

Mari Mengenal Python

Program pertama: "Hello World"

In []:

```
print("Hello World!")
```

Hello World!

Program Pertamaku

In []:

```
print("Halo Dunia")  
print("Riset Bahasa Python")
```

Halo Dunia

Riset Bahasa Python

Struktur Program Python - Part 1

In []:

```
# Statement  
print("Belajar Python menyenangkan")  
print("Halo Dunia")  
print("Hello World!")  
# Variables & Literals  
bilangan1 = 5  
bilangan2 = 10  
kalimat1 = "Belajar Bahasa Python"  
# Operators  
print(bilangan1 + bilangan2)
```

Belajar Python menyenangkan

Halo Dunia

Hello World!

15

Tugas Praktek

In []:

```
bilangan1 = 20  
bilangan2 = 10  
print(bilangan1 - bilangan2)
```

10

Tugas Praktek

In []:

```
harga_asli = 20000  
potongan = 2000  
harga_setelah_potongan = harga_asli - potongan  
harga_final = harga_setelah_potongan * 1.1  
print(harga_final)
```

19800.0

Python Variables & Data Types

Sequence Type – Part 1

In []:

```
contoh_list = [1, 'dua', 3, 4.0, 5]
print(contoh_list[0])
print(contoh_list[3])
contoh_list = [1, 'dua', 3, 4.0, 5]
contoh_list[3] = 'empat'
print(contoh_list[3])
```

```
1
4.0
empat
```

Sequence Type – Part 2

In []:

```
contoh_tuple = ('Januari', 'Februari', 'Maret', 'April')
print(contoh_tuple[0])
contoh_tuple = ('Januari', 'Februari', 'Maret', 'April')
contoh_tuple[0] = "Desember"
```

```
Januari
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-7-bf5fba48fc27> in <module>()
      2 print(contoh_tuple[0])
      3 contoh_tuple = ('Januari', 'Februari', 'Maret', 'April')
----> 4 contoh_tuple[0] = "Desember"
```

TypeError: 'tuple' object does not support item assignment

Set Type

In []:

```
contoh_list = ['Dewi', 'Budi', 'Cici', 'Linda', 'Cici']
print(contoh_list)
contoh_set = {'Dewi', 'Budi', 'Cici', 'Linda', 'Cici'}
print(contoh_set)
contoh_frozen_set = frozenset(['Dewi', 'Budi', 'Cici', 'Linda', 'Cici'])
print(contoh_frozen_set)
```

```
['Dewi', 'Budi', 'Cici', 'Linda', 'Cici']
{'Budi', 'Linda', 'Cici', 'Dewi'}
{'Budi', 'Linda', 'Cici', 'Dewi'}
```

Mapping Type

In []:

```
person = {'nama': 'John Doe', 'pekerjaan': 'Programmer'}
print(person['nama'])
print(person['pekerjaan'])
```

```
John Doe
Programmer
```

Tugas Praktek

In []:

```
sepatu = {"nama": "Sepatu Niko", "harga": 150000, "diskon": 30000}
baju = {"nama": "Baju Unikloh", "harga": 80000, "diskon": 8000}
```

```
celana = {"nama": "Celana Lepas", "harga": 200000, "diskon": 60000}
```

Tugas Praktek

In []:

```
sepatu = {"nama": "Sepatu Niko", "harga": 150000, "diskon": 30000}
baju = {"nama": "Baju Unikloh", "harga": 80000, "diskon": 8000}
celana = {"nama": "Celana Lepas", "harga": 200000, "diskon": 60000}
daftar_belanja = [sepatu, baju, celana]
```

Tugas Praktek

In []:

```
# Data yang dinyatakan ke dalam dictionary
sepatu = {"nama": "Sepatu Niko", "harga": 150000, "diskon": 30000}
baju = {"nama": "Baju Unikloh", "harga": 80000, "diskon": 8000}
celana = {"nama": "Celana Lepas", "harga": 200000, "diskon": 60000}
# Hitunglah harga masing-masing data setelah dikurangi diskon
harga_sepatu = sepatu["harga"] - sepatu["diskon"]
harga_baju = baju["harga"] - baju["diskon"]
harga_celana = celana["harga"] - celana["diskon"]
# Hitung harga total
total_harga = harga_sepatu + harga_baju + harga_celana
# Hitung harga kena pajak
total_pajak = total_harga * 0.1
# Cetak total_harga + total_pajak
print(total_harga + total_pajak)
```

365200.0

Python Operators

Nilai Prioritas Operator dalam Python – Part 1

In []:

```
# Kode awal
total_harga = 150000
potongan_harga = 0.3
pajak = 0.1 # pajak dalam persen ~ 10%
harga_bayar = 1 - 0.3 # baris pertama
harga_bayar *= total_harga # baris kedua
pajak_bayar = pajak * harga_bayar # baris ketiga
harga_bayar += pajak_bayar # baris ke-4
print("Kode awal - harga_bayar=", harga_bayar)
# Penyederhanaan baris kode dengan menerapkan prioritas operator
total_harga = 150000
potongan_harga = 0.3
pajak = 0.1 # pajak dalam persen ~ 10%
harga_bayar = (1 - 0.3) * total_harga #baris pertama
harga_bayar += harga_bayar * pajak # baris kedua
print("Penyederhanaan kode - harga_bayar=", harga_bayar)
```

Kode awal - harga_bayar= 115500.0

Penyederhanaan kode - harga_bayar= 115500.0

Tugas Praktek

In []:

```
sepatu = { "nama" : "Sepatu Niko", "harga": 150000, "diskon": 30000 }
baju = { "nama" : "Baju Unikloh", "harga": 80000, "diskon": 8000 }
celana = { "nama" : "Celana Lepas", "harga": 200000, "diskon": 60000 }
harga_sepatu = sepatu["harga"] - sepatu["diskon"]
harga_baju = baju["harga"] - baju["diskon"]
```

```
harga_celana = celana["harga"] - celana["diskon"]
total_harga = (harga_sepatu + harga_baju + harga_celana) * 1.1
print(total_harga)
```

365200.00000000006

Pythons Conditioning & Looping

Python Conditioning for Decision – Part 2

In []:

```
# Statement if
x = 4
if x % 2 == 0: # jika sisa bagi x dengan 2 sama dengan 0
    print("x habis dibagi dua") # statemen aksi lebih menjorok ke dalam
# Statement if ... elif ... else
x = 7
if x % 2 == 0: # jika sisa bagi x dengan 2 sama dengan 0
    print("x habis dibagi dua")
elif x % 3 == 0: # jika sisa bagi x dengan 3 sama dengan 0
    print("x habis dibagi tiga")
elif x % 5 == 0: # jika sisa bagi x dengan 5 sama dengan 0
    print("x habis dibagi lima")
else:
    print("x tidak habis dibagi dua, tiga ataupun lima")
```

x habis dibagi dua
x tidak habis dibagi dua, tiga ataupun lima

Python Conditioning for Decision – Part 3

In []:

```
jam = 13
if jam >= 5 and jam < 12: # selama jam di antara 5 s.d. 12
    print("Selamat pagi!")
elif jam >= 12 and jam < 17: # selama jam di antara 12 s.d. 17
    print("Selamat siang!")
elif jam >= 17 and jam < 19: # selama jam di antara 17 s.d. 19
    print("Selamat sore!")
else: # selain kondisi di atas
    print("Selamat malam!")
```

Selamat siang!

Tugas Praktek

In []:

```
tagihan_ke = 'Mr. Yoyo'
warehousing = { 'harga_harian': 1000000, 'total_hari':15 }
cleansing = { 'harga_harian': 1500000, 'total_hari':10 }
integration = { 'harga_harian':2000000, 'total_hari':15 }
transform = { 'harga_harian':2500000, 'total_hari':10 }
sub_warehousing = warehousing['harga_harian'] * warehousing['total_hari']
sub_cleansing = cleansing['harga_harian'] * cleansing['total_hari']
sub_integration = integration['harga_harian'] * integration['total_hari']
sub_transform = transform['harga_harian'] * transform['total_hari']
total_harga = sub_warehousing + sub_cleansing + sub_integration + sub_transform
print("Tagihan kepada:")
print(tagihan_ke)
print("Selamat pagi, anda harus membayar tagihan sebesar:")
print(total_harga)
```

Tagihan kepada:
Mr. Yoyo
Selamat pagi, anda harus membayar tagihan sebesar:
85000000

Tugas Praktek

In []:

```
jam = 17
tagihan_ke = 'Mr. Yoyo'
warehousing = { 'harga_harian': 1000000, 'total_hari':15 }
cleansing = { 'harga_harian': 1500000, 'total_hari':10 }
integration = { 'harga_harian':2000000, 'total_hari':15 }
transform = { 'harga_harian':2500000, 'total_hari':10 }
sub_warehousing = warehousing['harga_harian']*warehousing['total_hari']
sub_cleansing = cleansing['harga_harian']*cleansing['total_hari']
sub_integration = integration['harga_harian']*integration['total_hari']
sub_transform = transform['harga_harian']*transform['total_hari']
total_harga = sub_warehousing+sub_cleansing+sub_integration+sub_transform
print("Tagihan kepada:")
print(tagihan_ke)
if jam > 19:
    print("Selamat malam, anda harus membayar tagihan sebesar:")
elif jam > 17:
    print("Selamat sore, anda harus membayar tagihan sebesar:")
elif jam > 12:
    print("Selamat siang, anda harus membayar tagihan sebesar:")
else:
    print("Selamat pagi, anda harus membayar tagihan sebesar:")
print(total_harga)
```

Tagihan kepada:
Mr. Yoyo
Selamat siang, anda harus membayar tagihan sebesar:
85000000

Python while loops – Part 1

In []:

```
# Tagihan
tagihan = [50000, 75000, 125000, 300000, 200000]
# Tanpa menggunakan while loop
total_tagihan = tagihan[0] + tagihan[1] + tagihan[2] + tagihan[3] + tagihan[4]
print(total_tagihan)
# Dengan menggunakan while loop
i = 0 # sebuah variabel untuk mengakses setiap elemen tagihan satu per satu
jumlah_tagihan = len(tagihan) # panjang (jumlah elemen dalam) list tagihan
total_tagihan = 0 # mula-mula, set total_tagihan ke 0
while i < jumlah_tagihan: # selama nilai i kurang dari jumlah_tagihan
    total_tagihan += tagihan[i] # tambahkan tagihan[i] ke total_tagihan
    i += 1 # tambahkan nilai i dengan 1 untuk memproses tagihan selanjutnya.
print(total_tagihan)
```

750000
750000

Python while loops – Part 2

In []:

```
tagihan = [50000, 75000, -150000, 125000, 300000, -50000, 200000]
i = 0
jumlah_tagihan = len(tagihan)
total_tagihan = 0
while i < jumlah_tagihan:
    # jika terdapat tagihan ke-i yang bernilai minus (di bawah nol),
    # pengulangan akan dihentikan
    if tagihan[i] < 0:
        total_tagihan = -1
        print("terdapat angka minus dalam tagihan, perhitungan dihentikan!")
        break
```

```
        total_tagihan += tagihan[i]
        i += 1
print(total_tagihan)
```

terdapat angka minus dalam tagihan, perhitungan dihentikan!
-1

Python while loops – Part 3

In []:

```
tagihan = [50000, 75000, -150000, 125000, 300000, -50000, 200000]
i = 0
jumlah_tagihan = len(tagihan)
total_tagihan = 0
while i < jumlah_tagihan:
    # jika terdapat tagihan ke-i yang bernilai minus (di bawah nol),
    # abaikan tagihan ke-i dan lanjutkan ke tagihan berikutnya
    if tagihan[i] < 0:
        i += 1
        continue
    total_tagihan += tagihan[i]
    i += 1
print(total_tagihan)
```

750000

Python for loops – Part 1

In []:

```
list_tagihan = [50000, 75000, -150000, 125000, 300000, -50000, 200000]
total_tagihan = 0
for tagihan in list_tagihan: # untuk setiap tagihan dalam list_tagihan
    total_tagihan += tagihan # tambahkan tagihan ke total_tagihan
print(total_tagihan)
```

550000

Python for loops – Part 2

In []:

```
list_tagihan = [50000, 75000, -150000, 125000, 300000, -50000, 200000]
total_tagihan = 0
for tagihan in list_tagihan:
    if tagihan < 0:
        print("terdapat angka minus dalam tagihan, perhitungan dihentikan!")
        break
    total_tagihan += tagihan
print(total_tagihan)
```

terdapat angka minus dalam tagihan, perhitungan dihentikan!
125000

Python for loops – Part 3

In []:

```
list_daerah = ['Malang', 'Palembang', 'Medan']
list_buah = ['Apel', 'Duku', 'Jeruk']
for nama_daerah in list_daerah:
    for nama_buah in list_buah:
        print(nama_buah+" "+nama_daerah)
```

Apel Malang
Duku Malang
Jeruk Malang
Apel Palembang
Duku Palembang

Duku Palembang
Jeruk Palembang
Apel Medan
Duku Medan
Jeruk Medan

Tugas Praktek

In []:

```
list_cash_flow = [
2500000, 5000000, -1000000, -2500000, 5000000, 10000000,
-5000000, 7500000, 10000000, -1500000, 25000000, -2500000
]
total_pengeluaran, total_pemasukan = 0, 0
for dana in list_cash_flow:
    if dana > 0:
        total_pemasukan += dana
    else:
        total_pengeluaran += dana
total_pengeluaran *= -1
print(total_pengeluaran)
print(total_pemasukan)
```

12500000
65000000

Mini Quiz

In []:

```
# Data
uang_jalan = 1500000
jumlah_hari = 31
list_plat_nomor = [8993, 2198, 2501, 2735, 3772, 4837, 9152]
# Pengecekan kendaraan dengan nomor pelat ganjil atau genap
# Deklarasikan kendaraan_genap dan kendaraan_ganjil = 0
kendaraan_genap = 0
kendaraan_ganjil = 0
for plat_nomor in list_plat_nomor:
    if plat_nomor %2 == 0:
        kendaraan_genap += 1
    else:
        kendaraan_ganjil += 1
# Total pengeluaran untuk kendaraan dengan nomor pelat ganjil
# dan genap dalam 1 bulan
i = 1
total_pengeluaran = 0
while i <= jumlah_hari:
    if i % 2 == 0:
        total_pengeluaran += (kendaraan_genap * uang_jalan)
    else:
        total_pengeluaran += (kendaraan_ganjil * uang_jalan)
    i += 1
# Cetak total pengeluaran
print(total_pengeluaran)
```

163500000

Kesimpulan

1. Alasan Python secara luas digunakan dalam komputasi saintifik, web, ranah data (data domain).
2. Konstruksi dari struktur bahasa pemrograman Python.
3. Teknik mempraktekkan penggunaan tipe data pada Python.
4. Teknik mempraktekkan penggunaan jenis-jenis operator pada Python.
5. Teknik mempraktekkan penggunaan pengkondisian untuk pengambilan keputusan dan perulangan pada Python.
6. Program Python untuk penyelesaian kasus bisnis sederhana.

