



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	<71220917>
Nama Lengkap	<Anggrayni Layuk Mangayun>
Minggu ke / Materi	06 / Struktur Kontrol Perulangan

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

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.

MATERI 1 Definisi Perulangan

Perulangan adalah adalah sesuatu yang diulang- ulang berkali-kali,sama hal nya dengan perulangan dalam code artinya code yang diulang berkali-kali dengan jumlah yang tertentu selama masih memenuhi sebuah kondisi. Selain itu dikatan perulangan jika terjadi proses perulangan yang dilakukan secara bertahap dan mempunyai langkah yang sama perulangan juga bisa diakses dalam bentuk data contohnya : list,tuple,queue,stack contoh code nya ada pada gambar dibawah.

```
tes.py > ...
1  #list
2  buah=["Apel", "Mangga", "Jeruk", "Pisang"]
3  for i in buah:
4      print(i)
5
6  #tuple
7  numbers = (10, 20, 30, 40)
8  for num in numbers:
9      print(num)
10
11 #queue
12 from collections import deque
13 queue = deque(["A", "B", "C", "D"])
14 while queue:
15     print(queue.popleft()) # Mengakses dan menghapus elemen dari depan
```

Selain itu perulangan juga bisa menggunakan for while dan cara rekursif.berikut penjelasan mengenai perulangan for dan while.

MATERI 2 Bentuk Perulangan for

Perulangan for adalah perulangan yang terhitung atau perulangan yang perulangan sudah diketahui banyak nya.

```
#contoh bentuk perulangan for
```

```
nilai = 10 # jumlah perulangan
```

```
for i in range(nilai):  
    print(f"Perulangan ke-{i}")#tampilkan nilai i dalam string
```

pada penjelasan code perulangan for di atas bisa di lihat nilai=10 yang artinya ada 10kali perulangan, kemudian terdapat variable i yang bertujuan untuk membuat list dari menampung indeks dan range() dari 0-10. Kemudian ada str() untuk mengubah tipe data int ke string.

CONTOH lain perulangan for :Menghitung Jumlah Bilangan dari 1 sampai 50.

```
total = 0  
  
for i in range(1, 10):  
    total += i #menambah nilai I dalam total  
print("Jumlah:", total) # output jumlah:45  
#penjumlahan 1+2+3+4...+9
```

```
awal = 1  
akhir = 10  
# Perulangan for untuk menampilkan bilangan ganjil  
for bilangan in range(awal, akhir + 1):  
    if bilangan % 2 != 0: # Mengecek apakah bilangan ganjil (sisa bagi dengan 2  
        tidak nol)  
        print(bilangan) # Jika ganjil, cetak bilangan tersebut
```

Menampilkan Bilangan Ganjil dari 1 sampai 10 -> for bilangan in range(awal, akhir + 1): mengulang dari nilai 1 sampai 10, akhir +1 untuk memastikan 1 -10 ikut diperiksa

```
#Menampilkan Bilangan Genap dari 60 ke 3 (Mundur)  
for i in range(10, 1, -2):  
    print(i)
```

for i in range(60, 2, -3)-> range(start, stop, step): artinya start = 60 → Perulangan dimulai dari 60.
stop = 2 → Perulangan akan **berhenti sebelum** mencapai angka 2 (jadi 2 tidak dicetak). step = -3
→ Setiap iterasi, nilai i akan dikurangi 3.

```
#menampilkan teks berulang kali  
for _ in range(5):  
    print("selamat anda lulus")#output: selamat anda lulus (sebanyak 5kali)
```

contoh contoh perulangan for:

```
#membuat segitiga ke kiri  
def cetak_pola(n):  
    for i in range(n):  
        hasil = 1  
        for j in range(n - i):  
            hasil *= (j + 1)
```

```

        print(hasil, end=' ')
        for k in range(n - i, 0, -1):
            print(k, end=' ')
        print()

# Input dari pengguna
n = int(input("Masukkan nilai n: "))
cetak_pola(n)

```

perulangan membuat persegi:

```

#membuat persegi
def persegi(tinggi, lebar):

    n = tinggi*lebar
    for i in range(n):
        print(i+1, end=" ")
        if (i+1) % lebar == 0:
            print()
    persegi(5, 4)

```

contoh lain

```

# konversi huruf ke angka
def huruf_ke_angka(huruf):
    if 'A' <= huruf <= 'Z': # Cek apakah huruf kapital
        return ord(huruf) - ord('A') + 1
    elif 'a' <= huruf <= 'z': # Cek apakah huruf kecil
        return ord(huruf) - ord('a') + 1
    else:
        return "Bukan huruf A-Z"

# Contoh penggunaan
huruf_list = ["A", "C", "Z", "B", "b"]
for huruf in huruf_list:
    print(f"{huruf} = {huruf_ke_angka(huruf)}")

```

MATERI 3 Bentuk Perulangan While

Perulangan while adalah perulangan yang tidak terhitung atau belum diketahui pada perulangan nya. perulangan while akan berulang kali apabila nilai nya masih terpenuhi atau masih bernilai true jadi mengulang code sampai kondisi tidak lagi terpenuhi. Berikut contoh dari code while : menghitung jumlah angka genap

```
angka = 1 # angka mulai dari 1
jumlahg = 0 # untuk menghitung jumlah angka genap

while angka <= 20: # Perulangan while akan berjalan selama angka ≤ 10
    if angka % 2 == 0: # Mengecek apakah angka genap (sisanya bagi dengan 2 == 0)
        jumlahg += 1 # Jika angka genap, tambahkan 1 ke jumlah_genap
    angka += 1 # Menambah nilai angka agar perulangan tidak berjalan selamanya

print("Jumlah angka genap:", jumlahg) # Mencetak jumlah angka genap dari 1 hingga 20
```

Bentuk perulangan lainnya

```
#perulangan while menggunakan else
nomor = 0
while nomor < 3: # akan berulang selama nomor < 3
    print("mengulang ke-", nomor) # Mencetak nilai nomor saat ini
    nomor += 1 # Menambah nilai nomor agar perulangan tidak berjalan selamanya
else:
    print("perulangan selesai") # Akan dieksekusi setelah perulangan while berakhir
```

perulangan while dalam bentuk list

```
#menghitung rata rata dari sebuah list
angka = [1, 3, 7, 9, 11] # angka yang mau dihitung rata-ratanya
total = 0 # Variabel untuk menyimpan jumlah total angka dalam list
i = 0 # untuk perulangan while

while i < len(angka): # berulang selama i kurang dari panjang list angka
    total += angka[i] # Menambahkan angka pada indeks ke-i ke total
    i += 1 # Menambah nilai i agar bisa berhenti

rata_rata = total / len(angka) # Menghitung rata-rata dengan membagi total dengan jumlah dalam list
print('Rata-rata data adalah:', rata_rata) # output:6.2
```

len ()=menghitung jumlah elemen

MATERI 4 Penggunaan Break dan Continue

1.Continue adalah penggunaan untuk melanjutkan sebuah perulangan namun secara paksa,continue digunakan pada melewati sisa code untuk lanjut kecode berikutnya:

```
81 #contoh penggunaan continue
82 namaku = "anggrayni layuk"
83 for huruf in namaku:
84     if huruf in "anggi": # Jika huruf termasuk dalam string "anggi"
85         continue # Lewati iterasi ini (tidak mencetak huruf tersebut)
86     print(huruf, end=" ")
87
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
n r y n   l y u k
D:\laporan alpro\code_laporan_6>python -u "d:\laporan alpro\code_laporan_6\tes.py"
```

Continue berfungsi Jika kondisi `if` terpenuhi (huruf adalah a, n, g, i), maka huruf tersebut tidak akan dicetak dan akan lompat ke huruf berikutnya.

2.break adalah fungsi yang digunakan untuk menghentikan paksa sebuah perulangan. Contoh code nya

```
88 #contoh break pada perulangan for
89 for i in range(1,20):
90     print(i, ' x ', i, ' = ', i*i)
91     if i == 5:
92         break
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
1 x 1 = 1
2 x 2 = 4
3 x 3 = 9
4 x 4 = 16
5 x 5 = 25
```

Jadi pada gambar diatas `for i in range(1, 20)`: berjalan dari 1 samapai 19 dan berhenti pada `if i == 5`: karena ada break

```
#contoh break pada while
i=0
while i < 10:
    print(i)
```

```
if i == 6:
    break
i += 1
```

while i < 10: akan berjalan selama i kurang dari 10 namun karena ada if i == 6 break maka perulangan langsung berhenti di 6

MATERI 5 Konversi dari Bentuk for Menjadi Bentuk while

Pada perulangan for dan while dapat saling dikonversi karena memiliki elemen yang sama contoh code kedua perulangan:

```
104 #perulangan for
105 for i in range(10, 0, -1): # Dari 10 sampai 1 (mundur)
106     print(i)
107
108 #perulangan while
109 i = 10 # Nilai awal
110 while i > 0: # Kondisi akhir
111     print(i) # Cetak nilai i
112     i = i - 1 # Step: turunkan i sebesar 1
113
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
10
9
8
7
6
5
4
3
2
1
```

Fibonaci

```
#Menampilkan Deret Fibonacci
def fibonacci(n):
    a, b = 0, 1
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b
    print()

def main():
    n = 10 # Jumlah bilangan Fibonacci yang tetap
    print(f"Deret Fibonacci {n} bilangan pertama:")
```

```
    fibonacci(n)

main()
```

soal lain## menentukan

```
def fibonacci(bil):
    fibo_minus_satu = 0
    fibo_minus_dua = 0
    fibo_hasil = 0
    for x in range(1, bil + 1):
        fibo_minus_dua = fibo_minus_satu
        fibo_minus_satu = fibo_hasil
        if x <= 2:
            fibo_hasil = 1
        else:
            fibo_hasil = fibo_minus_satu + fibo_minus_dua
    return fibo_hasil

# Nilai n sudah ditentukan tanpa input pengguna
angka = 6 # Ubah sesuai kebutuhan
fibonacci_bil = fibonacci(angka)
print(f'Bilangan Fibonacci ke-{angka} adalah {fibonacci_bil}')
```

Deret Fibonacci hingga 100: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

```
def fibonacci_hingga_batas(batas):
    fib_minus_satu = 0
    fib_minus_dua = 1
    fib_hasil = 0
    deret = []

    while fib_hasil <= batas:
        fib_minus_dua = fib_minus_satu
        fib_minus_satu = fib_hasil
        if len(deret) < 2:
            fib_hasil = 1
        else:
            fib_hasil = fib_minus_satu + fib_minus_dua

        if fib_hasil <= batas:
            deret += [fib_hasil]
    return deret

# Batas nilai sudah ditentukan
batas = 200
```



```
hasil_fibonacci = fibonacci_hingga_batas(batas)
# Menampilkan hasil dengan output memanjang ke bawah
print(f"Deret Fibonacci hingga {batas}:")
for angka in hasil_fibonacci:
    print(angka)
```

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

Latihan 6.1 Buatlah program yang menerapkan perhitungan perkalian dengan menggunakan penjumlahan. Buatlah fungsi perkalian() dalam program tersebut! Berikut ini adalah beberapa contoh perhitungan yang diharapkan:

- $6 \times 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30$.
- $7 \times 10 = 10 + 10 + 10 + 10 + 10 + 10 + 10 = 70$.

```
def perkalian(a, b):# UNTUK MENDEFINISIKAN FUNGSI
    hasil = 0# menampung hasil
    for _ in range(a): #membuat nilai perulangan sebanyak a
        hasil += b #menambahkan b kedalam hasil

    print(f"{a} x {b} =", " + ".join([str(b)] * a), "=", hasil) #untuk
menampilkan format perkalian dan hasilnya
    #" + ".join([str(b)] * a) Mengubah angka b menjadi string, lalu mengulangnya
a kali dengan pemisah " + "
    return hasil

# pemanggilan
perkalian(6, 5)
perkalian(7, 10)
perkalian(3, 8)
perkalian(4, 12)
```

output

```
D:\laporan alpro\code_laporan_6>python -u "d:\laporan alpro\code_laporan_6\6.1.py"
6 x 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30
7 x 10 = 10 + 10 + 10 + 10 + 10 + 10 + 10 = 70
3 x 8 = 8 + 8 + 8 = 24
4 x 12 = 12 + 12 + 12 + 12 = 48
```

SOAL 2

Latihan 6.2 Buatlah program yang dapat menampilkan deret bilangan ganjil dari batas bawah dan batas atas yang diberikan oleh pengguna. Jika ternyata batas atas < batas bawah, berarti deret tersebut dimulai dari batas atas, sampai batas bawah (negatif range). Buatlah fungsi ganjil() dalam program tersebut! Berikut ini adalah contoh hasil yang diharapkan:

- bawah = 10, atas = 30. Karena bawah < atas, berarti dari kecil ke besar, maka hasilnya adalah: 11, 13, 15, 17, 19, 21, 23, 25, 27, 29.
- bawah = 97, atas = 82. Karena bawah > atas, berarti dari besar ke kecil, maka hasilnya adalah: 97, 95, 93, 91, 89, 87, 85, 83.

```
def ganjil(bawah, atas):
    if bawah < atas: # Jika batas bawah lebih kecil, iterasi dari bawah ke atas
        angka = range(bawah, atas + 1)
    else: # Jika batas bawah lebih besar, iterasi dari atas ke bawah
        angka = range(bawah, atas - 1, -1)

    # untuk melihat bilangan ganjil
    hasil = [i for i in angka if i % 2 != 0]

    # Cetak hasil
    print(f"hasilnya adalah: {bawah} ke {atas}: {'', '.join(map(str, hasil))}")

# memanggil output
ganjil(10, 30)
ganjil(97, 82)
```

ouputnya

```
D:\laporan alpro\code_laporan_6>python -u "d:\laporan alpro\code_laporan_6\6.2.py"
hasilnya adalah: 10 ke 30: 11, 13, 15, 17, 19, 21, 23, 25, 27, 29
hasilnya adalah: 97 ke 82: 97, 95, 93, 91, 89, 87, 85, 83
```

SOAL 3

Latihan 6.3 Buatlah sebuah program penghitung nilai Indeks Prestasi Semester (IPS). Input bagi program:

- Jumlah mata kuliah
- Nilai A, B, C, dan D untuk setiap mata kuliah mahasiswa. Diasumsikan sks setiap mata kuliah selalu 3. Kemudian bobot dari masing-masing nilai adalah: A=4, B=3, C=2, D=1.

Output program ialah hasil IPS yang didapatkan. Jalannya program seperti pada Gambar 6.7. Tips: Gunakan kontrol percabangan di dalam perulangan.

```

def hitung_ips():
    # Bobot nilai berdasarkan soal
    bobot_nilai = {'A': 4, 'B': 3, 'C': 2, 'D': 1}
    sks_per_mk = 3 # SKS tiap mata kuliah selalu 3

    # Meminta jumlah mata kuliah
    jumlah_mk = int(input("Masukkan jumlah mata kuliah: "))

    total_bobot = 0
    total_sks = jumlah_mk * sks_per_mk # Total SKS

    for i in range(1, jumlah_mk + 1):
        while True:
            nilai = input(f"Masukkan nilai untuk mata kuliah ke-{i} (A/B/C/D): ").upper()
            if nilai in bobot_nilai:
                total_bobot += bobot_nilai[nilai] * sks_per_mk
                break
            else:
                print("Nilai tidak valid! Masukkan A, B, C, atau D.")

    # Menghitung IPS
    ips = total_bobot / total_sks
    print(f"\nIndeks Prestasi Semester (IPS) Anda: {ips:.2f}")

# Pemanggilan fungsi
hitung_ips()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Masukkan jumlah mata kuliah: 6
Masukkan nilai untuk mata kuliah ke-1 (A/B/C/D): a
Masukkan nilai untuk mata kuliah ke-2 (A/B/C/D): b
Masukkan nilai untuk mata kuliah ke-3 (A/B/C/D): b
Masukkan nilai untuk mata kuliah ke-4 (A/B/C/D): b
Masukkan nilai untuk mata kuliah ke-5 (A/B/C/D): b
Masukkan nilai untuk mata kuliah ke-6 (A/B/C/D): c

```

```

Indeks Prestasi Semester (IPS) Anda: 3.00

```