

Collection Manipulation dengan Python

Mengakses List dan Tuple – Part 1

In []:

```
bulan_pembelian = ('Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni', 'Juli', 'Agustus', 'September', 'Oktober', 'November', 'Desember')
print(bulan_pembelian[0])
print(bulan_pembelian[5])
print(bulan_pembelian[-1])
print(bulan_pembelian[-2])
```

```
Januari
Juni
Desember
November
```

Mengakses List dan Tuple – Part 2

In []:

```
bulan_pembelian = ('Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni', 'Juli', 'Agustus', 'September', 'Oktober', 'November', 'Desember')
pertengahan_tahun = bulan_pembelian[4:8]
print(pertengahan_tahun)
awal_tahun = bulan_pembelian[:5]
print(awal_tahun)
akhir_tahun = bulan_pembelian[8:]
print(akhir_tahun)
print(bulan_pembelian[-4:-1])
```

```
('Mei', 'Juni', 'Juli', 'Agustus')
('Januari', 'Februari', 'Maret', 'April', 'Mei')
('September', 'Oktober', 'November', 'Desember')
('September', 'Oktober', 'November')
```

Penggabungan Dua atau Lebih List atau Tuple

In []:

```
list_makanan = ['Gado-gado', 'Ayam Goreng', 'Rendang']
list_minuman = ['Es Teh', 'Es Jeruk', 'Es Campur']
list_menu = list_makanan + list_minuman
print(list_menu)
```

```
['Gado-gado', 'Ayam Goreng', 'Rendang', 'Es Teh', 'Es Jeruk', 'Es Campur']
```

List Manipulation – Part 1

In []:

```
# Fitur .append()
print(">>> Fitur .append()")
list_makanan = ['Gado-gado', 'Ayam Goreng', 'Rendang']
list_makanan.append('Ketoprak')
print(list_makanan)
# Fitur .clear()
print(">>> Fitur .clear()")
list_makanan = ['Gado-gado', 'Ayam Goreng', 'Rendang']
list_makanan.clear()
print(list_makanan)
# Fitur .copy()
print(">>> Fitur .copy()")
list_makanan1 = ['Gado-gado', 'Ayam Goreng', 'Rendang']
```

```

list_makanan2 = list_makanan1.copy()
list_makanan3 = list_makanan1
list_makanan2.append('Opor')
list_makanan3.append('Ketoprak')
print(list_makanan1)
print(list_makanan2)
# Fitur .count()
print(">>> Fitur .count()")
list_score = ['Budi', 'Sud', 'Budi', 'Budi', 'Budi', 'Sud', 'Sud']
score_budi = list_score.count('Budi')
score_sud = list_score.count('Sud')
print(score_budi) # akan menampilkan output 4
print(score_sud) # akan menampilkan output 3
# Fitur .extend()
print(">>> Fitur .extend()")
list_menu = ['Gado-gado', 'Ayam Goreng', 'Rendang']
list_minuman = ['Es Teh', 'Es Jeruk', 'Es Campur']
list_menu.extend(list_minuman)
print(list_menu)

```

```

>>> Fitur .append()
['Gado-gado', 'Ayam Goreng', 'Rendang', 'Ketoprak']
>>> Fitur .clear()
[]
>>> Fitur .copy()
['Gado-gado', 'Ayam Goreng', 'Rendang', 'Ketoprak']
['Gado-gado', 'Ayam Goreng', 'Rendang', 'Opor']
>>> Fitur .count()
4
3
>>> Fitur .extend()
['Gado-gado', 'Ayam Goreng', 'Rendang', 'Es Teh', 'Es Jeruk', 'Es Campur']

```

List Manipulation – Part 2

In []:

```

# Fitur .index()
print(">>> Fitur .index()")
list_score = ['Budi', 'Sud', 'Budi', 'Budi', 'Budi', 'Sud', 'Sud']
score_pertama_sud = list_score.index('Sud') + 1
print(score_pertama_sud) # akan menampilkan output 2
# Fitur .insert()
print(">>> Fitur .insert()")
list_score = ['Budi', 'Sud', 'Budi', 'Budi', 'Sud']
list_score.insert(3, 'Sud')
print(list_score)
# Fitur .pop()
print(">>> Fitur .pop()")
list_menu = ['Gado-gado', 'Ayam Goreng', 'Rendang']
list_menu.pop(1)
print(list_menu)
# Fitur .remove()
print(">>> Fitur .remove()")
list_menu = ['Gado-gado', 'Ayam Goreng', 'Rendang', 'Ketoprak']
list_menu.remove('Rendang')
print(list_menu)
# Fitur .reverse()
print(">>> Fitur .reverse()")
list_menu = ['Gado-gado', 'Ayam Goreng', 'Rendang', 'Ketoprak']
list_menu.reverse()
print(list_menu)
# Fitur .sort()
print(">>> Fitur .sort()")
list_menu = ['Gado-gado', 'Ayam Goreng', 'Rendang', 'Ketoprak']
list_menu.sort() # Default: Ascending
print(list_menu)
list_menu.sort(reverse=True) # Descending
print(list_menu)

```

```

>>> Fitur .index()
2

```

```

>>> Fitur .insert()
['Budi', 'Sud', 'Budi', 'Sud', 'Budi', 'Sud']
>>> Fitur .pop()
['Gado-gado', 'Rendang']
>>> Fitur .remove()
['Gado-gado', 'Ayam Goreng', 'Ketoprak']
>>> Fitur .reverse()
['Ketoprak', 'Rendang', 'Ayam Goreng', 'Gado-gado']
>>> Fitur .sort()
['Ayam Goreng', 'Gado-gado', 'Ketoprak', 'Rendang']
['Rendang', 'Ketoprak', 'Gado-gado', 'Ayam Goreng']

```

Tuple Manipulation

In []:

```

# Fitur .count()
print(">>> Fitur .count()")
tuple_score = ('Budi', 'Sud', 'Budi', 'Budi', 'Budi', 'Sud', 'Sud')
score_budi = tuple_score.count('Budi')
score_sud = tuple_score.count('Sud')
print(score_budi) # akan menampilkan output 4
print(score_sud) # akan menampilkan output 3
# Fitur .index()
print(">>> Fitur .index()")
tuple_score = ('Budi', 'Sud', 'Budi', 'Budi', 'Budi', 'Sud', 'Sud')
score_pertama_sud = tuple_score.index('Sud')+1
print(score_pertama_sud) # akan menampilkan output 2

```

```

>>> Fitur .count()
4
3
>>> Fitur .index()
2

```

Set Manipulation – Part 1

In []:

```

# Fitur .add()
print(">>> Fitur .add()")
set_buah = {'Jeruk', 'Apel', 'Anggur'}
set_buah.add('Melon')
print(set_buah)
# Fitur .clear()
print(">>> Fitur .clear()")
set_buah = {'Jeruk', 'Apel', 'Anggur'}
set_buah.clear()
print(set_buah)
# Fitur .copy()
print(">>> Fitur .copy()")
set_buah1 = {'Jeruk', 'Apel', 'Anggur'}
set_buah2 = set_buah1
set_buah3 = set_buah1.copy()
set_buah2.add('Melon')
set_buah3.add('Kiwi')
print(set_buah1)
print(set_buah2)
# Fitur .update()
print(">>> Fitur .update()")
parcel1 = {'Anggur', 'Apel', 'Jeruk'}
parcel2 = {'Apel', 'Kiwi', 'Melon'}
parcel1.update(parcel2)
print(parcel1)
# Fitur .pop()
print(">>> Fitur .pop()")
parcel = {'Anggur', 'Apel', 'Jeruk'}
buah = parcel.pop()
print(buah)
print(parcel)

```

```
# Fitur .remove()
print(">>> Fitur .remove()")
parcel = {'Anggur', 'Apel', 'Jeruk'}
parcel.remove('Apel')
print(parcel)

>>> Fitur .add()
{'Melon', 'Jeruk', 'Anggur', 'Apel'}
>>> Fitur .clear()
set()
>>> Fitur .copy()
{'Melon', 'Jeruk', 'Anggur', 'Apel'}
{'Melon', 'Jeruk', 'Anggur', 'Apel'}
>>> Fitur .update()
{'Melon', 'Kiwi', 'Jeruk', 'Anggur', 'Apel'}
>>> Fitur .pop()
Jeruk
{'Anggur', 'Apel'}
>>> Fitur .remove()
{'Jeruk', 'Anggur'}
```

Set Manipulation – Part 2

In []:

```
# Fitur .union()
print(">>> Fitur .union()")
parcel1 = {'Anggur', 'Apel', 'Jeruk'}
parcel2 = {'Apel', 'Kiwi', 'Melon'}
parcel3 = parcel1.union(parcel2)
print(parcel1)
print(parcel3)
# Fitur .isdisjoint()
print(">>> Fitur .isdisjoint()")
parcel1 = {'Anggur', 'Apel', 'Jeruk'}
parcel2 = {'Kiwi', 'Melon', 'Pisang'}
parcel3 = {'Apel', 'Srikaya', 'Semangka'}
parcel1_parcel2_disjoint = parcel1.isdisjoint(parcel2)
print(parcel1_parcel2_disjoint)
parcel1_parcel3_disjoint = parcel1.isdisjoint(parcel3)
print(parcel1_parcel3_disjoint)
# Fitur .issubset()
print(">>> Fitur .issubset()")
parcel_A = {'Anggur', 'Apel'}
parcel_B = {'Durian', 'Semangka', 'Apel'}
parcel_C = {'Anggur', 'Kiwi', 'Apel', 'Jeruk', 'Melon'}
parcel_A_dalam_C = parcel_A.issubset(parcel_C)
parcel_B_dalam_C = parcel_B.issubset(parcel_C)
print(parcel_A_dalam_C)
print(parcel_B_dalam_C)
# Fitur .issuperset()
print(">>> Fitur .issuperset()")
parcel_C_mengandung_A = parcel_C.issuperset(parcel_A)
parcel_C_mengandung_B = parcel_C.issuperset(parcel_B)
print(parcel_C_mengandung_A)
print(parcel_C_mengandung_B)
# Fitur .intersection()
print(">>> Fitur .intersection()")
parcel_A = {'Anggur', 'Kiwi', 'Apel', 'Jeruk', 'Melon'}
parcel_B = {'Apel', 'Jeruk', 'Semangka', 'Durian', 'Tomat'}
parcel_C = parcel_A.intersection(parcel_B)
print(parcel_C)
# Fitur .difference()
print(">>> Fitur .difference()")
parcel_C = parcel_A.difference(parcel_B)
print(parcel_C)
# Fitur .symmetric_difference()
print(">>> Fitur .symmetric_difference()")
parcel_A = {'Anggur', 'Apel', 'Jeruk', 'Melon'}
parcel_B = {'Apel', 'Jeruk', 'Semangka', 'Leci'}
parcel_C = parcel_A.symmetric_difference(parcel_B)
```

```
print(parcel_C)

>>> Fitur .union()
{'Jeruk', 'Anggur', 'Apel'}
{'Melon', 'Kiwi', 'Jeruk', 'Anggur', 'Apel'}
>>> Fitur .isdisjoint()
True
False
>>> Fitur .issubset()
True
False
>>> Fitur .issuperset()
True
False
>>> Fitur .intersection()
{'Jeruk', 'Apel'}
>>> Fitur .difference()
{'Melon', 'Anggur', 'Kiwi'}
>>> Fitur .symmetric_difference()
{'Leci', 'Semangka', 'Anggur', 'Melon'}
```

Dictionary Manipulation

In []:

```
# Fitur .clear()
print(">>> Fitur .clear()")
info_karyawan = {'nama' : 'Aksara',
                  'nik' : '1211011',
                  'pekerjaan' : 'Data Analyst'}
info_karyawan.clear()
print(info_karyawan)
# Fitur .copy()
print(">>> Fitur .copy()")
info_karyawan1 = {'nama' : 'Aksara',
                  'nik' : '1211011',
                  'pekerjaan' : 'Data Analyst'}
info_karyawan2 = info_karyawan1.copy()
info_karyawan2['nama'] = 'Senja'
info_karyawan2['nik'] = '1211056'
print(info_karyawan1)
print(info_karyawan2)
# Fitur .keys()
print(">>> Fitur .keys()")
info_karyawan = {'nama' : 'Aksara',
                  'nik' : '1211011',
                  'pekerjaan' : 'Data Analyst'}
kunci_akses = list(info_karyawan.keys())
print(kunci_akses)
# Fitur .values()
print(">>> Fitur .values()")
value_dict = list(info_karyawan.values())
print(value_dict)
# Fitur .update()
print(">>> Fitur .update()")
info_karyawan.update({'skillset': ['Python', 'R']})
print(info_karyawan)
```

```
>>> Fitur .clear()
{}
>>> Fitur .copy()
{'nama': 'Aksara', 'nik': '1211011', 'pekerjaan': 'Data Analyst'}
{'nama': 'Senja', 'nik': '1211056', 'pekerjaan': 'Data Analyst'}
>>> Fitur .keys()
['nama', 'nik', 'pekerjaan']
>>> Fitur .values()
['Aksara', '1211011', 'Data Analyst']
>>> Fitur .update()
{'nama': 'Aksara', 'nik': '1211011', 'pekerjaan': 'Data Analyst', 'skillset': ['Python', 'R']}
```

Useful Tips and Tricks

In []:

```
# Tuple
print(">>> Tuple")
tuple_menu = ('Gado-gado', 'Ayam Goreng', 'Rendang', 'Ketoprak')
jumlah_menu = len(tuple_menu)
print(jumlah_menu)
# List
print(">>> List")
list_menu = ['Gado-gado', 'Ayam Goreng', 'Rendang', 'Ketoprak']
jumlah_menu = len(list_menu)
print(jumlah_menu)
# Konversi tipe data
print(">>> Konversi tipe data")
list_buah = ['Apel', 'Apel', 'Jeruk', 'Markisa', 'Jeruk', 'Markisa', 'Apel']
set_buah = set(list_buah)
print(set_buah)
list_buah = list(set_buah)
list_buah.sort()
print(list_buah)
```

```
>>> Tuple
4
>>> List
4
>>> Konversi tipe data
{'Jeruk', 'Markisa', 'Apel'}
['Apel', 'Jeruk', 'Markisa']
```

Tugas Praktek

In []:

```
# Data keuangan
keuangan = {
    'pengeluaran': [2, 2.5, 2.25, 2.5, 3.2, 2.5, 3.5, 4, 3],
    'pemasukan': [7.8, 7.5, 9, 7.6, 7.2, 7.5, 7, 10, 7.5]
}
# Perhitungan rata-rata pemasukan dan rata-rata pengeluaran
total_pengeluaran = 0
total_pemasukan = 0
for biaya in keuangan['pengeluaran']:
    total_pengeluaran += biaya
for biaya in keuangan['pemasukan']:
    total_pemasukan += biaya
rata_rata_pengeluaran = total_pengeluaran/len(keuangan['pengeluaran'])
rata_rata_pemasukan = total_pemasukan/len(keuangan['pemasukan'])
print(rata_rata_pengeluaran)
print(rata_rata_pemasukan)
```

```
2.8277777777777775
7.8999999999999995
```

String Manipulation dengan Python

Apa itu String Manipulation?

In []:

```
nama_produk = "Sepatu Niko"
nama_produk = nama_produk[:2] + "P" + nama_produk[3:9] + "K" + nama_produk[-1]
print(nama_produk)
print(nama_produk[:7])
print(nama_produk[7:])
print(len(nama_produk))
```

SePatu NiKo

SePatu
NiKo
11

Operator “+” untuk Tipe Data String

In []:

```
nama_depan = 'John'
nama_belakang = 'Doe'
nama_lengkap = nama_depan + ' ' + nama_belakang
print(nama_lengkap)
umur = '27 tahun'
alamat = 'Jl. Anggrek No. 100'
nama_umur_alamat = 'Hi, saya ' + nama_lengkap + ' umur ' + umur + ', tinggal di ' + alamat + '.'
```

John Doe

Hi, saya John Doe umur 27 tahun, tinggal di Jl. Anggrek No. 100.

Menghilangkan Spasi di Awal dan/atau di Akhir

In []:

```
# Fitur .strip()
print(">>> Fitur.strip()")
kata_sambutan = ' halo, selamat siang! '
kata_sambutan = kata_sambutan.strip()
print(kata_sambutan)
# Fitur .lstrip()
print(">>> Fitur.lstrip()")
kata_sambutan = ' halo, selamat siang! '
kata_sambutan = kata_sambutan.lstrip()
print(kata_sambutan)
# Fitur .rstrip()
print(">>> Fitur.rstrip()")
kata_sambutan = ' halo, selamat siang! '
kata_sambutan = kata_sambutan.rstrip()
print(kata_sambutan)
```

```
>>> Fitur.strip()
halo, selamat siang!
>>> Fitur.lstrip()
halo, selamat siang!
>>> Fitur.rstrip()
halo, selamat siang!
```

Merubah Caps pada String

In []:

```
# Fitur .capitalize()
print(">>> Fitur.capitalize()")
judul_buku = 'belajar bahasa Python'
print(judul_buku.capitalize())
# Fitur .lower()
print(">>> Fitur.lower()")
judul_buku = 'Belajar Bahasa PYTHON.'
print(judul_buku.lower())
# Fitur .upper()
print(">>> Fitur.upper()")
judul_buku = 'Belajar Bahasa PYTHON.'
print(judul_buku.upper())
```

```
>>> Fitur.capitalize()
Belajar bahasa python
>>> Fitur.lower()
belajar bahasa python.
>>> Fitur.upper()
BELAJAR BAHASA PYTHON.
```

Pemecahan, Penggabungan, dan Penggantian String

In []:

```
# Fitur .split()
print(">>> Fitur .split()")
bilangan = "ani dan budi dan wati dan johan"
karakter = bilangan.split("dan")
print(karakter)
kata = bilangan.split(" ")
print(kata)
# Fitur .join()
print(">>> Fitur .join()")
pemisah = " dan "
karakter = ["Ricky", "Peter", "Jordan"]
kalimat = pemisah.join(karakter)
print(kalimat)
# Fitur .replace()
print(">>> Fitur .replace()")
kalimat = "apel malang apel yang paling segar, apel sehat, apel nikmat"
kalimat = kalimat.replace("apel", "jeruk")
print(kalimat)
```

```
>>> Fitur .split()
['ani ', ' budi ', ' wati ', ' johan']
['ani', 'dan', 'budi', 'dan', 'wati', 'dan', 'johan']
>>> Fitur .join()
Ricky dan Peter dan Jordan
>>> Fitur .replace()
jeruk malang jeruk yang paling segar, jeruk sehat, jeruk nikmat
```

Menentukan Posisi dan Jumlah Sub-string pada String

In []:

```
teks = "Apel malang adalah apel termanis dibanding apel-apel lainnya"
# Fitur .find()
print(">>> Fitur .find()")
print(teks.find("Apel"))
print(teks.find("malang"))
# Fitur .count()
print(">>> Fitur .count()")
kemunculan_kata_apel = teks.count("apel")
print(kemunculan_kata_apel)
```

```
>>> Fitur .find()
0
5
>>> Fitur .count()
3
```

Menentukan String Apakah Diawali/Diakhiri oleh Sub-string

In []:

```
# Fitur .startswith()
print(">>> Fitur .startswith()")
teks = "Apel malang adalah apel termanis dibanding apel-apel lainnya"
print(teks.startswith("Apel"))
print(teks.startswith("apel"))
# Fitur .endswith()
print(">>> Fitur .endswith()")
print(teks.endswith("lainnya"))
print(teks.endswith("apel"))
```

```
>>> Fitur .startswith()
True
False
```



```
>>> Fitur.endswith()  
True  
False
```

Tugas Praktek

In []:

```
judul_artikel = [  
    "Buah Salak Baik untuk Mata", "Buah Salak Kaya Potasium",  
    "Buah Jeruk Kaya Vitamin C", "Buah Salak Kaya Manfaat",  
    "Salak Baik untuk Jantung", "Jeruk dapat Memperkuat Tulang",  
    "Jeruk Mencegah Penyakit Asma", "Jeruk Memperkuat Gigi",  
    "Jeruk Mencegah Kolesterol Jahat", "Salak Mencegah Diabetes",  
    "Salak Memperkuat Dinding Usus", "Salak Baik untuk Darah",  
    "Jeruk Kaya Manfaat untuk Jantung", "Salak si Kecil yang Baik",  
    "Jeruk dan Salak Buah Kaya Manfaat", "Buah Jeruk Enak",  
    "Tips Panen Jeruk Ribuan Kilo", "Tips Bertanam Salak",  
    "Salak Manis untuk Berbuka", "Jeruk Baik untuk Wajah"  
]  
jumlah_artikel_jeruk = 0  
jumlah_artikel_salak = 0  
for judul in judul_artikel:  
    if judul.count("Jeruk") > 0:  
        jumlah_artikel_jeruk += 1  
    if judul.count("Salak") > 0:  
        jumlah_artikel_salak += 1  
print(jumlah_artikel_jeruk)  
print(jumlah_artikel_salak)
```

```
10  
11
```

Tugas Praktek

In []:

```
judul_artikel = [  
    "Buah Salak Baik untuk Mata", "Buah Salak Kaya Potasium",  
    "Buah Jeruk Kaya Vitamin C", "Buah Salak Kaya Manfaat",  
    "Salak Baik untuk Jantung", "Jeruk dapat Memperkuat Tulang",  
    "Jeruk Mencegah Penyakit Asma", "Jeruk Memperkuat Gigi",  
    "Jeruk Mencegah Kolesterol Jahat", "Salak Mencegah Diabetes",  
    "Salak Memperkuat Dinding Usus", "Salak Baik untuk Darah",  
    "Jeruk Kaya Manfaat untuk Jantung", "Salak si Kecil yang Baik",  
    "Jeruk dan Salak Buah Kaya Manfaat", "Buah Jeruk Enak",  
    "Tips Panen Jeruk Ribuan Kilo", "Tips Bertanam Salak",  
    "Salak Manis untuk Berbuka", "Jeruk Baik untuk Wajah"  
]  
kata_positif = ["Kaya", "Baik", "Mencegah", "Memperkuat"]  
kata_positif_jeruk = 0  
kata_positif_salak = 0  
for judul in judul_artikel:  
    for kata in kata_positif:  
        if judul.count("Jeruk") > 0 and judul.count(kata) > 0:  
            kata_positif_jeruk += 1  
        if judul.count("Salak") > 0 and judul.count(kata) > 0:  
            kata_positif_salak += 1  
print(kata_positif_jeruk)  
print(kata_positif_salak)
```

```
8  
9
```

Functions

Fungsi Pertama

In []:

```
# Definisikan fungsi
def contoh_fungsi():
    print("Halo Dunia")
    print("Aku sedang belajar bahasa Python")
# Panggil fungsi yang telah didefinisikan
contoh_fungsi()
```

Halo Dunia

Aku sedang belajar bahasa Python

Fungsi Kedua

In []:

```
# Definsikan fungsi
def fungsi_dengan_argumen(nama_depan, nama_belakang):
    print(nama_depan+" "+nama_belakang)
# Panggil fungsi dengan memasukkan argumen
# nama_depan yaitu "John" dan nama_belakang "Doe"
fungsi_dengan_argumen("John", "Doe")
```

John Doe

Fungsi Ketiga

In []:

```
# Definsikan fungsi dengan nilai default argument kedua adalah "".
def fungsi_dengan_argumen(nama_depan, nama_belakang = ""):
    print(nama_depan+" "+nama_belakang)
# Panggil fungsi dengan memasukkan argumen nama_depan "John"
fungsi_dengan_argumen("John")
# Panggil fungsi dengan memasukkan argumen
# nama_depan yaitu "John" dan nama_belakang "Doe"
fungsi_dengan_argumen("John", "Doe")
```

John

John Doe

Tugas Praktek

In []:

```
# Dua buah data yang tersimpan dalam tipe list
data1 = [70, 70, 70, 100, 100, 100, 120, 120, 150, 150]
data2 = [50, 60, 60, 50, 70, 70, 100, 80, 100, 90]
# Definisikan fungsi hitng_rata_rata
def hitung_rata_rata(data):
    jumlah = 0
    for item in data:
        jumlah += item
    rata_rata = jumlah/len(data)
    return rata_rata
# Hitung nilai rata-rata dari kedua data yang dimiliki
print('Rata-rata data1:')
print(hitung_rata_rata(data1))
print('Rata-rata data2:')
print(hitung_rata_rata(data2))
```

Rata-rata data1:

105.0

Rata-rata data2:

73.0

Tugas Praktek

In []:

```
# Dua buah data yang tersimpan dalam tipe list
data1 = [70, 70, 70, 100, 100, 100, 120, 120, 150, 150]
data2 = [50, 60, 60, 50, 70, 70, 100, 80, 100, 90]
# Fungsi rata-rata data
def hitung_rata_rata(data):
    jumlah = 0
    for item in data:
        jumlah += item
    rata_rata = jumlah/len(data)
    return rata_rata
# Definisikan fungsi hitung standar deviasi
def hitung_standar_deviasi(data):
    rata_rata_data = hitung_rata_rata(data)
    varians = 0
    for item in data:
        varians += (item - rata_rata_data) ** 2
    varians /= len(data)
    standar_deviasi = varians ** (1/2)
    return standar_deviasi
# Hitung nilai standar deviasi dari kedua data yang dimiliki
print('Standar deviasi data1:')
print(hitung_standar_deviasi(data1))
print('Standar deviasi data2:')
print(hitung_standar_deviasi(data2))
```

Standar deviasi data1:
14.93311056335886
Standar deviasi data2:
6.025948374480153

Tugas Praktek

In []:

```
# Data properti
tabel_properti = {
    'luas_tanah': [70, 70, 70, 100, 100, 100, 120, 120, 150, 150],
    'luas_bangunan': [50, 60, 60, 50, 70, 70, 100, 80, 100, 90],
    'jarak': [15, 30, 55, 30, 25, 50, 20, 50, 50, 15],
    'harga': [500, 400, 300, 700, 1000, 650, 2000, 1200, 1800, 3000]
}
# Fungsi rata-rata data
def hitung_rata_rata(data):
    jumlah = 0
    for item in data:
        jumlah += item
    rata_rata = jumlah/len(data)
    return rata_rata
# Fungsi hitung standar deviasi
def hitung_standar_deviasi(data):
    rata_rata_data = hitung_rata_rata(data)
    varians = 0
    for item in data:
        varians += (item - rata_rata_data) ** 2
    varians /= len(data)
    standar_deviasi = varians ** (1/2)
    return standar_deviasi
# Definisikan fungsi untuk menghitung rata-rata dan standar deviasi
# setiap kolom pada tabel_properti yang diberikan oleh key dict.
def deskripsi_properti(tabel):
    for key in tabel.keys():
        print('Rata-rata ' + key + ':')
        print(hitung_rata_rata(tabel[key]))
        print('Standar deviasi ' + key + ':')
        print(hitung_standar_deviasi(tabel[key]))
        print('')
# Panggil fungsi deskripsi_properti untuk menghitung rata-rata
# dan standar deviasi setiap kolom pada tabel_properti
deskripsi_properti(tabel_properti)
```

Rata-rata luas_tanah:
105.0

105.0
Standar deviasi luas_tanah:
14.93311056335886

Rata-rata luas_bangunan:
73.0
Standar deviasi luas_bangunan:
6.025948374480153

Rata-rata jarak:
34.0
Standar deviasi jarak:
6.24005184770928

Rata-rata harga:
1155.0
Standar deviasi harga:
587.0594351517378

Manipulasi Berkas Teks dan Library Matematika pada Python

Membaca Berkas Teks – Part 1

In []:

```
from google.colab import drive
drive.mount('/content/drive')
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aob&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
.....

Mounted at /content/drive

In []:

```
!ls drive/'My Drive'/'Colab Notebooks'/'DQLab'/'Dataset'
```

hello.txt

In []:

```
# Membaca file hello.txt dengan fungsi read()
print(">>> Membaca file hello.txt dengan fungsi read()")
file = open('drive/My Drive/Colab Notebooks/DQLab/Dataset/hello.txt', 'r')
content = file.read()
file.close()
print(content)
# Membaca file hello.txt dengan fungsi readline()
print(">>> Membaca file hello.txt dengan fungsi readline()")
file = open('drive/My Drive/Colab Notebooks/DQLab/Dataset/hello.txt', 'r')
first_line = file.readline()
second_line = file.readline()
file.close()
print(first_line)
print(second_line)
```

```
>>> Membaca file hello.txt dengan fungsi read()
Kita sedang belajar Python
Tepatnya belajar memanipulasi berkas teks
Memanipulasi berkas dengan Python sangatlah mudah!
>>> Membaca file hello.txt dengan fungsi readline()
Kita sedang belajar Python
```

Tepatnya belajar memanipulasi berkas teks

Tepatnya belajar memanipulasi berkas teks

Membaca Berkas Teks – Part 2

In []:

```
# Membaca file hello.txt dengan fungsi readlines()
print(">>> Membaca file hello.txt dengan fungsi readlines()")
file = open('drive/My Drive/Colab Notebooks/DQLab/Dataset/hello.txt', 'r')
all_lines = file.readlines()
file.close()
print(all_lines)
# Membaca file hello.txt dengan menerapkan looping
print(">>> Membaca file hello.txt dengan menerapkan looping")
file = open('drive/My Drive/Colab Notebooks/DQLab/Dataset/hello.txt', 'r')
for line in file:
    print(line)
```

```
>>> Membaca file hello.txt dengan fungsi readlines()
['Kita sedang belajar Python\n', 'Tepatnya belajar memanipulasi berkas teks\n', 'Memanipulasi berkas dengan Python sangatlah mudah!']
>>> Membaca file hello.txt dengan menerapkan looping
Kita sedang belajar Python
```

Tepatnya belajar memanipulasi berkas teks

Memanipulasi berkas dengan Python sangatlah mudah!

Menulis Berkas Teks – Part 1

In []:

```
# Menulis ke file hello_write.txt
file = open('drive/My Drive/Colab Notebooks/DQLab/Dataset/hello.txt', 'w')
file.write("Sekarang kita belajar menulis dengan menggunakan Python")
file.write("Menulis konten file dengan mode w (write).")
file.close()
```

Menulis Berkas Teks – Part 2

In []:

```
# Menulis ke file dengan mode append
file = open('drive/My Drive/Colab Notebooks/DQLab/Dataset/hello.txt', 'a')
file.writelines([
    "Sekarang kita belajar menulis dengan menggunakan Python",
    "Menulis konten file dengan mode a (append).")
])
file.close()
```

Fungsi dalam Library Matematika – Part 1

In []:

```
# Import library math
import math
# Fungsi math.ceil()
print(">>> Fungsi math.ceil()")
x = 10.32
y = 13.87
x_ceil = math.ceil(x)
y_ceil = math.ceil(y)
print(x_ceil)
print(y_ceil)
# Fungsi math.floor()
print(">>> Fungsi math.floor()")
x_floor = math.floor(x)
y_floor = math.floor(y)
```

```

print(x_floor)
print(y_floor)
# Fungsi math.fabs()
print(">>> Fungsi math.fabs() ")
x = 10.32
y = -13.87
x = math.fabs(x)
y = math.fabs(y)
print(x)
print(y)
# Fungsi math.factorial()
print(">>> Fungsi math.factorial() ")
x_factorial = math.factorial(5)
print(x_factorial)
# Fungsi math.fsum()
print(">>> Fungsi math.fsum() ")
x = [1, 2, 3, 4, 5, 6, -6, -5, -4]
total = math.fsum(x)
print(total)

```

```

>>> Fungsi math.ceil()
11
14
>>> Fungsi math.floor()
10
13
>>> Fungsi math.fabs()
10.32
13.87
>>> Fungsi math.factorial()
120
>>> Fungsi math.fsum()
6.0

```

Fungsi dalam Library Matematika – Part 2

In []:

```

# Import library math
import math
# Fungsi math.log()
print(">>> Fungsi math.log() ")
# x = log basis 2 dari 8
x = math.log(8, 2)
# y = log basis 3 dari 81
y = math.log(81, 3)
# z = log basis 10 dari 10000
z = math.log(10000, 10)
print(x)
print(y)
print(z)
# Fungsi math.sqrt()
print(">>> Fungsi math.sqrt() ")
# akar kuadrat dari 100
x = math.sqrt(100)
print(x)
# akar kuadrat dari 2
y = math.sqrt(2)
print(y)
# Fungsi math.copysign()
print(">>> Fungsi math.copysign() ")
x = 10.32
y = -13.87
z = -15
x = math.copysign(x, z)
y = math.copysign(y, z)
z = math.copysign(z, 10)
print(x)
print(y)
print(z)

```

```

>>> Fungsi math.log()

```

```

>>> Fungsi math.log()
3.0
4.0
4.0
>>> Fungsi math.sqrt()
10.0
1.4142135623730951
>>> Fungsi math.copysign()
-10.32
-13.87
15.0

```

Mini Quiz

In []:

```

# STEP 1:
# Baca file "harga_rumah.txt"
file_harga_rumah = open('drive/My Drive/Colab Notebooks/DQLab/Dataset/harga_rumah.txt', 'r')

data_harga_rumah = file_harga_rumah.readlines()
file_harga_rumah.close()
# Buat list of dict dengan nama harga rumah
key_harga_rumah = data_harga_rumah[0].replace("\n", "").split(",")
harga_rumah = []
for baris in data_harga_rumah[1:]:
    baris_harga_rumah = baris.replace("\n", "").split(",")
    dict_harga_rumah = dict()
    for i in range(len(baris_harga_rumah)):
        dict_harga_rumah[key_harga_rumah[i]] = baris_harga_rumah[i]
    harga_rumah.append(dict_harga_rumah)
print(harga_rumah)

# STEP 2:
# Buat fungsi get_all_specified_attribute yang menerima parameter list_of_dictionary
# (tipe data list yang berisikan sekumpulan tipe data dictionary) dan specified_key
# (tipe data string). Fungsi akan mengembalikan sebuah list yang berisikan seluruh
# atribut dengan kunci (key) specified_key.
def get_all_specified_attributes(list_of_dictionary, specified_key):
    list_attributes = []
    for data in list_of_dictionary:
        attribute = data[specified_key]
        list_attributes.append(attribute)
    return list_attributes

# STEP 3:
# Buat fungsi fungsi min_value yang menerima parameter list_attributes (berupa
# tipe data list) dan mengembalikan nilai terkecil dalam list_attributes
def min_value(list_attributes):
    min_attribute = 9999
    for attr in list_attributes:
        if int(attr) < min_attribute:
            min_attribute = int(attr)
    return min_attribute

# Buat fungsi dan max_value yang menerima parameter list_attribute dan
# mengembalikan nilai terbesar dalam list_attributes.
def max_value(list_attributes):
    max_attribute = -9999
    for attr in list_attributes:
        if int(attr) > max_attribute:
            max_attribute = int(attr)
    return max_attribute

# STEP 4:
# Buat fungsi transform_attribute yang menerima parameter attr (sebuah
# bilangan), max_attr (sebuah bilangan) dan min_attr (sebuah bilangan)
# yang mengembalikan nilai transformasi dari sebuah attribute.
def transform_attribute(attr, max_attr, min_attr):
    nilai_transformasi = (attr - min_attr) / (max_attr - min_attr)
    return nilai_transformasi

# STEP 5:
# Buat fungsi data_transformation yang menerima parameter list_of_dictionary
# (sebuah list yang berisikan tipe data dictionary) dan list_attribute_names

```

```

# (sebuah list yang berisikan tipe data string) mengembalikan hasil
# transformasi data dari list_of_dictionary berdasarkan list_attribute_names
# dan attr_info telah dispesifikasikan.
def data_transformation(list_of_dictionary, list_attribute_names):
    attr_info = {}
    for attr_name in list_attribute_names:
        specified_attributes = get_all_specified_attributes(list_of_dictionary, attr_name)
        max_attr = max_value(specified_attributes)
        min_attr = min_value(specified_attributes)
        attr_info[attr_name] = {'max': max_attr, 'min': min_attr}
        data_idx = 0
        while (data_idx < len(list_of_dictionary)):
            list_of_dictionary[data_idx][attr_name] = transform_attribute(int(list_of_dictionary[
ata_idx][attr_name]), max_attr, min_attr)
            data_idx += 1
        return list_of_dictionary, attr_info
# STEP 6:
# Berdasarkan data baru dan attr_info ini, buat fungsi transform_data yang
# menerima parameter data dan attr_info dan mengembalikan nilai atribut
# dari data baru yang telah ditransformasikan.
def transform_data(data, attr_info):
    for key_name in data.keys():
        data[key_name] = (data[key_name] - attr_info[key_name]['min']) / (
            attr_info[key_name]['max'] - attr_info[key_name]['min'])

    return data
# STEP 7:
# Buat fungsi yang digunakan untuk sistem prediksi harga berdasarkan
# nilai kemiripan atribut!
def abs_value(value):
    if value < 0:
        return -value
    else:
        return value
def price_based_on_similarity(data, list_of_data):
    prediksi_harga = 0
    perbedaan_terkecil = 999
    for data_point in list_of_data:
        perbedaan= abs_value(data['tanah'] - data_point['tanah'])
        perbedaan+= abs_value(data['bangunan'] - data_point['bangunan'])
        perbedaan+= abs_value(data['jarak_ke_pusat'] - data_point['jarak_ke_pusat'])
        if perbedaan < perbedaan_terkecil:
            prediksi_harga = data_point['harga']
            perbedaan_terkecil = perbedaan
    return prediksi_harga
# STEP 8:
# Hitung harga rumah yang telah ditransformasikan ke dalam variabel
# harga_rumah berikut dengan atributnya attr_info
harga_rumah, attr_info = data_transformation(harga_rumah,
                                             ['tanah','bangunan','jarak_ke_pusat'])

# Gunakan variabel data untuk memprediksi harga rumah
data = {'tanah': 110, 'bangunan': 80, 'jarak_ke_pusat': 35}
# Transformasikan data tersebut dengan dengan menggunakan attr_info yang telah
# diperoleh yang kembali disimpan ke variabel data.
data = transform_data(data, attr_info)
# Hitunglah prediksi harga dari variabel data tersebut.
harga = price_based_on_similarity(data, harga_rumah)
print("Prediksi harga rumah: ", harga)

```

```

[{'tanah': '70', 'bangunan': '50', 'jarak_ke_pusat': '15', 'harga': '500'}, {'tanah': '70',
', 'bangunan': '60', 'jarak_ke_pusat': '30', 'harga': '400'}, {'tanah': '70', 'bangunan':
'60', 'jarak_ke_pusat': '55', 'harga': '300'}, {'tanah': '100', 'bangunan': '50', 'jarak_
ke_pusat': '30', 'harga': '700'}, {'tanah': '100', 'bangunan': '70', 'jarak_ke_pusat': '2
5', 'harga': '1000'}, {'tanah': '100', 'bangunan': '70', 'jarak_ke_pusat': '50', 'harga':
'650'}, {'tanah': '120', 'bangunan': '100', 'jarak_ke_pusat': '20', 'harga': '2000'}, {'t
anah': '120', 'bangunan': '80', 'jarak_ke_pusat': '50', 'harga': '1200'}, {'tanah': '150'
, 'bangunan': '100', 'jarak_ke_pusat': '50', 'harga': '1800'}, {'tanah': '150', 'bangunan
': '90', 'jarak_ke_pusat': '15', 'harga': '3000'}]
Prediksi harga rumah: 1200

```

Kesimpulan

1. Memahami & mempraktekkan penggunaan collection manipulations (list, tuple, set dan dictionary) pada Python.
2. Memahami & mempraktekkan penggunaan string manipulation pada Python.
3. Memahami & mempraktekkan pembuatan fungsi yang didefinisikan sendiri untuk kebutuhan tertentu dengan menggunakan Python.
4. Memahami & mempraktekkan cara membaca data dari file menggunakan Python
5. Mengerjakan latihan dalam membuat program Python untuk penyelesaian kasus bisnis sederhana.