot



Gambar 4.13 Visualisasi korelasi antara rasio bekerja jarak jauh dengan jumlah pendapatan mata uang asal dalam rentang tahun 2020-2023

Scatter plot diatas menunjukkan korelasi antara rasio bekerja jarak jauh dengan jumlah pendapatan mata uang asal dalam rentang tahun 2020-2023. Pada sumbu x mewakili rasio bekerja jarak jauh dan sumbu y mewakili pendapatan mata uang asal.

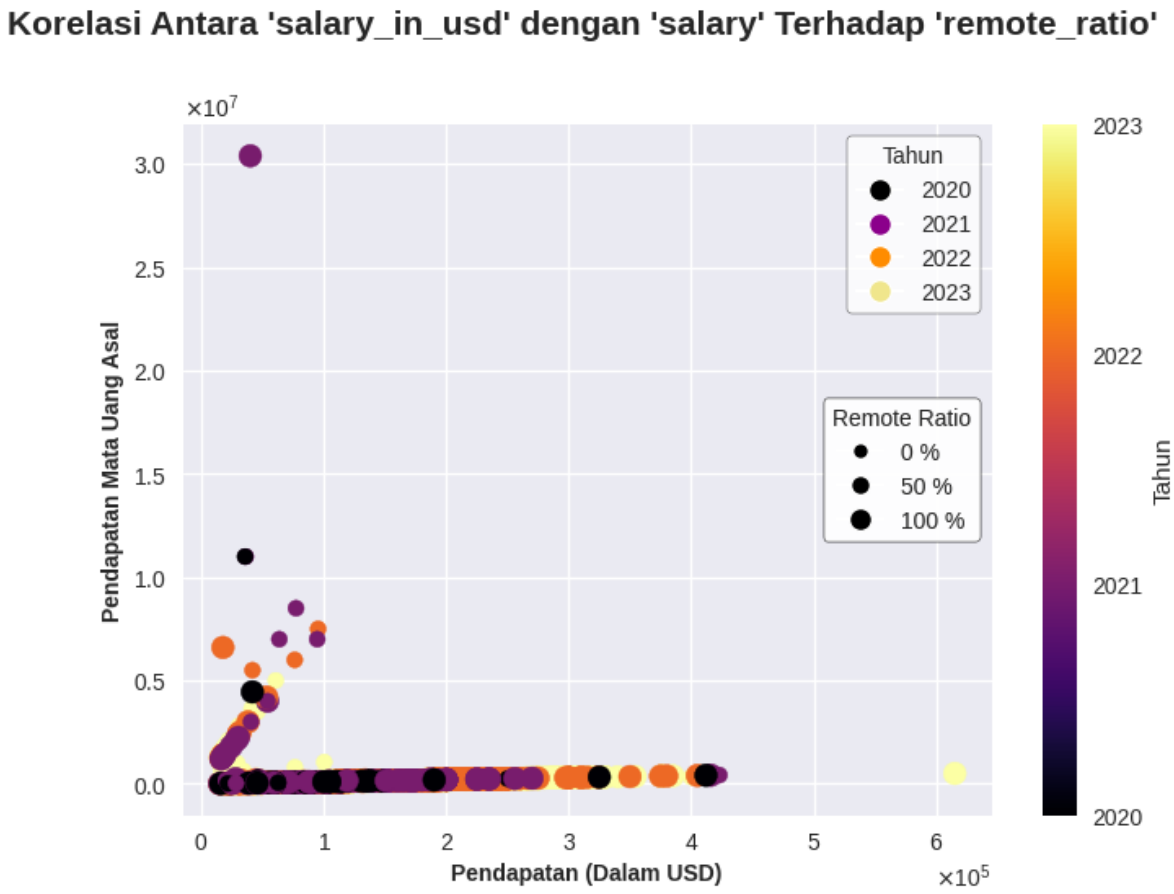
Arti pemilihan warna: Warna yang digunakan merupakan warna dari gelap ke terang, yang menyatakan jumlah dari frekuensi munculnya tahun dari pendapatan tersebut. Semakin cerah maka jumlahnya semakin banyak (tahun 2023 muncul paling banyak pada dataset). Sebaliknya, semakin gelap maka jumlahnya semakin sedikit. (tahun 2020 muncul paling sedikit pada dataset)

Insight: Pada scatter plot di atas, jika kita menganalisis grafik atau melakukan regresi linear terhadap grafik, maka didapatkan bahwa kemiringan garis regresi bernilai positif tetapi sangat kecil dan mendekati 0, menunjukkan bahwa atribut 'remote_ratio' dan 'salary' tidak berkorelasi.

```
# Visualisasi
from matplotlib import rcParams
import matplotlib as mpl
import matplotlib.lines as mln
rcParams['figure.figsize'] = 12,8

legend_elements = [mln.Line2D([0],[0], marker='o', color='w',label='2020', markerfacecolor='black', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2021', markerfacecolor='darkmagenta', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2022', markerfacecolor='darkorange', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2023', markerfacecolor='khaki', markersize=10)]

# Plotting
plt.style.use('seaborn-v0_8');
plt.scatter(df.remote_ratio, df.salary, c=df.work_year, cmap='inferno');
plt.title("Korelasi Antara 'remote_ratio' dengan 'salary' \n", loc='center', pad=10, fontsize=15, fontweight='bold', family='sans-serif');
plt.xlabel('Rasio Bekerja Secara Remote (Jarak Jauh)', fontsize=10, fontweight='bold');
plt.ylabel('Pendapatan Mata Uang Asal', fontsize=10, fontweight='bold');
plt.ticklabel_format(axis='y', style='sci', scilimits=(7,7), useMathText=True);
legend = plt.legend(handles=legend_elements, loc='upper left', title='Tahun', frameon=True);
frame = legend.get_frame()
frame.set_facecolor('white')
frame.set_edgecolor('black')
plt.colorbar(label='Tahun', ticks=[2020, 2021, 2022, 2023]);
```

Gambar 4.14 *Syntax* untuk menampilkan visualisasi korelasi antara rasio bekerja jarak jauh dengan jumlah pendapatan mata uang asal dalam rentang tahun 2020-2023

4.4.2. Bubble Plot

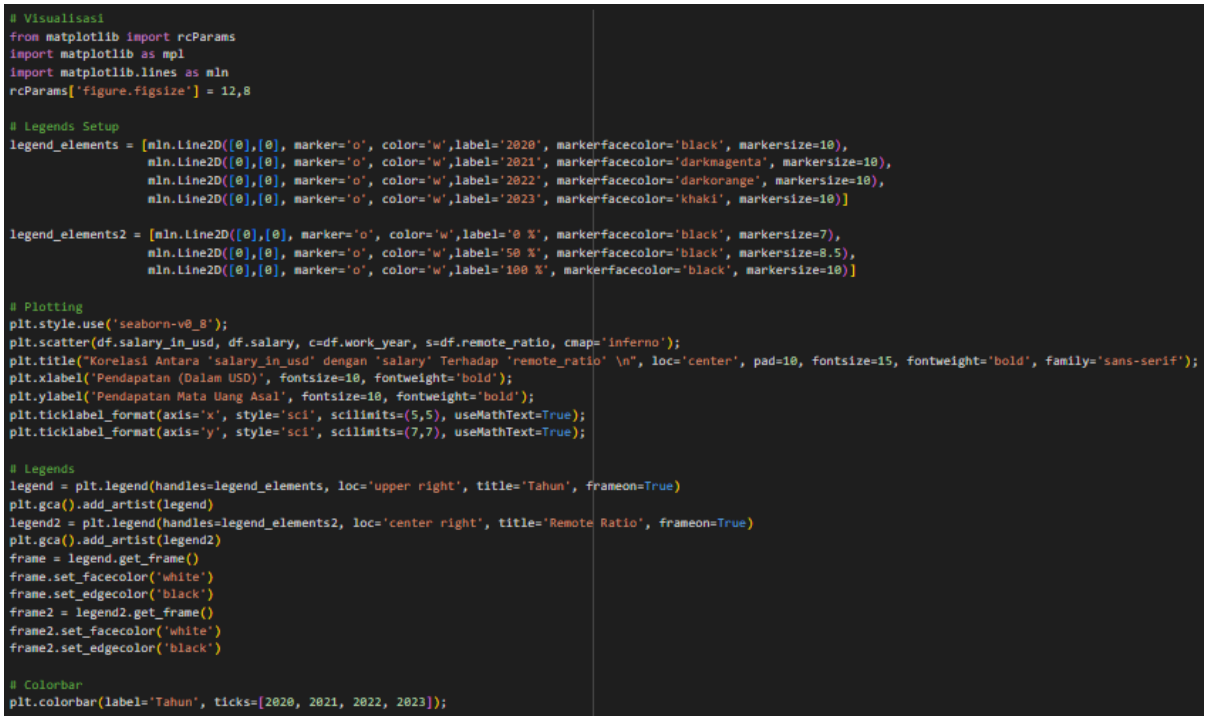


Gambar 4.15 Visualisasi korelasi antara pendapatan mata uang asal dengan pendapatan (dalam USD) terhadap rasio bekerja jarak jauh dalam rentang tahun 2020-2023

Bubble plot diatas menunjukkan korelasi antara pendapatan mata uang asal dengan pendapatan (dalam USD) terhadap rasio bekerja jarak jauh dalam rentang tahun 2020-2023. Pada sumbu x mewakili jumlah pendapatan mata uang asal dan sumbu y mewakili jumlah pendapatan (dalam USD).

Arti pemilihan warna: Warna yang digunakan merupakan warna dari gelap ke terang, yang menyatakan jumlah dari frekuensi munculnya tahun dari pendapatan tersebut. Semakin cerah maka jumlahnya semakin banyak (tahun 2023 muncul paling banyak pada dataset). Sebaliknya, semakin gelap maka jumlahnya semakin sedikit. (tahun 2020 muncul paling sedikit pada dataset)

Insight: Pada bubble plot di atas, jika kita menganalisis grafik atau melakukan regresi linear terhadap grafik, maka didapatkan bahwa kemiringan garis regresi bernilai positif tetapi sangat kecil dan mendekati 0, menunjukkan bahwa atribut 'remote_ratio' dan 'salary' tidak berkorelasi. Sedangkan atribut 'remote_ratio' menunjukkan tingkat remote ratio dari suatu pekerjaan, semakin besar ukuran bubble plot, semakin tinggi ratio dari remotability pekerjaan.



```
# Visualisasi
from matplotlib import rcParams
import matplotlib as mpl
import matplotlib.lines as mln
rcParams['figure.figsize'] = 12,8

# Legends Setup
legend_elements = [mln.Line2D([0],[0], marker='o', color='w',label='2020', markerfacecolor='black', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2021', markerfacecolor='darkmagenta', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2022', markerfacecolor='darkorange', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2023', markerfacecolor='khaki', markersize=10)]

legend_elements2 = [mln.Line2D([0],[0], marker='o', color='w',label='0 %', markerfacecolor='black', markersize=7),
                   mln.Line2D([0],[0], marker='o', color='w',label='50 %', markerfacecolor='black', markersize=8.5),
                   mln.Line2D([0],[0], marker='o', color='w',label='100 %', markerfacecolor='black', markersize=10)]

# Plotting
plt.style.use('seaborn-v0_8');
plt.scatter(df.salary_in_usd, df.salary, c=df.work_year, s=df.remote_ratio, cmap='inferno');
plt.title("Korelasi Antara 'salary_in_usd' dengan 'salary' Terhadap 'remote_ratio' \n", loc='center', pad=10, fontsize=15, fontweight='bold', family='sans-serif');
plt.xlabel('Pendapatan (Dalam USD)', fontsize=10, fontweight='bold');
plt.ylabel('Pendapatan Mata Uang Asal', fontsize=10, fontweight='bold');
plt.ticklabel_format(axis='x', style='sci', scilimits=(5,5), useMathText=True);
plt.ticklabel_format(axis='y', style='sci', scilimits=(7,7), useMathText=True);

# Legends
legend = plt.legend(handles=legend_elements, loc='upper right', title='Tahun', frameon=True)
plt.gca().add_artist(legend)
legend2 = plt.legend(handles=legend_elements2, loc='center right', title='Remote Ratio', frameon=True)
plt.gca().add_artist(legend2)
frame = legend.get_frame()
frame.set_facecolor('white')
frame.set_edgecolor('black')
frame2 = legend2.get_frame()
frame2.set_facecolor('white')
frame2.set_edgecolor('black')

# Colorbar
plt.colorbar(label='Tahun', ticks=[2020, 2021, 2022, 2023]);
```

Gambar 4.16 *Syntax* untuk menampilkan visualisasi korelasi antara pendapatan mata uang asal dengan pendapatan (dalam USD) terhadap rasio bekerja jarak jauh dalam rentang tahun 2020-2023

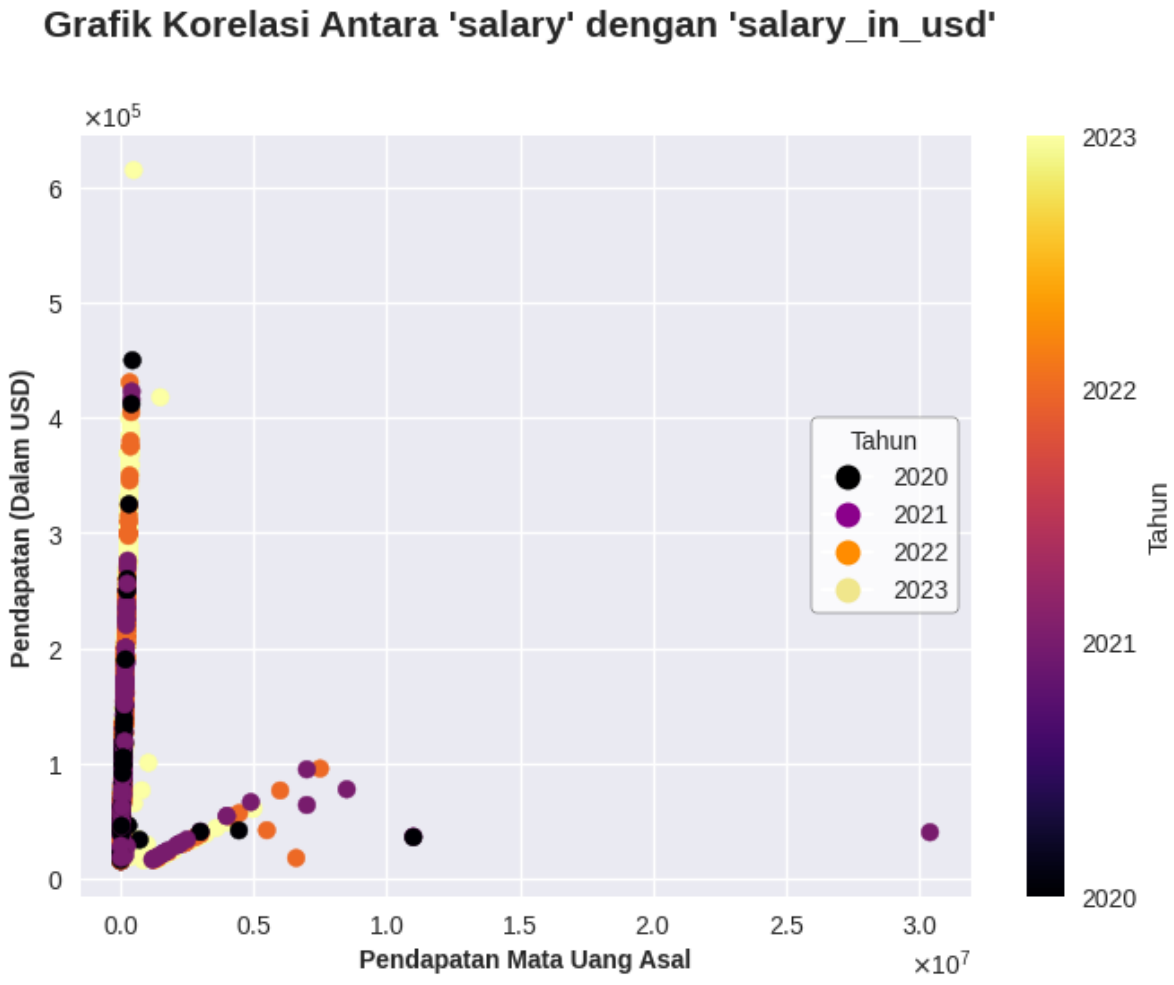
BAB V
KORELASI

Korelasi data dinyatakan dengan deskripsi sebagai berikut.

1. Jika nilai korelasi mendekati 0, maka kedua atribut tidak berkorelasi.
2. Jika nilai korelasi mendekati 1, maka kedua atribut berkorelasi positif (lurus). Menandakan bahwa jika salah satu atribut membesar, maka atribut lain juga akan ikut membesar. Berlaku sebaliknya, yakni jika salah satu atribut mengecil, maka atribut lain juga akan ikut mengecil.
3. Jika nilai korelasi mendekati -1, maka kedua atribut berkorelasi negatif (berkebalikan). Menandakan bahwa jika salah satu atribut membesar, maka atribut lain akan mengecil. Berlaku sebaliknya, yakni jika salah satu atribut mengecil, maka maka atribut lain akan membesar.

5.1. Korelasi Antara 'salary' dengan 'salary_in_usd'

Nilai korelasi antara 'salary' dan 'salary_in_usd' adalah 0.04984095666541916. Karena nilai berkorelasi sekitar 0.049 dan mendekati 0, maka kedua atribut tidak berkorelasi.



Gambar 5.1 Visualisasi korelasi antara 'salary' dengan 'salary_in_usd'

Pada grafik korelasi di atas, jika kita melakukan regresi linear terhadap grafik, maka didapatkan bahwa kemiringan garis regresi bernilai positif tetapi sangat kecil dan mendekati 0, menunjukkan bahwa atribut 'salary' dan 'salary_in_usd' tidak berkorelasi.

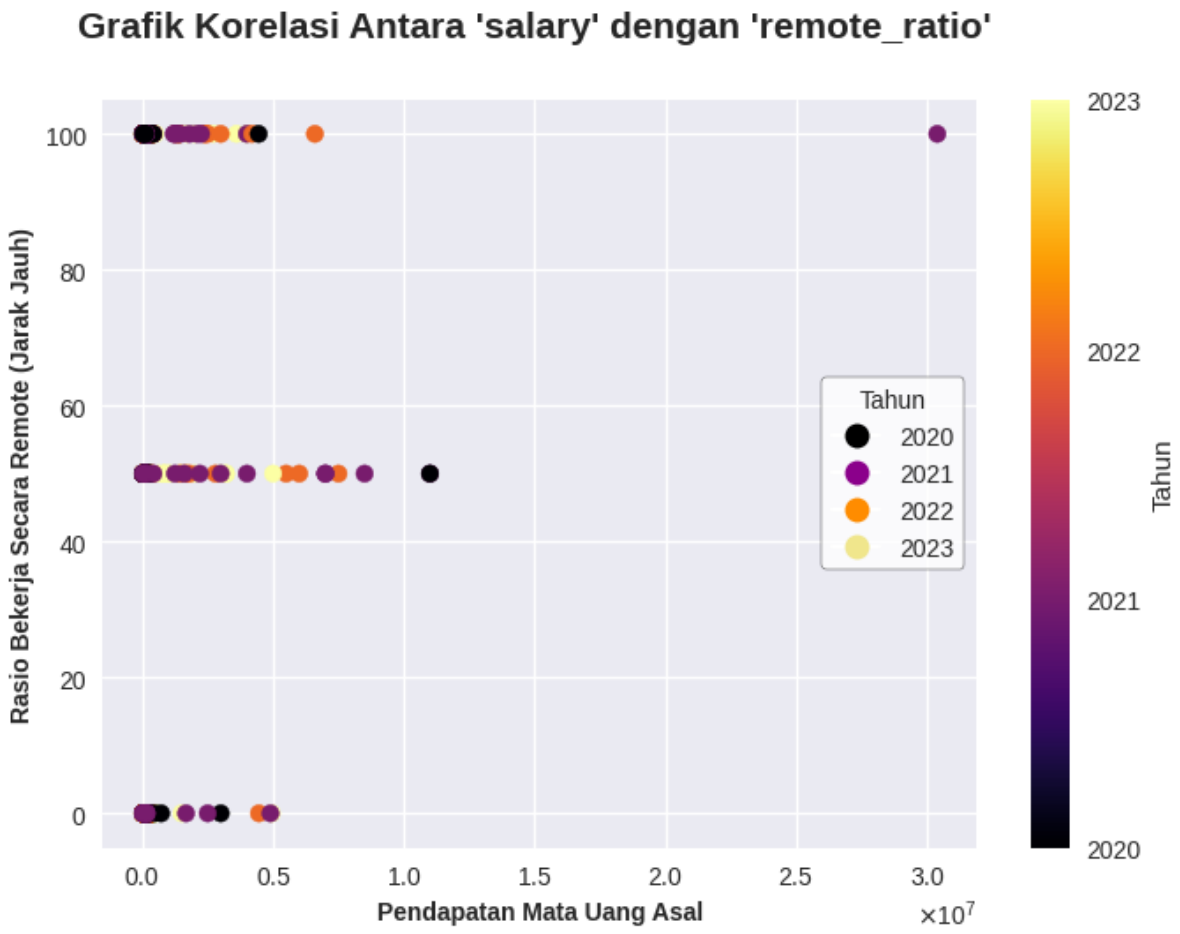
Arti pemilihan warna: Warna yang digunakan merupakan warna dari gelap ke terang, yang menyatakan jumlah dari frekuensi munculnya tahun dari pendapatan tersebut. Semakin cerah maka jumlahnya semakin banyak (tahun 2023 muncul paling banyak pada dataset). Sebaliknya, semakin gelap maka jumlahnya semakin sedikit. (tahun 2020 muncul paling sedikit pada dataset)

```
# Nilai Korelasi
corr1 = df['salary'].corr(df['salary_in_usd'])

print(f"\n Nilai korelasi antara 'salary' dan 'salary_in_usd' adalah {corr1} \n")
print("# Karena nilai berkorelasi sekitar 0.049 dan mendekati 0, maka kedua atribut tidak berkorelasi. \n")
print("Grafik korelasi: \n")

# Visualisasi
from matplotlib import rcParams
import matplotlib as mpl
import matplotlib.lines as mln
rcParams['figure.figsize'] = 12,8

legend_elements = [mln.Line2D([0],[0], marker='o', color='w',label='2020', markerfacecolor='black', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2021', markerfacecolor='darkmagenta', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2022', markerfacecolor='darkorange', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2023', markerfacecolor='khaki', markersize=10)]

# Plotting
plt.style.use('seaborn-v0_8');
plt.scatter(df.salary, df.salary_in_usd, c=df.work_year, cmap='inferno');
plt.title("Grafik Korelasi Antara 'salary' dengan 'salary_in_usd' \n", loc='center', pad=10, fontsize=15, fontweight='bold', family='sans-serif');
plt.xlabel('Pendapatan Mata Uang Asal', fontsize=10, fontweight='bold');
plt.ylabel('Pendapatan (Dalam USD)', fontsize=10, fontweight='bold');
plt.ticklabel_format(axis='y', style='sci', scilimits=(5,5), useMathText=True);
plt.ticklabel_format(axis='x', style='sci', scilimits=(7,7), useMathText=True);
legend = plt.legend(handles=legend_elements, loc='center right', title='Tahun', frameon=True);
frame = legend.get_frame()
frame.set_facecolor('white')
frame.set_edgecolor('black')
plt.colorbar(label='Tahun', ticks=[2020, 2021, 2022, 2023]);
```

Gambar 5.2 *Syntax* untuk menampilkan nilai korelasi dan visualisasi korelasi antara 'salary' dengan 'salary_in_usd'

5.2. Korelasi Antara 'salary' dengan 'remote_ratio'

Nilai korelasi antara 'salary' dan 'remote_ratio' adalah 0.019284988739124296. Karena nilai berkorelasi sekitar 0.019 dan mendekati 0, maka kedua atribut tidak berkorelasi.



Gambar 5.3 Visualisasi korelasi antara 'salary' dengan 'remote_ratio'

Pada grafik korelasi di atas, jika kita melakukan regresi linear terhadap grafik, maka didapatkan bahwa kemiringan garis regresi bernilai positif tetapi sangat kecil dan mendekati 0, menunjukkan bahwa atribut 'salary' dan 'remote_ratio' tidak berkorelasi.

Arti pemilihan warna: Warna yang digunakan merupakan warna dari gelap ke terang, yang menyatakan jumlah dari frekuensi munculnya tahun dari pendapatan tersebut. Semakin cerah maka jumlahnya semakin banyak (tahun 2023 muncul paling

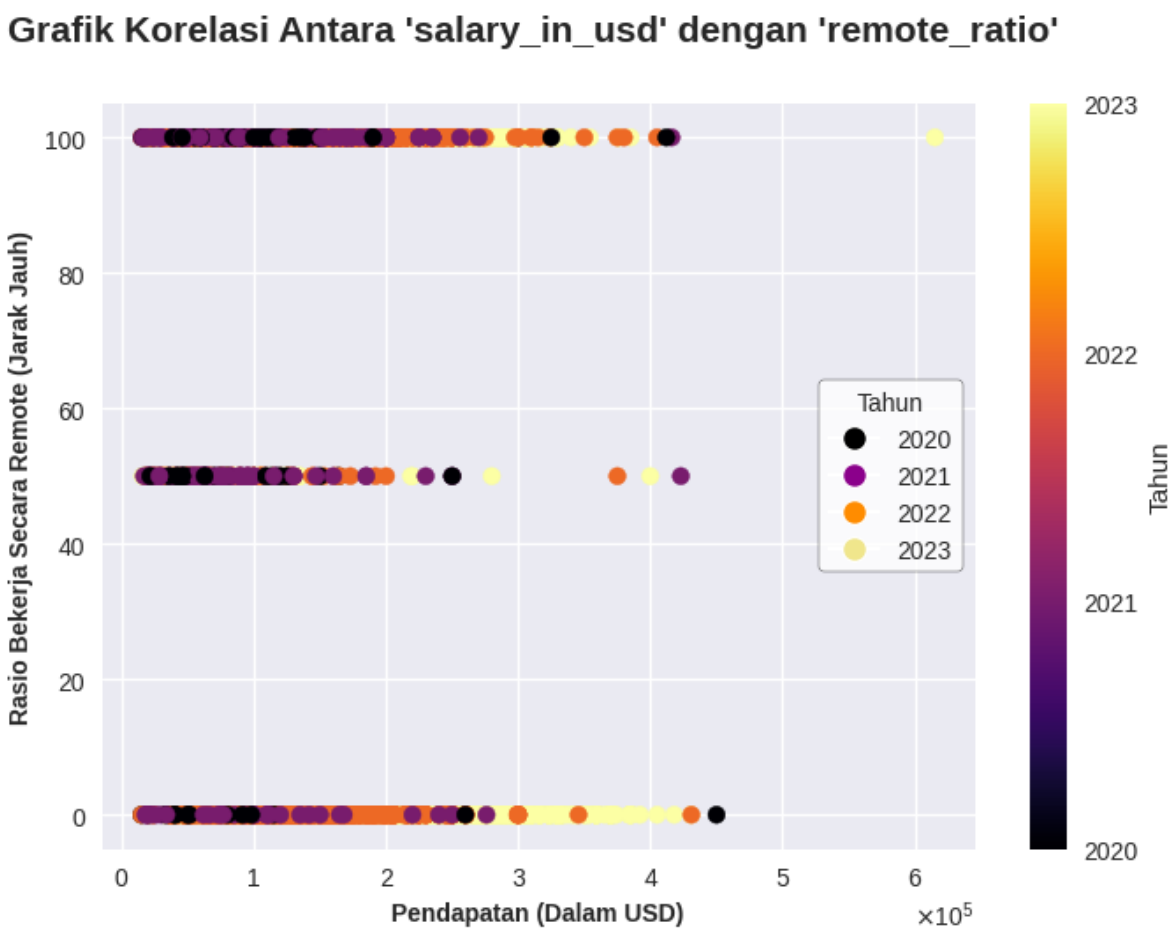
banyak pada dataset). Sebaliknya, semakin gelap maka jumlahnya semakin sedikit. (tahun 2020 muncul paling sedikit pada dataset)

```
# Nilai Korelasi
corr2 = df['salary'].corr(df['remote_ratio'])

print(f"\n Nilai korelasi antara 'salary' dan 'remote_ratio' adalah {corr2} \n")
print("# Karena nilai berkorelasi sekitar 0.019 dan mendekati 0, maka kedua atribut tidak berkorelasi. \n")
print("Grafik korelasi: \n")

# Visualisasi
from matplotlib import rcParams
import matplotlib as mpl
import matplotlib.lines as mln
rcParams['figure.figsize'] = 12,8

legend_elements = [mln.Line2D([0],[0], marker='o', color='w',label='2020', markerfacecolor='black', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2021', markerfacecolor='darkmagenta', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2022', markerfacecolor='darkorange', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2023', markerfacecolor='khaki', markersize=10)]

# Plotting
plt.style.use('seaborn-v0_8');
plt.scatter(df.salary, df.remote_ratio, c=df.work_year, cmap='inferno');
plt.title("Grafik Korelasi Antara 'salary' dengan 'remote_ratio' \n", loc='center', pad=10, fontsize=15, fontweight='bold', family='sans-serif');
plt.xlabel('Pendapatan Mata Uang Asal', fontsize=10, fontweight='bold');
plt.ylabel('Rasio Bekerja Secara Remote (Jarak Jauh)', fontsize=10, fontweight='bold');
plt.ticklabel_format(axis='x', style='sci', scilimits=(7,7), useMathText=True);
legend = plt.legend(handles=legend_elements, loc='center right', title='Tahun', frameon=True);
frame = legend.get_frame()
frame.set_facecolor('white')
frame.set_edgecolor('black')
plt.colorbar(label='Tahun', ticks=[2020, 2021, 2022, 2023]);
```

Gambar 5.4 *Syntax* untuk menampilkan nilai korelasi dan visualisasi korelasi antara 'salary' dengan 'remote_ratio'

5.3. Korelasi Antara 'salary_in_usd' dengan 'remote_ratio'

Nilai korelasi antara 'salary_in_usd' dan 'remote_ratio' adalah -0.09178909087399226. Karena nilai berkorelasi sekitar -0.091 dan mendekati 0, maka kedua atribut tidak berkorelasi.



Gambar 5.5 Visualisasi korelasi antara atribut 'salary_in_usd' dengan 'remote_ratio'

Pada grafik korelasi di atas, jika kita melakukan regresi linear terhadap grafik, maka didapatkan bahwa kemiringan garis regresi bernilai negatif tetapi sangat kecil dan mendekati 0, menunjukkan bahwa atribut 'salary_in_usd' dan 'remote_ratio' tidak berkorelasi.

Arti pemilihan warna: Warna yang digunakan merupakan warna dari gelap ke terang, yang menyatakan jumlah dari frekuensi munculnya tahun dari pendapatan tersebut. Semakin cerah maka jumlahnya semakin banyak (tahun 2023 muncul paling banyak pada dataset). Sebaliknya, semakin gelap maka jumlahnya semakin sedikit. (tahun 2020 muncul paling sedikit pada dataset)

```
# Nilai Korelasi
corr3 = df['salary_in_usd'].corr(df['remote_ratio'])

print(f"\n Nilai korelasi antara 'salary_in_usd' dan 'remote_ratio' adalah {corr3} \n")
print("# Karena nilai berkorelasi sekitar -0.091 dan mendekati 0, maka kedua atribut tidak berkorelasi. \n")
print("Grafik korelasi: \n")

# Visualisasi
from matplotlib import rcParams
import matplotlib as mpl
import matplotlib.lines as mln
rcParams['figure.figsize'] = 12,8

legend_elements = [mln.Line2D([0],[0], marker='o', color='w',label='2020', markerfacecolor='black', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2021', markerfacecolor='darkmagenta', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2022', markerfacecolor='darkorange', markersize=10),
                   mln.Line2D([0],[0], marker='o', color='w',label='2023', markerfacecolor='khaki', markersize=10)]

# Plotting
plt.style.use('seaborn-v0_8');
plt.scatter(df.salary_in_usd, df.remote_ratio, c=df.work_year, cmap='inferno');
plt.title("Grafik Korelasi Antara 'salary_in_usd' dengan 'remote_ratio' \n", loc='center', pad=10, fontsize=15, fontweight='bold', family='sans-serif');
plt.xlabel('Pendapatan (Dalam USD)', fontsize=10, fontweight='bold');
plt.ylabel('Rasio Bekerja Secara Remote (Jarak Jauh)', fontsize=10, fontweight='bold');
plt.ticklabel_format(axis='x', style='sci', scilimits=(5,5), useMathText=True);
legend = plt.legend(handles=legend_elements, loc='center right', title='Tahun', frameon=True);
frame = legend.get_frame()
frame.set_facecolor('white')
frame.set_edgecolor('black')
plt.colorbar(label='Tahun', ticks=[2020, 2021, 2022, 2023]);
```

Gambar 5.6 *Syntax* untuk menampilkan nilai korelasi dan visualisasi korelasi antara atribut 'salary_in_usd' dengan 'remote_ratio'

5.4. Tabel Korelasi Antar Atribut Kuantitatif

Korelasi antar atribut kuantitatif:

	salary	salary_in_usd	remote_ratio
salary	1.000000	0.049841	0.019285
salary_in_usd	0.049841	1.000000	-0.091789
remote_ratio	0.019285	-0.091789	1.000000

Tabel 5.1 Tabel korelasi antar atribut kuantitatif

```
print("Korelasi antar atribut kuantitatif: \n")

df[['salary', 'salary_in_usd', 'remote_ratio']].corr()
```

Gambar 5.7 *Syntax* untuk menampilkan tabel korelasi antar atribut kuantitatif

BAB VI
DATA CLEANSING

Mengecek Tingkat Kekotoran Data secara Umum

```
[ ]  print("Tingkat kekotoran data: \n")

     df.isnull()
```

Tingkat kekotoran data:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
8800	False	False	False	False	False	False	False	False	False	False	False
8801	False	False	False	False	False	False	False	False	False	False	False
8802	False	False	False	False	False	False	False	False	False	False	False
8803	False	False	False	False	False	False	False	False	False	False	False
8804	False	False	False	False	False	False	False	False	False	False	False

8805 rows × 11 columns

Gambar 6.1 Mengecek data kotor

Pada Gambar 6.1, terlihat nilai boolean. Pada tabel tersebut, "False" menandakan bahwa data tersebut merupakan data bersih (memiliki suatu value), dan "True" menandakan bahwa data tersebut merupakan data kotor (bernilai '-', NaN,dll).

Mengecek Tingkat Kekotoran Data per Kolom

```
[ ]  print("Data kotor per kolom: \n")

     clean = df.isnull().sum()
     print(clean.to_string())
```

```
Data kotor per kolom:

work_year            0
experience_level     0
employment_type      0
job_title            0
salary               0
salary_currency      0
salary_in_usd        0
employee_residence   0
remote_ratio         0
company_location     0
company_size         0
```

Mengecek Tingkat Kekotoran Data Secara Menyeluruh

```
[ ]  kotor = df.isnull().sum().sum()

     print(f"Total data kotor: {kotor} data.")
```

```
Total data kotor: 0 data.
```

(a) (b)

Gambar 6.2 Tingkat kekotoran per kolom

Pada Gambar 6.2 (a), ditunjukkan bahwa data kotor per kolom bernilai 0, menandakan bahwa dataset ini memiliki tingkat kekotoran data sebesar 0% seperti yang ditunjukkan pada Gambar 6.2 (b).

Mengecek Apakah Ada Data Duplikat dalam Kolom Bertipe Categorical-Nominal yang Bersifat Unique

```
[ ]  # Cek salah satu kolom dengan data kategorikal, pada kasus ini adalah kolom 'job_title'

     # Cek apakah data unik atau tidak dengan method .is_unique
     keunikan = df['job_title'].is_unique
     jumlahUnik = df['job_title'].nunique()

     print(f"Apakah data unik?           : {keunikan}")
     print(f"Jumlah data                 : {len(df)} data")
     print(f"Jumlah data tanpa duplikat  : {jumlahUnik} data")
```

```
Apakah data unik?           : False
Jumlah data                 : 8805 data
Jumlah data tanpa duplikat  : 124 data
```

Gambar 6.3 Mengecek keunikan tiap nilai data

Pada Gambar 6.3, atribut "job_title" ditunjukkan sebagai data yang tidak unik. Artinya, dalam atribut ini terdapat beberapa data dengan nilai yang sama. Hal ini bukan merupakan sebuah masalah sebab di dunia ini dua buah pekerjaan dapat memiliki nama yang sama. *Dataset* ini juga memiliki atribut lain dengan sifat serupa bahkan kategorinya telah ditentukan sehingga terdapat kemungkinan adanya data duplikat atau bernilai sama. Hanya pada kasus lain, seperti memeriksa data nama penerima bantuan sosial, penerima beasiswa, atau kependudukan, data duplikat harus diatasi.

BAB VII
KESIMPULAN, PEMBELAJARAN, DAN PEMBAGIAN TUGAS

7.1. Kesimpulan

Dataset yang digunakan dalam tugas ini adalah "salaries.csv" yang diambil dari "ai-jobs.net" melalui situs web Kaggle. Data ini berisi pendapatan berbagai atribut lain untuk pekerjaan di bidang *artificial intelligence*, *machine learning*, dan *data science*. *Dataset* berukuran 8805 baris dan 11 kolom ini memiliki format *comma separated values* dan berukuran 496 kb.

Setiap atribut memiliki karakteristiknya masing-masing. Selain jenis datanya (kategorikal atau kuantitatif), setiap atribut memiliki ciri-ciri seperti *range*, persentase kekotoran, keunikan, dan ciri-ciri lainnya. Dari setiap atribut juga bisa diperoleh berbagai informasi statistik. Perlu diperhatikan bahwa statistik menjadi bermakna ketika data yang disimpan dalam atribut memiliki tolak ukur yang konsisten, misalnya mata uang yang sama untuk setiap data. Statistik ini dapat memberikan informasi, seperti standar deviasi yang besar mengimplikasikan data yang tersebar dan mean yang menyatakan kecenderungan data secara umum (misalnya mean "remote_ratio" yang bernilai 38,6% menyatakan rata-rata pekerjaan dikerjakan secara hybrid).

7.2. Pembelajaran

Analisis data adalah sebuah ilmu untuk mengolah data yang mentah menjadi sebuah informasi atau *insight* yang dapat digunakan untuk mengambil berbagai keputusan yang lebih optimal. Terdapat berbagai peralatan atau *tools* yang dapat menunjang pekerjaan analisis data seperti *library* Pandas dan Matplotlib serta aplikasi perangkat lunak untuk mengedit program seperti Google Colab. Pemilihan jenis dan warna grafik sangat berpengaruh terhadap pemaknaan data yang disajikan. Sebagai seorang analis data, dibutuhkan ketelitian, mengolah informasi, dan empati serta kreativitas agar data yang mentah dapat diinterpretasikan dengan baik dan tepat oleh orang lain.

7.3. Pembagian Tugas

Nama	NIM	Tugas
Karol Yangqian Poetracahya	19623206	Laporan bagian Deskripsi Data dan *File*, Statistik, *Data Cleansing*, dan Kesimpulan serta Pembelajaran
Nayaka Ghana Subrata	19623031	Program dalam Google Colab

Julian Benedict	16523178	1. Laporan bagian Karakteristik Data 2. Mengedit video
Dimas Anggiat	16523052	Laporan bagian Visualisasi dan Korelasi