NIM	: 2211522011	Tanggal	: Rabu, 06 Feb 2024
Nama	: Ghifari Rizki Ramadhan	Asisten	:1. Muhammad Irsyadul Fikri
			2. Athifa Rifda Andra
			3. Annisa Gita Subhi
			4. Syakina Triyana
			5. Husna Afiqah Yossyafra
			6. Vania Zerlina Utami
			7. Ghina Fitri Hidayah
			8. Sukma Anggarmadi
			9. Rafiqatul Ulya

Mata Kuliah : Praktikum Data Mining
Modul : 02
Kelas : A

Resume dan Tugas "Operasi File dan Eskpresi "\

Ada 2 tipe file yang dikelompokkan pada bahasa python :

1. File teks : file ini berisikan teks; contoh : file txt, csv, tsv, json, md, dll
2. File binary : file ini berisikan non-teks yang hanya diproses oleh program tertentu contoh: jpg, jpeg, exe, mkv, m4a, dll

Pada python terdapat beberapa operasi hitung yaitu:

Operator	Keterangan
+	Menambahkan dua obyek
-	Mengurangi obyek dengan obyek yang lain
*	Perkalian
**	Pangkat
/	Pembagian
//	Pembagian bulat ke bawah

%	Sisa hasil bagi (modulus)
<<	(geser kiri) Menggeser bit ke sebelah kiri sesuai dengan jumlah bit yang ditentukan. `2 << 2` menghasilkan `8`. `2` direpresentasikan `10` dalam bit (binary digit). Menggeser 2 bit kekiri akan menghasilkan `1000` yang merupakan representasi dari desimal `8`.
>>	(geser kanan) Menggeser bit ke sebelah kanan sesuai dengan jumlah bit yang ditentukan. `11 > 1` menghasilkan `5`. `11` direpresentasikan oleh bit dengan `1011` kemudian digeser kekanan 1 bit menghasilkan `101` yang merupakan desimal angka `5`.
&	(bit-wise AND) Operasi bit-wise AND dari angka (bit-wise adalah operasi angka berbasis bit yakni dengan `0` dan `1`). `5 & 3` menghasilkan `1`.
|	(bit-wise OR) Operasi bit-wise OR dari angka. `5 | 3` menghasilkan `7`.
^	(bit-wise XOR) Operasi bit-wise XOR (ekslusif OR). `5 ^ 3` menghasilkan `6`.
~	(bit-wise invert) Operasi membalikkan angka bitwise dari `x`, menghasilkan `-x - 1`. `~5` akan menghasilkan `-6`. lihat two's complement.
<	(kurang dari) Mengembalikan apakah x kurang dari y. Semua operator pembanding mengembalikan `True` atau `False`. `5 < 3` mengembalikan `False`, `3 < 5` mengembalikan `True` dan `2 < 5 < 7` mengembalikan `True`.
>	(lebih dari) Mengembalikan apakah x lebih dari y. `5 > 3` mengembalikan `True`.
<=	(kurang dari atau sama dengan) Mengembalikan apakah x kurang dari atau sama dengan y. `5 <= 5` mengembalikan `True`.
>=	(lebih dari atau sama dengan) Mengembalikan apakah x lebih dari atau sama dengan y. `5 >= 5` mengembalikan `True`.
==	(sama dengan) Membandingkan apakah kedua obyek sama. `2 == 2` mengembalikan `True`, `'nama' == 'Nama'` mengembalikan `False`, `'nama' == 'nama'` mengembalikan `True`.
!=	(tidak sama dengan) Membandingkan apakah kedua obyek berbeda. `2 != 3` mengembalikan `True`.
not	(boolean NOT) Jika x bernilai `True` akan mengembalikan `False`. Jika x bernilai `False` akan mengembalikan `True`. `x = True; not x` mengembalikan `False`.
and	(boolean AND) `x and y` mengembalikan `False` jika x bernilai `False`, selain itu akan mengembalikan nilai `y`. `x = False; y = True; x and y` akan mengembalikan `False` karena `x` bernilai `False`. Pada kasus ini Python tidak akan mengevaluasi `y` karena nilai `x`. Hal ini disebut *short-circuit* evaluasi.
or	(boolean OR) Jika `x` bernilai True, `x or y` akan mengembalikan `True`, selain itu akan mengembalikan nilai `y`. `x = True; y = False; x or y` mengembalikan `True`. *short-circuit* evaluasi berlaku juga disini.

Python juga bisa melakukan urutan evaluasi. Jika ada rantaian ekspresi seperti 2 + 3 * 4, apakah penambahan dilakukan terlebih dahulu atau perkalian? Saat pelajaran matematika kita diajari bahwa perkalian harus dikerjakan terlebih dahulu. Hal ini menandakan perkalian mempunyai urutan lebih tinggi daripada penambahan. Berikut tabel urutan evaluasi ekspresi dalam Python, dari terrendah sampai tertinggi.

Operator	Keterangan
lamda	Ekspresi `lambda`
or	Boolean OR
and	Boolean AND
not x	Boolean NOT
in, not in	Tes Keanggotaan
is, is not	Tes Identitas
<, <=, >, >=, !=, ==	Perbandingan
|	Bitwise OR
^	Bitwise XOR
&	Bitwise AND
<<, >>	Shift
+, -	Penambahan dan Pengurangan
*, /, //, %	Perkalian, Pembagian, Pembagian ke bawah, mod
+x, -x	Positif, Negatif
~x	Bitwise NOT / inverse
**	Pangkat
x.attribute	Referensi atribut
x[index]	Akses item
x[index1:index2]	Slicing
f(argument ...)	Pemanggilan fungsi
(ekspresi, ...)	literal tuple
[ekspresi, ...]	literal list
{key:value, ...}	literal dictionary

Tugas Rumah

1. **Bank ABC**

```
print("Bank ABC")

setor = float(input("masukkan jumlah uang yang akan di tabung :"))
tahun = int(input("kalkulasikan untuk berapa tahun :"))
opsi = input("apakah ingin menambah tabungan? (y/t) :")

if(opsi == 't'):
    saldo = setor*12
    bunga = 0.05
    saldoBaru = saldo + bunga*saldo
    for i in range(tahun-1):
        saldoBaru = saldoBaru+saldo+(saldoBaru + saldo)*bunga

elif(opsi == 'y'):
    tambahan = float(input("masukkan jumlah uang tambahan"))
    setor = setor + tambahan
    saldo = setor*12
    bunga = 0.05
    saldoBaru = saldo + bunga*saldo
    for i in range(tahun-1):
        saldoBaru = saldoBaru+saldo+(saldoBaru + saldo)*bunga

print("saldo setelah 3 tahun =", saldoBaru)
```

Program dibuat untuk mengkalkulasikan tabungan setelah 3 tahun menabung. Pada program ini terdapat percabangan if dikarenakan terdapat 2 kondisi yaitu jika budi menabung 1juta perbulan atau budi menambah nominal tabungan tiap bulannya ‘y’ untuk tambah dan ‘t’ untuk tidak. Selanjutnya ada perulangan for untuk mengkalkulasikan berapa tahun yang akan dijumlahkan beserta bunga. Dikarenakan yang diminta 3 tahun maka pada perulangan for dibuat range sebanyak tahun-1 karena dapat dilihat pada program untuk pencarian saldo dan bunga pertama sudah dilakukan terlebih dahulu yang berarti perulangan utk tahun pertama sudah dilakukan sebelumnya.

```
Bank ABC
masukkan jumlah uang yang akan di tabung : 1000000
kalkulasikan untuk berapa tahun : 3
apakah ingin menambah tabungan? (y/t) : t
saldo setelah 3 tahun = 39721500.0
```

```
Bank ABC
masukkan jumlah uang yang akan di tabung : 1000000
kalkulasikan untuk berapa tahun : 3
apakah ingin menambah tabungan? (y/t) : y
masukkan jumlah uang tambahan 1000000
saldo setelah 3 tahun = 79443000.0
```

2. **Import dataset dengan pandas dan drop columns**

```
[1]  import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
```

```
[4]  dt = pd.read_csv("/content/data_science_job.csv")
```

```
[7]  dt.head()
```

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_
0	8949	city_103	0.920	Male	Has relevent experience	no
1	29725	city_40	0.776	Male	No relevent experience	no
2	11561	city_21	0.624	NaN	No relevent experience	Full
3	33241	city_115	0.789	NaN	No relevent experience	
4	666	city_162	0.767	Male	Has relevent experience	no

```
[7] dt.head()
```

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_
0	8949	city_103	0.920	Male	Has relevent experience	no
1	29725	city_40	0.776	Male	No relevent experience	no
2	11561	city_21	0.624	NaN	No relevent experience	Full
3	33241	city_115	0.789	NaN	No relevent experience	
4	666	city_162	0.767	Male	Has relevent experience	no

```
drop.dtypes
```

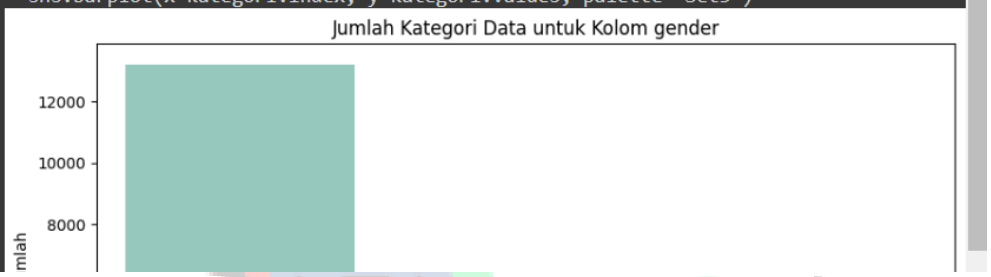
```
gender                  object
relevent_experience     object
enrolled_university     object
education_level         object
major_discipline        object
experience             float64
training_hours         float64
target                 float64
dtype: object
```

```
drop.isna().sum()
```

```
gender                  4508
relevent_experience        0
enrolled_university      386
education_level          460
major_discipline        2813
experience                65
training_hours           766
target                     0
dtype: int64
```

```
[10] kolom_plot = {}

     # Hitung jumlah kategori data untuk setiap kolom
     for kolom in drop.columns:
         kolom_plot[kolom] = drop[kolom].value_counts()

     # Buat plot untuk setiap kolom
     for kolom, kategori in kolom_plot.items():
         plt.figure(figsize=(10, 5))
         sns.barplot(x=kategori.index, y=kategori.values, palette="Set3")
         plt.title(f"Jumlah Kategori Data untuk Kolom {kolom}")
         plt.xlabel("Kategori")
         plt.ylabel("Jumlah")
         plt.xticks(rotation=45)
         plt.show()
```

```
<ipython-input-10-be6330937eb1>:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(x=kategori.index, y=kategori.values, palette="Set3")
```




```
drop['gender'] = drop['gender'].fillna("Male")
drop['enrolled_university'] = drop['enrolled_university'].fillna("no_enrollment")
drop['education_level'] = drop['education_level'].fillna("Graduate")
drop['major_discipline'] = drop['major_discipline'].fillna("STEM")
drop['experience'] = drop['experience'].fillna("experience")
drop['training_hours'] = drop['training_hours'].fillna(drop['training_hours'].mean().round(2))
drop.head(50)
```

6	Male	Has relevent experience	no_enrollment	High School	STEM	5.0	24.00	0.0
7	Male	Has relevent experience	no_enrollment	Graduate	STEM	13.0	18.00	1.0
8	Male	Has relevent experience	no_enrollment	Graduate	STEM	7.0	46.00	1.0
9	Male	Has relevent experience	no_enrollment	Graduate	STEM	17.0	123.00	0.0
10	Male	No relevent experience	Full time course	High School	STEM	2.0	32.00	1.0
11	Male	Has relevent experience	no_enrollment	Graduate	STEM	5.0	108.00	0.0
12	Male	Has relevent experience	no_enrollment	Graduate	STEM	20.0	23.00	0.0
13	Male	No relevent experience	no_enrollment	Graduate	STEM	2.0	24.00	0.0
14	Male	No relevent experience	Full time course	High School	STEM	5.0	26.00	0.0
15	Male	Has relevent experience	no_enrollment	Graduate	STEM	16.0	18.00	0.0
16	Male	Has relevent experience	no_enrollment	Graduate	STEM	1.0	106.00	0.0
17	Male	Has relevent experience	no_enrollment	Graduate	STEM	2.0	7.00	0.0

DAFTAR PUSTAKA

Operator dan Ekspresi — Workshop Python 101, https://sakti.github.io/python101/operator_dan_ekspresi.html. Accessed 10 March 2024.