



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230971
Nama Lengkap	James Marvin Santoso
Minggu ke / Materi	05 / Struktur Kontrol Perulangan

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

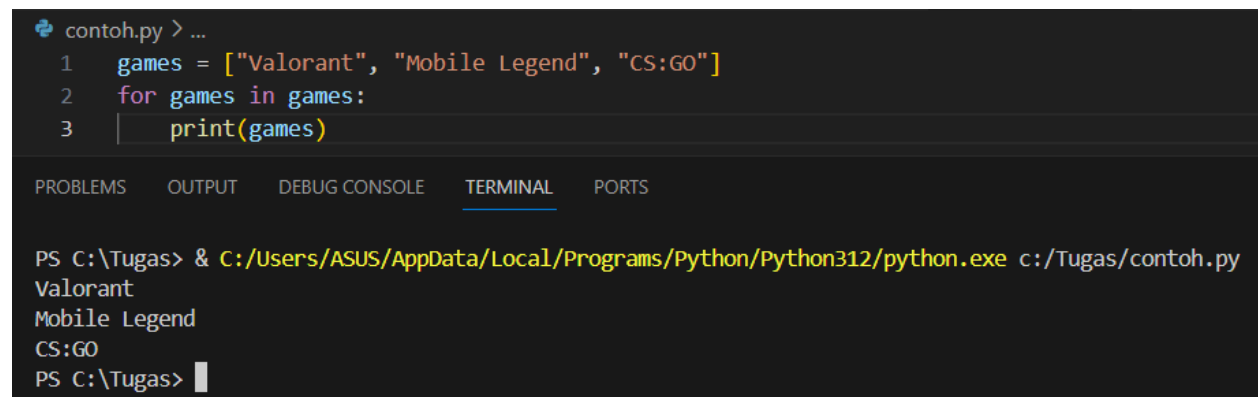
BAGIAN 1: MATERI MINGGU INI

Definisi Perulangan

Perulangan dalam Python adalah sebuah konsep pemrograman yang memungkinkan kita untuk menjalankan serangkaian instruksi atau blok kode secara berulang. Dua jenis perulangan yang umum digunakan dalam Python adalah for loop dan while loop.

- Loop

for loop digunakan untuk mengulangi blok kode untuk setiap elemen dalam sebuah urutan (seperti list, tuple, atau string) atau untuk setiap iterasi dalam rentang nilai tertentu. Contoh penggunaan for loop untuk mengulangi elemen dalam sebuah list:



```
contoh.py > ...
1  games = ["Valorant", "Mobile Legend", "CS:GO"]
2  for games in games:
3      print(games)
```

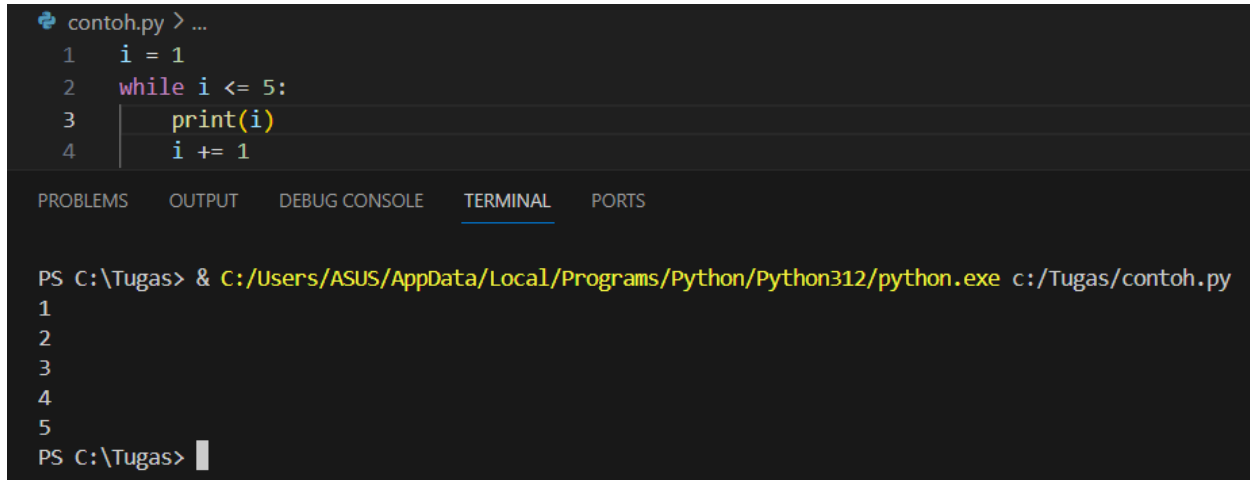
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
Valorant
Mobile Legend
CS:GO
PS C:\Tugas> |
```

Kode tersebut menggunakan `for` loop untuk mengiterasi melalui setiap elemen dalam list `games` dan mencetak setiap elemen tersebut ke layar. Variabel `game` digunakan untuk menyimpan nilai setiap elemen dalam setiap iterasi, dan kemudian nilai tersebut dicetak menggunakan fungsi `print()`. Hasilnya adalah mencetak setiap elemen dalam list `games` secara berurutan ke layar.

- While Loop

while loop digunakan untuk mengulangi blok kode selama kondisi yang diberikan benar. Perulangan akan terus berlanjut selama kondisi tersebut tetap benar. Contoh penggunaan while loop:



The screenshot shows a code editor with a Python file named 'contoh.py'. The code is as follows:

```
1 i = 1
2 while i <= 5:
3     print(i)
4     i += 1
```

Below the code editor is a terminal window. The command executed is:

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
```

The terminal output shows the numbers 1 through 5, each on a new line, indicating the successful execution of the while loop.

Kode tersebut menggunakan while loop untuk mencetak nilai *i* ke layar selama *i* kurang dari atau sama dengan 5. Pada awalnya, nilai *i* diatur sebagai 1. Selama nilai *i* kurang dari atau sama dengan 5, nilai *i* dicetak ke layar dan kemudian nilai *i* ditambah 1. Proses ini terus berlanjut sampai nilai *i* mencapai 6, di mana kondisi *i* <= 5 tidak lagi terpenuhi, sehingga while loop berhenti dan program keluar dari loop. Hasilnya adalah mencetak angka 1 hingga 5.

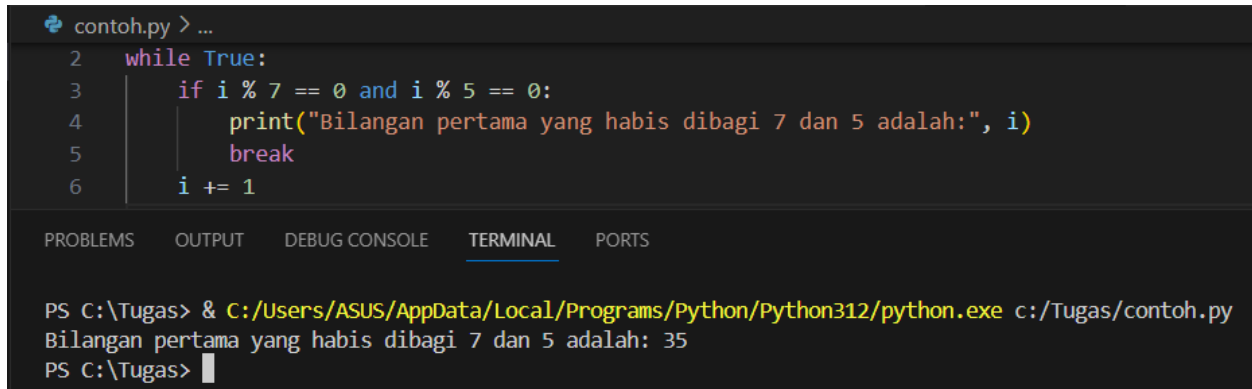
Dalam kedua contoh di atas, kita menggunakan loop untuk mencetak nilai dari suatu urutan atau iterasi sampai kondisi tertentu tercapai. Hal ini memungkinkan kita untuk menjalankan blok kode yang sama tanpa harus menuliskan kode tersebut secara berulang, sehingga membuat kode lebih efisien dan mudah dibaca.

Penggunaan Break dan Continue

- Break :

Break adalah sebuah pernyataan kontrol yang digunakan dalam Python untuk menghentikan eksekusi dari loop (baik itu for loop atau while loop) secara paksa, bahkan jika kondisi loop masih terpenuhi. Saat break dieksekusi, program akan keluar dari loop dan melanjutkan eksekusi dari pernyataan setelah loop.

Contoh penggunaan break dapat dilihat dalam sebuah while loop, sebagai berikut :



```
contoh.py > ...
2  while True:
3      if i % 7 == 0 and i % 5 == 0:
4          print("Bilangan pertama yang habis dibagi 7 dan 5 adalah:", i)
5          break
6      i += 1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
Bilangan pertama yang habis dibagi 7 dan 5 adalah: 35
PS C:\Tugas>
```

Dalam contoh ini, loop akan terus berjalan dengan nilai *i* yang bertambah satu setiap iterasi. Namun, ketika nilai *i* habis dibagi oleh 7 dan 5, pernyataan break akan dieksekusi, dan program keluar dari loop. Tanpa adanya break, loop akan berjalan tanpa henti karena kondisi while True selalu benar.

Penggunaan yang umum dari break adalah ketika kita ingin menghentikan eksekusi loop jika sebuah kondisi tertentu terpenuhi, dan kita tidak ingin melanjutkan iterasi lebih lanjut.

- Continue

Pernyataan continue dalam Python digunakan untuk melompati bagian dari loop saat ini dan melanjutkan ke iterasi berikutnya tanpa mengeksekusi kode yang ada di bawahnya dalam blok loop. Ini berguna ketika kita ingin mengabaikan beberapa iterasi dalam loop berdasarkan kondisi tertentu tanpa menghentikan loop secara keseluruhan.

Contoh penggunaan continue adalah dalam loop for, sebagai berikut :

```

1  for i in range(1, 6):
2      if i % 2 == 0:
3          continue
4      print(i)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
1
3
5
PS C:\Tugas>

```

Pada setiap iterasi, kode memeriksa apakah nilai i adalah bilangan genap. Jika ya, continue digunakan untuk melompati iterasi saat itu dan melanjutkan ke iterasi berikutnya. Sebaliknya, jika nilai i adalah bilangan ganjil, pernyataan print(i) akan dieksekusi, dan nilai ganjil tersebut akan dicetak ke layar.

Konversi dari Bentuk for Menjadi Bentuk while

Untuk mengkonversi bentuk for menjadi bentuk while, kita perlu mempertimbangkan bagaimana nilai loop dikelola dan bagaimana kondisi berhenti ditentukan. Berikut adalah contoh konversi dari loop for menjadi loop while.

- Bentuk for

```

1  for i in range(1, 6):
2      if i % 2 == 0:
3          continue
4      print(i)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
1
3
5
PS C:\Tugas>

```

- Konversi menjadi bentuk while

```
contoh.py > ...
1  i = 1
2  while i <= 5:
3      if i % 2 == 0:
4          i += 1
5          continue
6      print(i)
7      i += 1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/contoh.py
1
3
5
PS C:\Tugas> |
```

Dalam konversi ini, kita menggunakan variabel `i` sebagai pengganti iterator dalam loop `for`, dan kita mengatur kondisi berhenti loop `while` untuk saat `i` kurang dari atau sama dengan 5. Kita juga perlu memperbarui nilai `i` secara manual di dalam loop `while`, agar loop tidak menjadi loop tak terbatas.

BAGIAN 2: LATIHAN MANDIRI

SOAL 5.1

- Source Code :

```
Latihan 5,1.py > ...
1  def perkalian(a, b):
2      if b == 0:
3          return 0
4      elif b > 0:
5          return a + perkalian(a, b - 1)
6      else:
7          return -perkalian(a, -b)
8
9  print("6 x 5 =", perkalian(6, 5))
10 print("7 x 10 =", perkalian(7, 10))
```

- Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 5,1.py"
6 x 5 = 30
7 x 10 = 70
PS C:\Tugas>
```

- Penjelasan :

Kode ini adalah cara untuk mengalikan dua angka positif a dan b. Jika b sama dengan 0, hasilnya adalah 0. Jika b lebih besar dari 0, hasilnya adalah a ditambah hasil perkalian a dengan (b-1). Jika b negatif, hasilnya adalah negatif dari hasil perkalian a dengan -b.

SOAL 5.2

- Source Code :

```
Latihan 5,2.py > ...
1  def deret_ganjil(bawah, atas):
2      if bawah > atas:
3          bawah, atas = atas, bawah
4
5      if bawah % 2 == 0:
6          bawah += 1
7
8      return [bilangan for bilangan in range(bawah, atas + 1, 2)]
9
10 batas_bawah = int(input("Masukkan batas bawah: "))
11 batas_atas = int(input("Masukkan batas atas: "))
12
13 print("Deret bilangan ganjil dari", batas_bawah, "sampai", batas_atas, "adalah:")
14 print(deret_ganjil(batas_bawah, batas_atas))
```

- Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 5,2.py"
Masukkan batas bawah: 1
Masukkan batas atas: 10
Deret bilangan ganjil dari 1 sampai 10 adalah:
[1, 3, 5, 7, 9]
PS C:\Tugas> █
```

- Penjelasan :

Fungsi `deret_ganjil` menghasilkan deret bilangan ganjil antara dua bilangan, dengan memperhatikan urutan antara batas bawah dan batas atas, serta dimulai dari bilangan ganjil jika batas bawah genap.

SOAL 5.3

- Source Code :

```
Latihan 5,3.py > ...
1 def hitung_ips(jumlah_matkul):
2     total_sks = jumlah_matkul * 3
3     total_bobot = 0
4     for _ in range(jumlah_matkul):
5         nilai = input("Masukkan nilai (A, B, C, atau D): ")
6         if nilai.upper() == 'A':
7             total_bobot += 4 * 3
8         elif nilai.upper() == 'B':
9             total_bobot += 3 * 3
10        elif nilai.upper() == 'C':
11            total_bobot += 2 * 3
12        elif nilai.upper() == 'D':
13            total_bobot += 1 * 3
14        else:
15            print("Nilai yang dimasukkan tidak valid.")
16            return None
17    return total_bobot / total_sks
18
19 jumlah_matkul = int(input("Masukkan jumlah mata kuliah: "))
20
21 ips = hitung_ips(jumlah_matkul)
22 if ips is not None:
23     print("Indeks Prestasi Semester (IPS) Anda adalah:", ips)
```

- Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 5,3.py"
Masukkan jumlah mata kuliah: 6
Masukkan nilai (A, B, C, atau D): A
Masukkan nilai (A, B, C, atau D): B
Masukkan nilai (A, B, C, atau D): B
Masukkan nilai (A, B, C, atau D): D
Masukkan nilai (A, B, C, atau D): A
Masukkan nilai (A, B, C, atau D): A
Indeks Prestasi Semester (IPS) Anda adalah: 3.1666666666666665
PS C:\Tugas> 
```

- Penjelasan :

Ini adalah fungsi Python untuk menghitung Indeks Prestasi Semester (IPS) berdasarkan total SKS dari mata kuliah yang diambil, dengan mengumpulkan total bobot nilai dari setiap mata kuliah yang diinputkan, dan mengembalikan IPS dengan rumus total bobot dibagi total SKS.

- Link GitHub :

<https://github.com/Jamesmarvins/Tugas-Alpro-5.git>