



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71200581
Nama Lengkap	Dewangga Yuka Pratama
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 (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.

Perulangan

Struktur kontrol perulangan adalah sebuah cara atau fungsi dalam pemrograman yang digunakan untuk mengulang eksekusi sebuah blok kode selama kondisi-kondisi yang dibutuhkan terpenuhi. Ada beberapa jenis struktur kontrol perulangan yang umum digunakan, seperti:

1. Perulangan for: Digunakan untuk mengulang blok kode dengan jumlah iterasi yang sudah diketahui sebelumnya. Biasanya digunakan ketika kita sudah tahu berapa kali kita ingin mengulang suatu blok kode.
2. Perulangan while: Digunakan untuk mengulang blok kode selama kondisi tertentu terpenuhi. Biasanya digunakan ketika kita tidak tahu berapa kali kita ingin mengulang suatu blok kode, tetapi kita tahu kapan kita ingin berhenti.

Perulangan FOR

Perulangan "for" adalah jenis perulangan yang umum digunakan dalam pemrograman untuk mengulang sebuah blok kode dengan jumlah iterasi yang sudah diketahui sebelumnya. Sebagai contoh penggunaan perulangan for sebagai berikut :

```
3
4  urutan_angka = [1,2,3,4,5]
5  angka = 0
6
7  for i in urutan_angka :
8      angka += i
9
10 print('Jumlah : ', angka)
```

```
Jumlah : 15
```

Atau kita juga bisa menggunakan fungsi `range()` untuk membuat rentang nilai yang akan digunakan dalam perulangan `for`.

```
2
3  for i in range(10):
4      print("Perulangan ke-", i+1)
5
```

Perulangan WHILE

Perulangan `while` adalah struktur kontrol perulangan yang digunakan untuk mengulang blok kode selama kondisi-kondisi yang dibutuhkan terpenuhi. Berbeda dengan perulangan `for`, pada perulangan `while` umumnya digunakan disaat kita tidak tahu berapa kali kita ingin mengulang eksekusi blok kode tersebut. Sebagai contoh penggunaan `while` seperti berikut :

```
3
4  i = 1
5  while i <= 10:
6      print("Nomor urut:", i)
7      i += 1
8
```

```
Nomor urut: 1
Nomor urut: 2
Nomor urut: 3
Nomor urut: 4
Nomor urut: 5
Nomor urut: 6
Nomor urut: 7
Nomor urut: 8
Nomor urut: 9
Nomor urut: 10
```

Pada contoh di atas, `while i <= 10` adalah kondisi perulangan. Selama nilai `i` kurang dari atau sama dengan 10, blok kode di dalam perulangan akan terus dieksekusi. Setiap kali blok kode dieksekusi, nilai `i` akan ditambah 1. Perulangan akan berhenti ketika nilai `i` mencapai 11 karena kondisi `i <= 10` tidak lagi terpenuhi.

Penggunaan Break & Continue

Break dan continue adalah dua statement yang sering digunakan dalam struktur perulangan untuk mengontrol alur eksekusi perulangan. Break digunakan untuk menghentikan perulangan secara paksa jika suatu kondisi tertentu terpenuhi. Setelah break dieksekusi, program akan keluar dari perulangan dan melanjutkan eksekusi pernyataan setelah perulangan. Contoh penggunaan break:

```
3
4  i = 1
5  while i <= 10:
6      if i == 4:
7          break
8      print("Nomor urut:", i)
9      i += 1
10
```

```
Nomor urut: 1
Nomor urut: 2
Nomor urut: 3
```

Pada contoh di atas, perulangan while akan dihentikan ketika nilai i sama dengan 4 karena statement break dieksekusi.

Sedangkan continue digunakan untuk melanjutkan ke iterasi berikutnya dalam perulangan tanpa mengeksekusi bagian dari blok kode di bawahnya. Ketika continue dieksekusi, program akan langsung melanjutkan ke iterasi berikutnya tanpa mengeksekusi kode di bawahnya. Contoh penggunaan continue:

```
3
4  i = 0
5  while i < 5:
6      i += 1
7      if i == 3:
8          continue
9      print(i)
10
11
```

```
1
2
4
5
```

Pada contoh di atas, saat i sama dengan 3, pernyataan continue dieksekusi. Sehingga, perintah print(i) tidak dieksekusi pada iterasi tersebut, dan program langsung melanjutkan ke iterasi berikutnya.

Konversi dari Bentuk for Menjadi Bentuk while

Konversi dari bentuk for menjadi bentuk while adalah proses mengubah struktur perulangan for menjadi struktur perulangan while. Hal ini dapat dilakukan dengan memperhatikan cara kerja masing-masing struktur perulangan. Contohnya sebagai berikut

```
3  
4   for i in range(5):  
5       print(i)  
6  
7  
8
```

Menjadi

```
4   i = 0  
5   while i < 5:  
6       print(i)  
7       i += 1  
8
```

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 perkalian(x, y):
2     hasil = 0
3     for i in range(y):
4         hasil += x
5     return hasil
6
7 # Contoh penggunaan fungsi perkalian()
8 hitung = [(6, 5), (7, 10)]
9 for x, y in hitung:
10     hasil = perkalian(x, y)
11     print(f"{x}x{y}=", end="")
12     for i in range(y):
13         if i != y - 1:
14             print(f"{x}+", end="")
15         else:
16             print(f"{x}={hasil}.")
17
```

```
6x5=6+6+6+6+6=30.
7x10=7+7+7+7+7+7+7+7+7+7=70.
```

Penjelasan :

1. Buat fungsi def perkalian() dengan parameter x dan y
2. Variabel hasil untuk menyimpan hasil sementara dari x dan y
3. for i in range(y) Perulangan yang akan berjalan sebanyak y kali, yang mana y adalah parameter kedua dalam perkalian.
4. Hasil += x untuk menambahkan nilai kedalam variabel hasil
5. Mengembalikan nilai hasil
6. Variabel hitung merupakan perhitungan yang akan dilakukan
7. for x, y in hitung adalah perulangan for yang akan mengambil setiap pasangan nilai x dan y dari hitung.
8. Hasil = perkalian() Memanggil fungsi perkalian
9. print(f"{x}x{y}=", end="") Mencetak format string seperti di contoh
10. Melakukan perulangan untuk mengeprint/cetak hasil output dari perkalian
11. if i != y - 1 merupakan kondisi untuk menentukan apakah kita berada di iterasi terakhir dari perulangan atau tidak.
12. print(f"{x}+", end="") terjadi jika bukan iterasi terakhir, kita mencetak nilai x diikuti dengan tanda tambah +.
13. print(f"{x}={hasil}.") Mencetak hasil perkalian dengan format x=y.

SOAL 2

```
1 def ganjil(bts_bawah, bts_atas):
2     if bts_bawah < bts_atas:
3         deret_bil_ganjil = [i for i in range(bts_bawah, bts_atas+1) if i % 2 != 0]
4     else:
5         deret_bil_ganjil = [i for i in range(bts_bawah, bts_atas-1, -1) if i % 2 != 0]
6     return deret_bil_ganjil
7
8 bts_bawah = 10
9 bts_atas = 30
10
11 hasil = ganjil(bts_bawah, bts_atas)
12 if bts_bawah < bts_atas:
13     print("bawah =", bts_bawah, ", atas =", bts_atas, ". Karena bawah < atas, berarti dari", bts_bawah, "ke", bts_atas)
14 else:
15     print("bawah =", bts_bawah, ", atas =", bts_atas, ". Karena bawah > atas, berarti dari", bts_atas, "ke", bts_bawah)
16
17 print("adalah:", end=" ")
18 for i in range(len(hasil)):
19     if i == len(hasil) - 1:
20         print(hasil[i], end="")
21     else:
22         print(hasil[i], end=" ")
```

```
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
```

Penjelasan :

1. Membuat fungsi ganjil() dengan parameter bts_bawah dan bts_atas
2. Menggunakan fungsi if untuk melakukan statement jika nanti parameter bts_bawah lebih kecil dari bts_atas maka maka bilangan ganjil akan dihasilkan dari bts_bawah hingga bts_atas dan sebaliknya.
3. If $i \% 2$ digunakan untuk mengecek apakah bilangan tersebut ganjil atau tidak atau operator modulus.
4. Mengembalikan nilai deret_bil_ganjil
5. Sebutkan angka/nilai yang akan dieksekusi pada variabel bts_bawah dan bts_atas
6. Panggil fungsi ganjil
7. Kembali menggunakan if statement untuk menentukan bilangan mana yang akan diprint jika bts_bawah lebih kecil daripada bts_atas ataupun sebaliknya.
8. Mencetak hasil dengan menggunakan perulangan for ditambah fungsi len, hal ini digunakan untuk menghitung panjang objek yang urut karena nilai dari hasil eksekusi program tadi disimpan sebagai list.