LAPORAN PEMROGRAMAN FUNGSI

DOSEN PENGAMPU : Dian Septiani Santoso M. Kom.

Disusun untuk Memenuhi Tugas

Mata Kuliah:

PRAKTIKUM PEMROGRAMAN

Oleh:

Wahyu Ikbal Maulana

NRP 3323600056

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

PRODI SAINS DATA TERAPAN

SEPTEMBER 2023

Latihan 1

```
def is_int(data) :
    if type(data) == int:
        return True
    elif type(data) == float:
        return False

print(is_int(5) )
print(is_int(5.0))
print(is_int("5"))
```

```
True
False
None
```

Kode di atas berisi perulangan is_int yang berisi percabangan, jika type data bernilai integer maka akan menghasilkan output True dan False jika bernilai False. Karena disitu tidak ada else, maka percabangan di luar tipe data integer dan float maka akan menghasilkan None.

Terakhir print list angka genap. Dan kode berjalan tanpa error

```
def even_num_lst(ran):
    lst = []
    for num in range (ran) :
        if num % 2 == 0:
            lst.append(num)
    return lst

print(even_num_lst(11))
```

```
[0, 2, 4, 6, 8, 10]
```

Kode di atas berisi perulangan even_num_lst. Yang dimana merupakan fungsi untuk menampilkan bilangan genap hingga batas yang telah ditentukan.

lst berisi list kosong, lalu untuk setiap num di dalam parameter ran maka akan mengecek apakah parameter dibagi 2 sama dengan nol/sama dengan genap. Lalu masukkan ke list lst hingga batas akhir.

Terakhir print list angka genap. Dan kode berjalan tanpa error

```
def list_updater(lst) :
    upd_list = []
    for elem in lst:
        elem **= 2
        upd_list.append(elem)
    return upd_list
```

```
foo = [1,2,3,4,5]
print(list_updater(foo))
```

```
[1, 4, 9, 16, 25]
```

Kode di atas berisi perulangan list_updater. Fungsi ini berfungsi untuk mengubah setiap elemen dalam list lst menjadi kuadrat dari elemen tersebut.

upd_list berisi list kosong, lalu untuk setiap elem di dalam parameter lst maka elem akan dikuadratkan dan dimasukkkan kedalam upd_list.

Ketika kode dijalankan, sebuah list foo dengan angka-angka dari 1 hingga 5 dibuat. Kemudian, fungsi list_updater dipanggil dengan foo sebagai argumen, dan hasilnya dicetak. Hasilnya adalah list baru yang berisi kuadrat dari angka-angka dalam foo, yaitu [1, 4, 9, 16, 25].

Latihan 2

Buat fungsi untuk menghitung jumlah hari dalam bulan dan tahun tertentu.
Misal:
Masukkan bulan: 2
Masukkan tahun: 1993
Jumlah hari pada bulan Februari 1993 adalah 28

Masukkan bulan: 10
Masukkan tahun: 2000
Jumlah hari pada bulan Oktober 2000 adalah 31

```
#Fungsi untuk menghitung jumlah hari dalam bulan dan tahun tertentu
import calendar

def jumlah_hari(bulan, tahun):
        jumlah_hari = calendar.monthrange(tahun, bulan)
        return jumlah_hari
```

```
Masukkan bulan 12
Masukkan tahun 1000
Jumlah hari dalam bulan 12 tahun 1000 adalah 31.
```

```
#Tanpa input
jumlah_hari(10,2000)
print(f"Jumlah hari dalam bulan {bulan} tahun {tahun} adalah
{hasil[1]}.")
```

```
Jumlah hari dalam bulan 12 tahun 1000 adalah 31.
```

```
#Dengan input
bulan = int(input("Masukkan bulan "))
tahun = int(input("Masukkan tahun "))

hasil = jumlah_hari(bulan, tahun)
print(f"Jumlah hari dalam bulan {bulan} tahun {tahun} adalah
{hasil[1]}.")
```

Pertama import calendar

Masukkan baris kode untuk inputan bulan dan tahun.

Lalu untuk menghitung jumlah harinya disini perlu dideklarasikan def fungsi dengan isinya sebuah perhitungan untuk menghitung berapa jumlah hari dengan inputan bulan dan tahun. Dengan memakai fungsi monthrange dan parameter tahun dan bulan, didapat jumlah hari dalam bulan dan tahun. Lalu direturn dengan variabel yang telah diperhitungkan, yaitu variabel jumlah_hari.

Latihan 3

Buat fungsi untuk melakukan pengecekan bilangan prima. Bilangan prima adalah bilangan bulat lebih dari 1 yang tidak bisa dibagi oleh bilangan selain 1 dan bilangan itu sendiri.

```
#Fungsi untuk mencari bilangan prima
def is_prime (number):
  for i in range(2, number):
    if number % i == 0:
      return False
  return True

def prima_batas(atas):
    lst = []
    for i in range(1,batas + 1):
        if is_prime(i):
            lst.append(i)
    return lst

Masukan batas atas: 50
Bilangan prima pada rentang 1 s.d 50 adalah [1, 2, 3, 5, 7, 11, 13,
17, 19, 23, 29, 31, 37, 41, 43, 47]

#Tanpa input
prima_batas(50)

[1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

#Dengan input
batas_atas = int(input("Masukan batas atas: "))
```

```
prima_batas(batas_atas)
print(f"Bilangan prima pada rentang 1 s.d {batas} adalah
{prima_batas(batas)}")

Masukan batas atas: 50
Bilangan prima pada rentang 1 s.d 50 adalah [1, 2, 3, 5, 7, 11, 13,
17, 19, 23, 29, 31, 37, 41, 43, 47]
```

Pendeklarasian def dan mengisi fungsi untuk bilangan prima. Disini terdapat 2 fungsi

1. is_prime(number): Ini adalah fungsi yang digunakan untuk memeriksa apakah suatu angka number adalah bilangan prima atau tidak. Fungsi ini menggunakan perulangan for untuk mencoba semua angka dari 2 hingga number - 1. Jika ada angka lain selain 1 dan number sendiri yang dapat membagi number tanpa sisa, maka fungsi ini mengembalikan False, menunjukkan bahwa number bukan bilangan prima. Jika tidak ada angka lain yang dapat membagi number tanpa sisa, fungsi ini mengembalikan True, menunjukkan bahwa number adalah bilangan prima.
2. prima_batas(atas): Ini adalah fungsi yang digunakan untuk mencari dan mengumpulkan semua bilangan prima dalam rentang 1 hingga atas. Fungsi ini membuat list kosong lst untuk menyimpan bilangan prima yang ditemukan. Kemudian, dengan menggunakan perulangan for, fungsi ini memeriksa setiap angka dari 1 hingga atas dengan memanggil fungsi is_prime. Jika angka tersebut adalah bilangan prima, maka angka tersebut ditambahkan ke dalam list lst. Lalu print semua hasil prima nya dalam bentuk list

Latihan 4

Lakukan perhitungan luas segitiga (A) menggunakan pendekatan Heron's formula. Sebelum melakukan perhitungan, pastikan bahwa nilai ketiga sisi merupakan sisi-sisi segitiga. Untuk membuktikan bahwa ketiga sisi merupakan sisi-sisi segitiga, pastikan bahwa hasil penjumlahan 2 sisi segitiga lebih besar dibandingkan panjang sisi lainnya.

$$A = \sqrt{s(s-a)(s-b)(s-c)} \qquad s = \frac{a+b+c}{2}$$

Fungsi pengecekan segitiga

```
#Fungsi pengecekan segitiga
def cek_segitiga(a,b,c):
    if a + b > c:
        return "Merupakan segitiga"
    else:
        return "Bukan merupakan segitiga"

#Dengan inputan

sisi_kiri = float(input("Masukkan sisi kiri segitiga: "))
sisi_kanan = float(input("Masukkan sisi kanan segitiga: "))
sisi_bawah = float(input("Masukkan sisi bawah segitiga: "))

cek_segitiga(sisi_kiri, sisi_kanan, sisi_bawah)
```

```
Masukkan sisi kiri segitiga: 6
Masukkan sisi kanan segitiga: 6
Masukkan sisi bawah segitiga: 10

'Merupakan segitiga'
```

Membuat fungsi def yang berisi inputan 3 sisi segitiga lalu ada baris percabangan untuk mengecek segitiga atau bukan, yang dimana jika hasil penjumlahan 2 sisi segitiga lebih besar dibandingkan panjang sisi lainnya maka hasilnya merupakan segitiga dan jika tidak maka bukan termasuk segitiga.

Untuk ketiga segitiga disini menggunakan input satu per satu dari sisi kiri, kanan, dan sisi bawah segitiga

Fungsi luas segitiga dengan formula heron

```
#Fungsi untuk menghitung luas segitiga dengan formula Heron
import math

def luas_segitiga(a, b, c) -> float:
    s = (a + b + c) / 2
    luas = math.sqrt(s * (s - a) * (s - b) * (s - c))
    return luas

#Dengan input sisi sisi segitiga sebelumnya

hasil = luas_segitiga(sisi_kiri, sisi_kanan, sisi_bawah)
print(hasil)
```

```
16.583123951777
```

```
#Dengan input sisi sisi yang baru

luas_segitiga(4,3,5)
```

```
6.0
```

Membuat fungsi def yang berisi 3 parameter yaitu sisi a, b, dan c. Lalu berisi dua rumus, rumus pertama menghitung semi-perimeter (setengah dari keliling segitiga) dengan rumus s = (a + b + c) / 2.

Yang kedua menggunakan rumus Heron, menghitung luas segitiga dengan formula luas = math.sqrt(s * (s - a) * (s - b) * (s - c)). Dengan menggunakan formula Heron, fungsi ini menghitung luas segitiga dengan cara yang tepat, berdasarkan panjang sisi-sisi segitiga yang diberikan sebagai argumen fungsi.

Lalu print output hasil segitiga

Latihan 5

```
# Definisikan sebuah fungsi untuk menghitung rata-rata
def hitung_rata_rata(bilangan):
    total = sum(bilangan)
    rata_rata = total / len(bilangan)
    return rata_rata

# Masukkan sejumlah bilangan ke dalam sebuah list
bilangan = [10, 20, 30, 40, 50]

# Panggil fungsi hitung_rata_rata dengan list bilangan sebagai argumen
hasil_rata_rata = hitung_rata_rata(bilangan)

# Cetak hasilnya
print("Rata-rata dari bilangan tersebut adalah:", hasil_rata_rata)
```

```
Rata-rata dari bilangan tersebut adalah: 30.0
```

Penjelasan Singkat:

1. Definisikan sebuah fungsi bernama `hitung_rata_rata` yang menerima sebuah list bilangan sebagai argumen.
2. Di dalam fungsi ini, kami menghitung total dari semua bilangan dalam list menggunakan fungsi `sum()` dan kemudian membaginya dengan jumlah elemen dalam list untuk menghitung rata-ratanya.
3. Fungsi ini mengembalikan nilai rata-rata.
4. Kami membuat sebuah list bilangan sebagai contoh.
5. Kami memanggil fungsi `hitung_rata_rata` dengan list bilangan sebagai argumen.
6. Hasil rata-ratanya kemudian dicetak ke layar.

Program ini mengilustrasikan penggunaan fungsi sederhana dalam Python untuk menghitung rata-rata dari sejumlah bilangan.