



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71200581</b>
<b>Nama Lengkap</b>	<b>Dewangga Yuka Pratama</b>
<b>Minggu ke / Materi</b>	<b>04 / Modular Programming</b>

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2024

## BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

### Fungsi, Argumen & Parameter

Fungsi merupakan sebuah kumpulan beberapa perintah yang disatukan untuk suatu tujuan yang nantinya dapat digunakan secara berulang. Setiap fungsi memiliki kegunaan masing-masing tergantung dari tujuan fungsi tersebut dibuat, sebagai contoh adalah fungsi `print()` yang digunakan untuk menampilkan hasil output ataupun fungsi `type()` yang digunakan untuk mengetahui tipe dari objek pada suatu variabel.

```
5
6  angka = 12
7
8  print(type(angka))
```

Ada juga yang disebut sebagai block yaitu beberapa kode program yang dikelompokkan kedalam sebuah kode program yang lebih besar lagi.

```
2
3
4  def nama(name):
5      |
6      print("Hello, " + name + "!")
7      print("Bagaimana Kabarmu?")
8
9  # Calling the function
10 nama("Saya")
11 |
```

Maka dari itu modular merupakan program yang terdiri dari bagian-bagian kecil yang memiliki kegunaan khusus dan dapat digunakan ulang.

Sebagai berikut contoh dari fungsi:

```
2
3  #Definisikan fungsi pengurutan
4  def urut(angka):
5      |   return sorted(angka)
6
7  #Memanggil Fungsi
8  angka = [5, 2, 8, 1, 3]
9  pengurutan = urut(angka)
10 print("Urutan : ", pengurutan)
```

```
Urutan :  [1, 2, 3, 5, 8]
```

Kode tersebut berjalan dengan alur sebagai berikut:

1. Kode mendefinisikan sebuah fungsi bernama urut yang memiliki satu parameter angka.
2. Fungsi urut menggunakan fungsi sorted() untuk mengurutkan elemen-elemen dalam list angka.
3. Hasil pengurutan disimpan dalam variabel pengurutan.
4. Selanjutnya, kode mencetak hasil pengurutan dengan pesan "Urutan : " diikuti oleh nilai pengurutan.
5. Ketika kode dijalankan, list [5, 2, 8, 1, 3] akan diurutkan, dan hasilnya akan dicetak. Output yang diharapkan adalah:

## Return Value

Return value adalah nilai yang dikembalikan suatu fungsi setelah program selesai dijalankan/eksekusi, namun terdapat juga fungsi yang tidak mengembalikan fungsi yang dinamakan sebagai fungsi void. Dengan mengembalikan nilai dari suatu fungsi maka dapat digunakan untuk berbagai tujuan, misalnya menyimpan ke dalam variabel, menggunakan nilai untuk melakukan sebuah operasi, atau hanya untuk menampilkan nilainya saja.

```
4
5
6  def tambah(a, b):
7      return a + b
8
9  hasil = tambah(3, 4)
10 print(hasil) # Output: 7
```

Sebagai contoh lainnya dari penggunaan return :

```
2
3
4  def hitung_luas_persegi(sisi):
5      luas = sisi * sisi
6      return luas
7
8  sisi_persegi = 5
9  luas_persegi = hitung_luas_persegi(sisi_persegi)
10 print("Luas persegi dengan sisi", sisi_persegi, "adalah", luas_persegi)
11
```

Alurnya seperti berikut:

1. Membuat fungsi `hitung_luas_persegi` dengan rumus ' $\text{luas} = \text{sisi} * \text{sisi}$ '
2. Mengembalikan nilai luas
3. Membuat variabel untuk memanggil fungsi tersebut
4. Menggunakan return value dari fungsi untuk mengisi variabel luas persegi
5. Print hasilnya

## Optional Argument & Named Argument

Optional argument adalah parameter yang dapat diisi dengan nilai default jika tidak ada nilai yang diberikan saat memanggil fungsi atau metode.

```
1
2
3 def halo(name, salam="Halo"):
4     print(f"{salam}, {name}!")
5
6 halo("Budi") # Output: Hello, Budi
7 halo("Bob", "Hi") # Output: Hi, Bob
8
```

Disini 'salam' adalah optional argument dengan nilai default halo, tetapi masih bisa diganti dengan nilai lain saat memanggil fungsi halo. Sedangkan named argument adalah cara untuk memanggil fungsi dengan menyebutkan nama parameter beserta nilainya, sehingga urutan tidak perlu sesuai dengan definisi fungsi.

```
1
2
3 def halo(name, salam="Halo"):
4     print(f"{salam}, {name}!")
5
6 halo(salam='Hi', name='Budi') #Output = Hi, Alice
7
8
```

Dalam contoh ini, kita menyebutkan nama parameter saat memanggil fungsi halo, sehingga urutan argumennya tidak harus sesuai dengan definisi fungsi.

Jadi, optional argument memberi fleksibilitas dalam penggunaan nilai default, sementara named argument memungkinkan kita memanggil fungsi tanpa peduli dengan urutan parameter.

## AnonymousFunction(Lambda)

Fungsi anonim atau lambda dalam pemrograman adalah fungsi yang didefinisikan tanpa nama. Biasanya digunakan ketika kita memerlukan fungsi sederhana yang hanya akan digunakan sekali atau beberapa kali. Dalam Python, fungsi lambda didefinisikan menggunakan kata kunci lambda, diikuti oleh daftar argumen, kemudian diikuti oleh ekspresi tunggal yang merupakan nilai yang akan dikembalikan oleh fungsi tersebut.

```
# Mendefinisikan fungsi lambda untuk menghitung kuadrat dari sebuah angka
kuadrat = lambda x: x**2

# Menggunakan fungsi lambda
print(kuadrat(5)) # Output: 25
```

## BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

### SOAL 1

```
1 def cek_angka(n1, n2, n3):
2     """
3     Fungsi ini untuk menentukan apakah ketiga parameter memenuhi semua ketentuan:
4     • Ketiga parameter tersebut nilainya berbeda semua.
5     • Ada kemungkinan jika diambil dua parameter dan dijumlahkan hasilnya sama dengan
6       parameter lainnya (yang tersisa).
7
8     Fungsi ini akan menghasilkan nilai True jika semua ketentuan tersebut dipenuhi.
9     Jika tidak terpenuhi maka fungsi akan menghasilkan nilai False.
10    """
11    # Menentukan jika ketiga parameter tersebut tidak ada yang sama/nilainya berbeda-beda
12    syarat1 = (n1 != n2) & (n1 != n3) & (n2 != n3)
13
14    # Menentukan apakah jika ada 2 parameter yang dijumlahkan hasilnya adalah parameter yang lainnya
15    syarat2 = (n1 + n2 == n3) | (n1 + n3 == n2) | (n2 + n3 == n1)
16
17    return syarat1 & syarat2
18
19 # Contoh penggunaan
20 print(cek_angka(12, 5, 7))
21
```

```
Laporan/Modul4/Soal1.py"
True
```

Penjelasan:

Pada soal diminta untuk membuat fungsi dimana terdapat 3 parameter berupa angka yang nilainya berbeda dan terdapat kemungkinan jika 2 dari 3 nilai dari parameter tersebut dijumlahkan hasilnya adalah salah satu dari ketiga parameter. Jika ketentuan tersebut terpenuhi akan menghasilkan nilai TRUE dan sebaliknya akan menghasilkan nilai FALSE

1. Membuat fungsi cek\_angka() dengan 3 parameter n1,n2,n3
2. Membuat variabel syarat1 untuk mengecek jika n1 tidak sama dengan n2, n1 tidak sama dengan n3, dan n2 tidak sama dengan n3. Sehingga hasil dari ketiga variabel akan berbeda jika tidak akan menghasilkan false
3. Membuat variabel syarat2 untuk mengecek hasil jumlah dari tiap parameter apakah ada yang sama dengan parameter lainnya atau tidak
4. Lalu kembalikan nilai dari variabel syarat1 dan syarat2

## SOAL 2

```
1 def cek_digit_belakang(d1, d2, d3):
2     """
3     Fungsi untuk menentukan apakah minimal dua dari tiga parameter memiliki digit paling kanan yang sama.
4
5     Parameter:
6     d1: Bilangan pertama.
7     d2: Bilangan kedua.
8     d3: Bilangan ketiga.
9
10    Return:
11    True jika minimal dua dari tiga parameter memiliki digit paling kanan yang sama, False jika tidak.
12    """
13    # Mendapatkan digit paling kanan dari setiap bilangan
14    digit_bil1 = d1 % 10
15    digit_bil2 = d2 % 10
16    digit_bil3 = d3 % 10
17
```

```
18    # Mengecek apakah dua dari tiga digit sama
19    if digit_bil1 == digit_bil2:
20        return True
21    elif digit_bil1 == digit_bil3:
22        return True
23    elif digit_bil2 == digit_bil3:
24        return True
25    else:
26        return False
27
28    # Membaca input dari pengguna
29    d1 = int(input("Masukkan nilai pertama: "))
30    d2 = int(input("Masukkan nilai kedua: "))
31    d3 = int(input("Masukkan nilai ketiga: "))
32
33    # Menggunakan fungsi cek_digit_belakang untuk menguji input
34    output = cek_digit_belakang(d1, d2, d3)
35    print("Output yang diharapkan =", output)
36
```

```
Laporan/Modul4/Soal2.py"
Masukkan nilai pertama: 12
Masukkan nilai kedua: 22
Masukkan nilai ketiga: 10
Output yang diharapkan = True
```

Penjelasan:

Pada soal tersebut jika ada minimal dua dari tiga parameter yang memiliki digit paling kanan ada yang sama akan menghasilkan True dan sebaliknya.

1. Buat fungsi cek\_digit\_belakang() dengan parameter d1,d2,d3
2. Membuat fungsi untuk mendapat digit paling kanan dari parameter yang nanti diinputkan, yaitu dengan menggunakan operator % 10
3. Buat perintah perulangan sederhana untuk mengecek digit paling kanan dari tiap parameter
4. Membuat inputan untuk pengguna dalam menentukan nilai untuk parameter
5. Memanggil fungsi cek\_digit\_belakang setelah itu print untuk mengetahui hasilnya



### Soal 3

```
# Fungsi konversi Celcius ke Fahrenheit
celsius_to_fahrenheit = lambda C: (9/5) * C + 32

# Fungsi konversi Celcius ke Reamur
celsius_to_reamur = lambda C: 0.8 * C

def konversi_suhu():
    print("Pilih jenis konversi suhu:")
    print("1. Celcius ke Fahrenheit")
    print("2. Celcius ke Reamur")
    pilihan = int(input("Masukkan pilihan (1/2): "))

    if pilihan == 1:
        celsius = float(input("Masukkan suhu dalam Celcius: "))
        fahrenheit = celsius_to_fahrenheit(celsius)
        print(f"Input C = {celsius}. Output F = {fahrenheit}.")
    elif pilihan == 2:
        celsius = float(input("Masukkan suhu dalam Celcius: "))
        reamur = celsius_to_reamur(celsius)
        print(f"Input C = {celsius}. Output R = {reamur}.")
    else:
        print("Pilihan tidak valid. Silakan pilih 1 atau 2.")

# Memanggil fungsi konversi_suhu
konversi_suhu()
|
```

```
Laporan/Modul4/Soal3.py
Pilih jenis konversi suhu:
1. Celcius ke Fahrenheit
2. Celcius ke Reamur
Masukkan pilihan (1/2): 1
Masukkan suhu dalam Celcius: 80
Input C = 80.0. Output F = 176.0.
```

```
Laporan/Modul4/Soal3.py
Pilih jenis konversi suhu:
1. Celcius ke Fahrenheit
2. Celcius ke Reamur
Masukkan pilihan (1/2): 2
Masukkan suhu dalam Celcius: 80
Input C = 80.0. Output R = 64.0.
```

Penjelasan:

Soal diminta membuat fungsi konversi suhu dari celcius ke fahrenheit dan reamur

1. Membuat fungsi untuk rumus konversi celcius ke fahrenheit
2. Membuat fungsi untuk rumus konversi celcius ke reamur
3. Buat fungsi konversi\_suhu()
4. Disini saya membuat semacam tampilan menu untuk user supaya dapat memilih suhu apa yang mau dikonversi
5. Membuat perulangan sederhana dimana jika user memilih menu 1 akan mengkonversi celcius ke fahrenheit dan jika user memilih menu 2 akan mengkonversi celcius ke reamur
6. Memanggil fungsi konversi\_suhu