Nama : Ahmad mufli Ramadhan
Nim : 13020220227
Kelas : C1

DATA MINING

1. Berikan masing masing contoh penerapan data processing: data cleaning (semua metode yang ada pada slide data cleaning), data integration, data reduction, data transformation

1.1 Data Cleaning

a. Metode penghapusan Missing values

Contoh missing values tipe(MAR)

Input :

```python
import pandas as pd

data = {
    'Cabang': ['A', 'B', 'C', 'D'],
    'Penjualan_Jan': [35000, 42000, 38000, None],  # Data cabang D hilang
    'Penjualan_Feb': [38000, 39000, None, 40000]   # Data cabang C hilang
}

df = pd.DataFrame(data)
print("Data Awal:")
print(df)
```

Output :

```
Data Awal:
  Cabang  Penjualan_Jan  Penjualan_Feb
0      A        35000.0        38000.0
1      B        42000.0        39000.0
2      C        38000.0            NaN
3      D            NaN        40000.0
```

Dapat terlihat di output bahwa penjualan cabang D pada bulan januari menghilang dan penjualan cabang d pada bulan februari juga menghilang.

Untuk mengatasi missing values dapat menggunakan cara-cara dibawah ini :

- Imputasi : mengganti nilai yang hilang dengan nilai yang disimpulkan dari data yang tersedia
- Interpolasi
- Penghapusan poliomial

b. Noise data

Contoh penerapan noise data :

Input :

```python
import pandas as pd

# Membuat DataFrame dengan data suhu harian
data = {
    'Tanggal': pd.date_range(start='2024-01-01', end='2024-01-10'),
    'Suhu_C': [22.5, 23.0, 22.7, 24.5, 22.8, 23.2, 22.9, 45.0, 23.1, 22.6]
}

df = pd.DataFrame(data)
print("Data Awal:")
print(df)
```

Output :

```
Data Awal:
      Tanggal   Suhu_C
0  2024-01-01     22.5
1  2024-01-02     23.0
2  2024-01-03     22.7
3  2024-01-04     24.5
4  2024-01-05     22.8
5  2024-01-06     23.2
6  2024-01-07     22.9
7  2024-01-08     45.0
8  2024-01-09     23.1
9  2024-01-10     22.6
```

Pada baris ke 7 yaitu pada tanggal 2024-01-08 dapat terlihat bahwa suhunya melebihi batas normal. Cara-cara untuk mengatasi noise data dapat menggunakan cara di bawah ini :

- Deteksi noise : pada kasus di atas kita menemukan bahwa pada baris ke 7 suhu ada pada angka yang tidak wajar
- Filtering : jika noise sudah terdeteksi(baris ke-7), kita dapat memfilter data yang terpengaruh atau bisa menghapusnya dari dataset
- Preprocessing : kita juga bisa menggunakan teknik smoothing atau Averaging untuk mengurangi efek dari noise
- Validasi dan verifikasi : Sangat penting untuk memvalidasi dan memverifikasi data, terutama jika data tersebut memiliki dampak signifikan, seperti dalam pengambilan keputusan atau analisis kritis.

c. Inconsistent data

Input :

tugas.py

```python
import pandas as pd

# Membuat DataFrame dengan data transaksi
data = {
    'Tanggal': ['2023-01-15', '2023/02/20', '2023-03-25', '2023-04-30', '2023-05-05'],
    'Jumlah_Penjualan': [1500, 1800, 2000, 2200, 2100]
}

df = pd.DataFrame(data)
print("Data Awal:")
print(df)
```

Output :

```
Data Awal:
      Tanggal  Jumlah_Penjualan
0  2023-01-15              1500
1  2023/02/20              1800
2  2023-03-25              2000
3  2023-04-30              2200
4  2023-05-05              2100
```

Pada baris ke 2 yaitu pada data index ke 2(no 1) Tanggal mengalami perubahan format. Yang awalnya YYY-MMM-DDD menjadi YYY/MMM/DDD

Untuk menangani data yang tidak konsisten di atas kita dapat menggunakan langkah-langkah berikut :

- Standardisasi Format: Mengubah semua format tanggal menjadi format yang konsisten, misalnya, mengubah semua strip ("-") menjadi garis miring ("/").
- Parsing Tanggal: Menggunakan fungsi penguraian tanggal untuk mengidentifikasi dan mengonversi format tanggal yang berbeda ke format yang konsisten.

1.2 Data integration

Input :

```python
# Set data pelanggan
customer_data = [
    {"ID Pelanggan": 1, "Nama Pelanggan": "mufli", "Umur": 30, "Kota": "Manado"},
    {"ID Pelanggan": 2, "Nama Pelanggan": "baso", "Umur": 25, "Kota": "bone"},
    {"ID Pelanggan": 3, "Nama Pelanggan": "umul", "Umur": 35, "Kota": "Surabaya"},
    {"ID Pelanggan": 4, "Nama Pelanggan": "ikhsan", "Umur": 28, "Kota": "surabaya"},
    {"ID Pelanggan": 5, "Nama Pelanggan": "alif", "Umur": 40, "Kota": "makassar"}
]

# Set data transaksi
transaction_data = [
    {"ID Pelanggan": 1, "Tanggal Transaksi": "2024-02-20", "Total Transaksi": 1500000},
    {"ID Pelanggan": 2, "Tanggal Transaksi": "2024-02-21", "Total Transaksi": 2000000},
    {"ID Pelanggan": 3, "Tanggal Transaksi": "2024-02-20", "Total Transaksi": 1800000},
    {"ID Pelanggan": 4, "Tanggal Transaksi": "2024-02-21", "Total Transaksi": 1200000},
    {"ID Pelanggan": 5, "Tanggal Transaksi": "2024-02-20", "Total Transaksi": 2500000}
]

# Melakukan penggabungan data berdasarkan ID Pelanggan
merged_data = []
for customer in customer_data:
    for transaction in transaction_data:
        if customer["ID Pelanggan"] == transaction["ID Pelanggan"]:
            merged_data.append({**customer, **transaction})

# Menampilkan hasil penggabungan data
print("Hasil penggabungan data:")
for data in merged_data:
    print(data)
```

Output :

```
Hasil penggabungan data:
{'ID Pelanggan': 1, 'Nama Pelanggan': 'mufli', 'Umur': 30, 'Kota': 'Manado', 'Tanggal Transaksi': '2024-02-20', 'Total Transaksi': 1500000}
{'ID Pelanggan': 2, 'Nama Pelanggan': 'baso', 'Umur': 25, 'Kota': 'bone', 'Tanggal Transaksi': '2024-02-21', 'Total Transaksi': 2000000}
{'ID Pelanggan': 3, 'Nama Pelanggan': 'umul', 'Umur': 35, 'Kota': 'Surabaya', 'Tanggal Transaksi': '2024-02-20', 'Total Transaksi': 1800000}
{'ID Pelanggan': 4, 'Nama Pelanggan': 'ikhsan', 'Umur': 28, 'Kota': 'surabaya', 'Tanggal Transaksi': '2024-02-21', 'Total Transaksi': 1200000}
{'ID Pelanggan': 5, 'Nama Pelanggan': 'alif', 'Umur': 40, 'Kota': 'makassar', 'Tanggal Transaksi': '2024-02-20', 'Total Transaksi': 2500000}
```

Kita menggabungkan 2 classs yang berbeda yaitu transaction dan customer. Dalam setiap for loop akan diperiksa apakah id pelanggan cocok maka data akan disatukan menggunakan merged dictionary.

1.3 Data reduction

```python
import random
transaction_data = [
    {"ID Pelanggan": 1, "Tanggal Transaksi": "2024-01-01", "Total Transaksi": 1500000},
    {"ID Pelanggan": 2, "Tanggal Transaksi": "2024-01-02", "Total Transaksi": 2000000},
    {"ID Pelanggan": 3, "Tanggal Transaksi": "2024-01-03", "Total Transaksi": 1800000},
    {"ID Pelanggan": 4, "Tanggal Transaksi": "2024-01-04", "Total Transaksi": 1200000},
    {"ID Pelanggan": 5, "Tanggal Transaksi": "2024-01-05", "Total Transaksi": 2500000},
]
# Menentukan proporsi data yang akan disampling
proporsi_sampling = 0.3
sampled_data = random.sample(transaction_data, int(proporsi_sampling * len(transaction_data)))

# Menampilkan hasil sampling
print("Data setelah direduksi:")
for data in sampled_data:
    print(data)
```

Output:

```
Data setelah direduksi:
{'ID Pelanggan': 2, 'Tanggal Transaksi': '2024-01-02', 'Total Transaksi': 2000000}
```

Saya menggunakan metode random sampling untuk menggunakan sebagian data dari data transaksi. Saya mengambil 30% dari data asli untuk melakukan data reduction

1.4 Data transformation

Input:

```python
import pandas as pd

# Membuat dataset
nama_data = {
    'Nama Lengkap': ['Ahmad Mufli', 'Baso ummul ikhsan', 'Alif maullana', 'Abizar Maany', 'Nurul fajeriani']
}

df_nama = pd.DataFrame(nama_data)

# Mengekstrak inisial nama depan dan nama belakang
df_nama['Inisial Depan'] = df_nama['Nama Lengkap'].str.split().str[0].str[0]
df_nama['Inisial Belakang'] = df_nama['Nama Lengkap'].str.split().str[-1].str[0]

#print hasil
print("Hasil ekstraksi inisial nama:")
print(df_nama)
```

Output:

```
  import pandas as pd
Hasil ekstraksi inisial nama:
        Nama Lengkap Inisial Depan Inisial Belakang
0         Ahmad Mufli             A                 M
1  Baso ummul ikhsan              B                 i
2      Alif maullana              A                 m
3       Abizar Maany              A                 M
4    Nurul fajeriani              N                 f
```

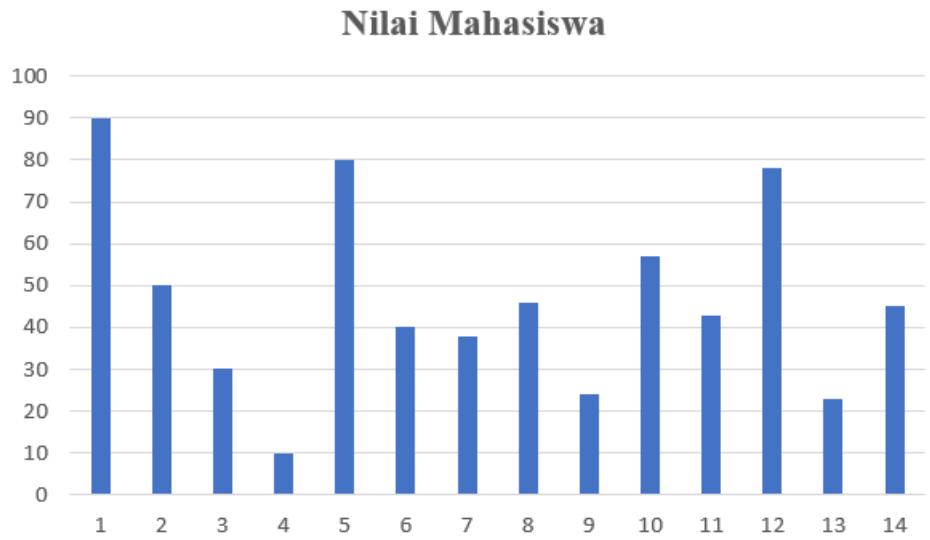
Pada kasus ini saya mentranformasi nama lengkap untuk mengetahui inisial nama depan dan belakang pengguna. Berkat transformasi data kita mendapatkan sebuah dataset yang baru berisi inisial depan dan inisial belakang.

2. Berikan 3 contoh tipe visualisasi dari artikel yang berbeda-beda

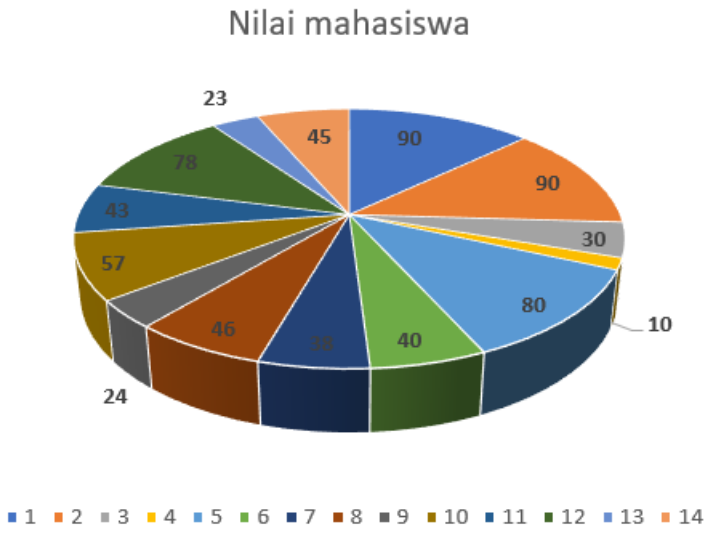
Untuk memudahkan membuat sebuah visualisasi data. Saya membuat data set untuk melihat nilai yang diperoleh mahasiswa.

NO	NAMA	NILAI			Grades
1	Ahmad	90			A = >85
2	Mufli	50			B = >75
3	Baso	30			C = >60
4	Iccank	10			D = <50
5	Alif	80			E = 0
6	Maulana	40			
7	Fateh	38			
8	Rafathar	46			
9	cipung	24			
10	Raffi	57			
11	Gigi	43			
12	Anang	78			
13	Hermansyah	23			
14	Atta	45			

a. Visualisasi menggunakan column chart



b. Visualisasi menggunakan pie chart



c. Visualisasi menggunakan doughnut