



Modul Praktikum

# Algoritma dan Pemrograman

Jan Wantoro, S.T., M.Eng.

Irma Yuliana, S.T., M.M., M.Eng.

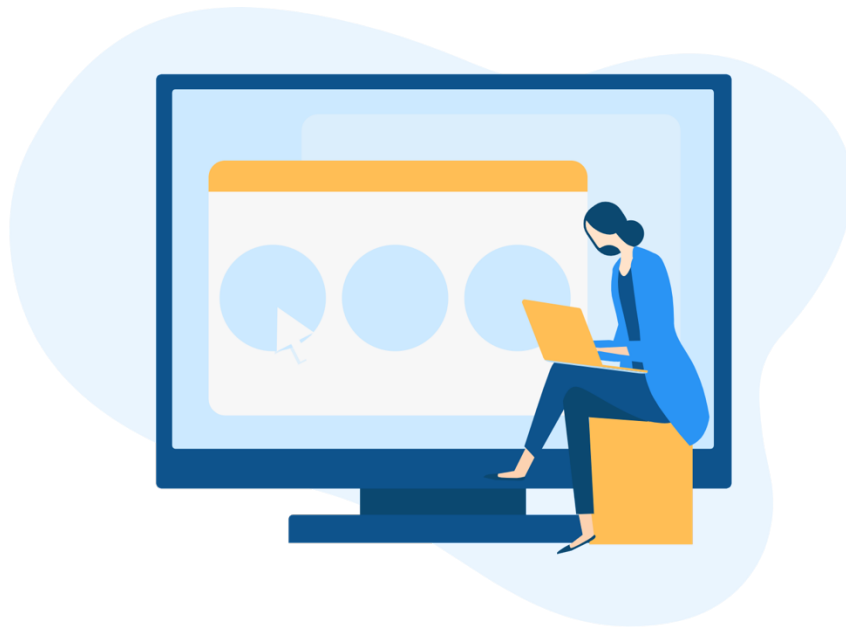
Arif Setiawan, S.Kom., M.Eng.

Program Studi Pendidikan Teknik Informatika

Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Surakarta

## BAB 5

### Percabangan dan Perulangan



#### 5.1 Tujuan

1. Dapat menjelaskan bentuk-bentuk kontrol percabangan dan perulangan pada program.
2. Dapat menjelaskan alur kontrol percabangan If, If..else dan variasinya.
3. Dapat menggunakan kontrol percabangan If, If..else dan variasinya.
4. Dapat menggunakan struktur kontrol perulangan sederhana

## 5.2 Pengantar Percabangan

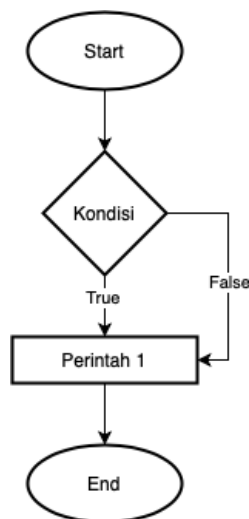
Pengambilan keputusan digunakan ketika kita ingin menjalankan kode jika ada suatu kondisi yang terpenuhi. Misalnya mencetak kata “Lulus” atau “Tidak Lulus” berdasarkan nilai ujian. Dalam bahasa pemrograman python, perintah untuk mengatur eksekusi kode dapat menggunakan beberapa cara seperti berikut.

### 5.2.1 IF

Perintah IF digunakan jika jalannya suatu kode didasarkan pada satu kondisi. Format penulisan IF adalah sebagai berikut

```
1. if kondisi:  
2.     perintah 1  
3.
```

Pada contoh format kode diatas, python akan mencari nilai dari kondisi. Jika kondisi bernilai True maka perintah 1 akan dijalankan. Begitupun sebaliknya, jika bernilai False maka perintah 1 tidak akan dijalankan. Perhatikan bahwa ada jarak indentasi pada penulisan perintah 1. Indentasi pada python merupakan tanda bahwa perintah 1 merupakan bagian dari IF. Gambar 7.1 berikut ini merupakan perintah IF dalam bentuk flowchart.



Gambar 7. 1 Flowchart Kondisi If

Perhatikan contoh kode berikut ini untuk penerapan perintah if

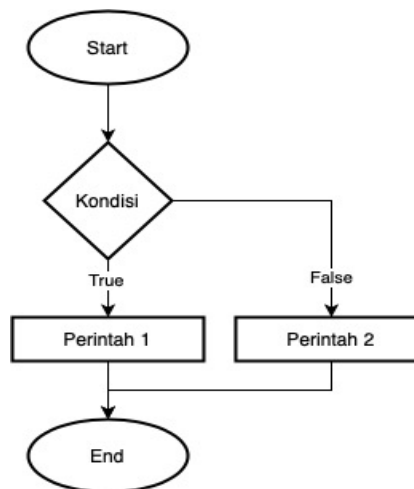
```
1. nilai = 90
2.
3. if nilai > 75:
4.     print('Kamu Lulus')
```

### 5.2.2 IF.. ELSE

Format perintah If..Else kita gunakan jika kita perlu mengeksekusi kode jika kondisi bernilai salah. Format dari perintah if..else adalah sebagai berikut

```
1. if kondisi:
2.     Perintah 1
3. else:
4.     Perintah 2
```

Pada contoh format diatas Perintah 1 akan dijalankan jika kondisi bernilai True sedangkan Perintah 2 akan dijalankan jika kondisi bernilai False. Gambar 7.2 berikut ini merupakan flowchart dari perintah if..else



Gambar 7. 2 Flowchart Kondisi If..Else

Perhatikan contoh kode berikut ini untuk penerapan perintah if..else

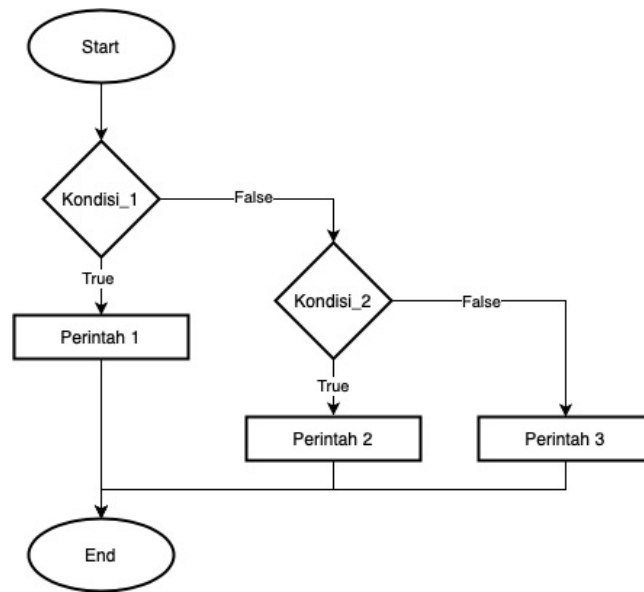
```
1. nilai = 60
2.
3. if nilai > 75:
4.     print('Kamu Lulus')
5. else:
6.     print('Kamu Mengulang')
```

### 5.2.3 IF..ELIF..ELSE

Sekarang bagaimana untuk program yang memerlukan seleksi lebih dari 2 kondisi. Maka kita menggunakan perintah yang ketiga yaitu if..elif..else. Format kodenya adalah sebagai berikut.

```
1. if kondisi_1:
2.     Perintah 1
3. elif kondisi_2:
4.     Perintah 2
5. else:
6.     Perintah 3
```

Pada contoh kode diatas Perintah 1 dan Perintah 2 hanya dijalankan jika memenuhi masing-masing kondisi. Sedangkan perintah 3 akan dijalankan jika tidak memenuhi kondisi\_1 dan kondisi\_2. Gambar 7.3 berikut merupakan flowchart dari perintah if..elif..else.



Gambar 7. 3 Flowchart kondisi if elif else

Perhatikan contoh kode berikut ini untuk penerapan perintah if..elif..else

```

1. nilai = 80
2.
3. if nilai > 90:
4.     print('Kamu Lulus Beasiswa Penuh')
5. elif nilai >=80 :
6.     print('Kamu Mendapat Beasiswa 1')
7. elif nilai >=70 :
8.     print('Kamu Mendapat Beasiswa 2')
9. else:
10.    print('Tidak Mendapat Beasiswa')
  
```

#### 5.2.4 IF Bersarang

Jika program kita sudah memerlukan seleksi kondisi yang lebih detail maka kita dapat membuat perintah If di dalam perintah if lainnya. Kondisi ini dinamakan if bersarang.

Perhatikan contoh kode untuk menentukan apakah angka yang diinput oleh user bernilai positif atau negatif.

```
1. angka = float(input('Masukkan angka :'))
2.
3. if angka > 0:
4.     if angka == 0:
5.         print('Angka nol')
6.     else:
7.         print('Angka Positif')
8. else:
9.     print('Angka Negatif')
10.
```

#### 5.2.5 Operator Logika pada IF

Selain IF bersarang kita dapat menggunakan operator logika AND dan OR untuk kondisi memerlukan seleksi lebih dari dua. Logika AND akan bernilai TRUE jika kedua kondisi bernilai benar. Sedangkan logika OR akan bernilai TRUE jika salah satu atau kedua kondisi bernilai benar. Perhatikan contoh penggunaan operator logika pada kode berikut ini

```
1. nilai = 60
2.
3. if nilai > 90:
4.     print('Nilai A')
5. if nilai >= 75 and nilai <= 89:
6.     print('Nilai B')
7. if nilai >= 60 and nilai <= 74:
8.     print('nilai C')
9. if nilai >= 45 and nilai <= 59:
10.    print('nilai D')
```

### 5.3 Pengantar Perulangan

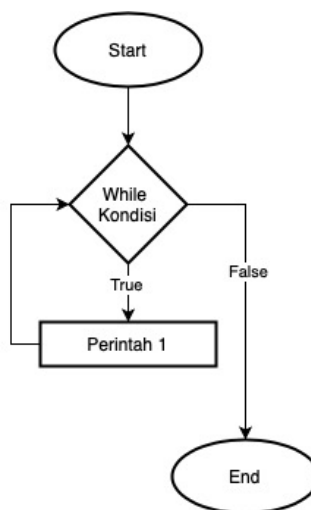
Pada bab sebelumnya kita sudah belajar membuat program yang dieksekusi secara runtut dari baris pertama ke baris selanjutnya. Pada bagian ini kita mencoba untuk mengatur alur program agar dapat menjalankan suatu kode lebih dari satu kali. Metode ini dinamakan dengan perulangan.

#### 5.3.1 Perulangan dengan While

Perulangan menggunakan while mirip dengan perintah if yang akan mengeksekusi kode didalamnya jika pengecekan kondisi bernilai True. Sedangkan pada while, kode didalamnya akan dijalankan terus berulang-ulang selama kondisinya bernilai benar. Sehingga agar kode tersebut tidak berjalan secara terus menerus, maka harus ada kode yang membuat kondisi di dalam while menjadi false. Format dari penulisan while adalah sebagai berikut

1. `while` kondisi:
2.     Perintah1

Pada contoh format diatas, Perintah1 akan diulang terus menerus hingga kondisi menjadi bernilai False. Gambar 8.1 berikut ini merupakan perintah while dalam bentuk flowchart.



Gambar 8. 1 Flowchart Perintah While



Perhatikan contoh kode berikut untuk penerapan while.

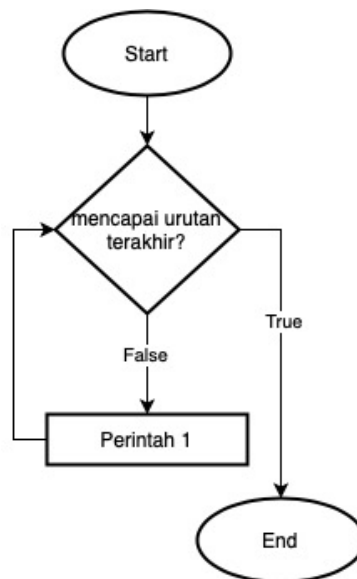
```
1. angka = 0
2. while angka < 10:
3.     print(angka, end=' ')
4.     angka = angka + 1
```

### 5.3.2 Perulangan dengan For

Perulangan for pada prinsipnya merupakan iterasi pada sebuah urutan atau rangkaian. Iterasi pada urutan ini disebut dengan traversal. Format penulisan for adalah sebagai berikut.

```
1. for item in nama_urutan:
2.     Perintah1
```

Pada contoh diatas, Perintah1 akan terus dijalankan hingga iterasi pada for mencapai urutan terakhir. Gambar 8.2 berikut ini merupakan flowchart dari perintah for.



Gambar 8. 2 Flowchart perintah for

Perhatikan contoh kode berikut untuk penerapan for

```
1. for i in range(5):
```

```
2.     print(i)
```

pada kode diatas kita menggunakan fungsi baru yaitu **range()**. Fungsi ini digunakan untuk menghasilkan suatu angka. Maka jika ada fungsi **range(5)** maka dia akan menghasilkan angka dari 0 sampai 5.

### 5.3.3 Perulangan dengan else

Kita juga dapat menggunakan bantuan else pada perulangan. Kode pada bagian else akan dijalankan jika urutan yang digunakan pada perulangan sudah habis. Perhatikan contoh kode berikut ini untuk perulangan while

```
1. counter = 0
2. while counter <= 3:
3.     print (counter)
4.     counter = counter + 1
5. else:
6.     print('counter sampai batas ')
```

Output dari kode diatas adalah

```
0
1
2
3
counter sampai batas
```

Sedangkan contoh berikut ini untuk perulangan for

```
1. angka = [1,2,3]
2.
3. for i in angka:
4.     print(i)
5. else:
6.     print('Angka sudah habis')
```

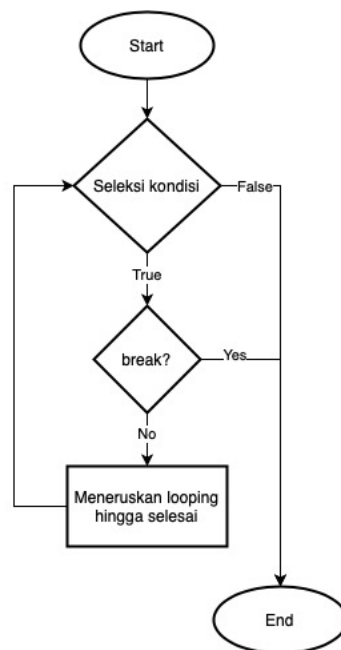
Output dari kode di atas adalah

```
1
2
3
Angka sudah habis
```

#### 5.3.4 Break dan Continue

Pada umumnya perulangan akan terus dilakukan hingga kondisi menemui false atau urutan sudah habis. Namun kita dapat menghentikan proses perulangan di tengah jalan menggunakan perintah break dan continue

Perintah break digunakan untuk menghentikan proses perulangan dan melanjutkan eksekusi kode diluar perulangan. Gambar 8.3 berikut ini merupakan flowchart dari break



Gambar 8. 3 Flowchart break

Contoh break dapat dilihat pada kode berikut ini

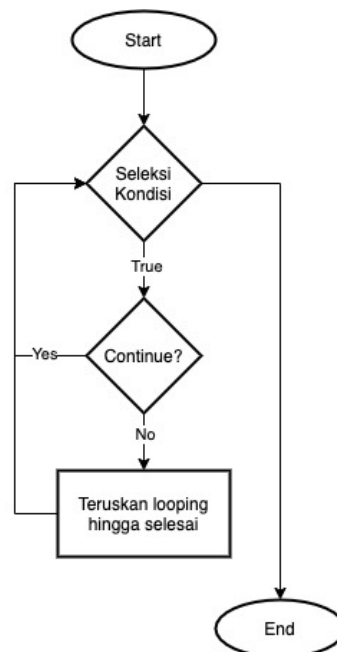
```
1. for i in range(10):
2.     if i == 5:
```

```
3.         break
4.     print(i, end=' ')
```

perulangan pada kode diatas seharusnya akan dilakukan dari 0 hingga 10 namun pada baris ke dua kita cek jika perulangan sampai ke 5 maka akan dilakukan break atau keluar dari perulangan. Sehingga output dari kode diatas adalah

```
0 1 2 3 4
```

Berbeda dengan break, perintah continue tidak menghentikan perulangan. Tapi digunakan untuk melewati kode pada iterasi tersebut kemudian melanjutkan perulangan ke iterasi selanjutnya. Gambar 8.4 berikut ini merupakan flowchart dari continue



Gambar 8. 4 Flowchart continue

Contoh continue dapat dilihat pada kode berikut ini

```
1. for i in range(10):
2.     if i == 5:
3.         continue
4.     print(i, end=' ')
```

perulangan pada kode diatas akan menghasilkan output

```
0 1 2 3 4 6 7 8 9
```

Dapat dilihat dari output diatas tidak ada angka 5 karena perintah continue akan melewati kode pada iterasi tersebut. Pada contoh diatas iterasi yang dilewati adalah jika i sama dengan 5.

## 5.4 Kegiatan Praktikum

### 5.4.1 Kegiatan Praktikum 1 : Bekerja dengan IF ELSE

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. angka = 5
2. if angka >= 0:
3.     print("Angka Positif atau Nol")
4. else:
5.     print("Angka Negatif")
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 1 ini

### 5.4.2 Kegiatan Praktikum 2 : Bekerja dengan IF Bersarang

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. a, b, c = input('Masukkan 3 angka tanpa spasi: ')
2. if a > b:
3.     if b > c:
4.         maks = a
5.     else:
6.         if c > a:
7.             maks = c
8. elif a > c:
9.     maks = b
10. elif c > b:
11.     maks = c
```

```
12.  
13. print(maks)
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 2 ini

### 5.4.3 Kegiatan Praktikum 3 : Perulangan While

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. counter = 0  
2. total = 0  
3.  
4. jumlah_barang = int(input('Masukkan Jumlah Barang : '))  
5. while counter < jumlah_barang:  
6.     harga = int(input('Masukkan harga barang ke-  
       {}: '.format(counter+1)))  
7.     jumlah = int(input('Masukkan jumlah barang ke-  
       {}: '.format(counter+1)))  
8.     total = total + (harga*jumlah)  
9.     counter = counter+1  
10.  
11. print('Total Harga Barang :',total)
```

3. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 3 ini

### 5.4.4 Kegiatan Praktikum 4 : Perulangan For

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. for i in range (1,10):  
2.     print(i, end=' ')  
3.  
4. print(' ')  
5. for j in range(10,0,-1):  
6.     print(j, end=' ')  
7.
```

```

8. print(' ')
9. for k in range(1,20,3):
10.     print(k, end=' ')
11.
12. print(' ')
13. for l in range(1, 20, 2):
14.     print(l, end=' ')

```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 4 ini

#### 5.4.5 Kegiatan Praktikum 3 : Mencari bilangan prima

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```

1. i = 2
2. while(i < 30):
3.     j = 2
4.     while(j <= (i/j)):
5.         if not(i%j): break
6.         j = j + 1
7.     if (j > i/j) : print (i, " adalah bilangan prima")
8.     i = i + 1

```

3. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 5 ini

### 5.5 Tugas

1. Buat sebuah program python untuk menentukan diskon tarif kereta api berdasarkan umur penumpang. Kondisi yang perlu di seleksi adalah
  - Usia 0 sampai 4 tahun mendapat diskon 100%
  - Usia 5 sampai 11 tahun mendapat diskon 50%
  - Usia 13 tahun keatas tidak mendapat diskon

Data yang perlu diinputkan adalah tahun kelahiran penumpang dan harga tiket kereta api

2. Buat sebuah program python untuk menampilkan lirik lagu 'Anak Bebek',  
dengan inputan jumlah anak bebek

```
Masukkan jumlah anak bebek : 5
Anak bebek turun 5, mati satu tinggal 4
Anak bebek turun 4, mati satu tinggal 3
Anak bebek turun 3, mati satu tinggal 2
Anak bebek turun 2, mati satu tinggal 1
Anak bebek turun 1, mati satu tinggal induknya
```



## DAFTAR PUSTAKA

- Hunt, J. (2019). *A Beginners Guide to Python 3 Programming*. In Springer
- Romano, Fabrizio. (2015). *Learning Python*. Packt Publishing.
- Swastika, W. (2019). *Pengantar Algoritma dan Penerapannya pada Python*. Ma Chung Press.
- Wadi, H. *Pemrograman Python untuk Mahasiswa dan Pelajar*. TR Publisher
- Python File Handling Tutorial: How to Create, Open, Read, Write, Append*. (n.d.). Retrieved September 7, 2020, from <https://www.softwaretestinghelp.com/python/python-file-reading-writing/>
- Learn Python Programming*. (n.d.). Retrieved September 7, 2020, from <https://www.programiz.com/python-programming>
- Tutorial Pemrograman Python*. (n.d.). Retrieved September 5 , 2020, from <https://www.petanikode.com/tutorial/python/>
- PY4E - Python for Everybody*. (n.d.). Retrieved September 2, 2020, from <https://www.py4e.com/>
- Google's Python Class | Python Education | Google Developers*. (n.d.). Retrieved September 9, 2020, from <https://developers.google.com/edu/python/>
- Learn Python the Hard Way*. (n.d.). Retrieved September 2, 2020, from <https://learnpythonthehardway.org/book/>
- Python Programming Tutorials*. (n.d.). Retrieved September 9, 2020, from <https://pythonprogramming.net/python-fundamental-tutorials/>

## **LAMPIRAN**

### **Laporan Sementara**

Laporan sementara dilakukan dengan langkah sebagai berikut:

1. Buatlah screenshot atau gambar dari hasil kegiatan yang dilakukan.
2. Ganti nama file dengan format: bab-kegiatan-no.gambar-nim.jpg Misalnya file adalah gambar ke 1 pada Bab 1, Kegiatan 1, dan nim anda adalah A40009001, maka nama filenya adalah **1-1-1-A40009001.jpg**
3. Simpan file tersebut kemudian tempatkan pada folder sesuai instruksi dosen/asisten praktikum.

### **Laporan Praktikum**

Laporan ditulis dalam kertas putih ukuran A4. Sedangkan urutan susunan laporan adalah sebagai berikut:

1. Cover depan: Berwarna sama dengan cover modul praktikum
2. Halaman Cover: Contoh dapat di-download di <http://bit.ly/lap-alpro>
3. Kata Pengantar
4. Daftar isi
5. Laporan tiap modul (1-10) sesuai dengan format terlampir
6. Penulis: berisi biodata penulis (disertai foto), pesan dan kesan, kritik dan saran demi kemajuan praktikum berikutnya.

## Format Laporan Tiap Bab

Matakuliah : <b>Algoritma dan Pemrograman</b>	<b>Acc</b>
NIM :	
Nama :	
Tgl. Prakt.:	Tgl:

### **BAB I** **Judul**

#### 1. Dasar Teori

300 sampai dengan 350 kata

#### 2. Tujuan

#### 3. Analisa Hasil

##### 3.1. Kegiatan 1: ...

Tampilkan hasil praktikum berupa kode yang dibuat atau hasil output, kemudian berikan analisisnya. Jika terdapat gambar, berikan juga nomor gambar.

##### 3.2. Kegiatan 2: ...

#### 4. Penyelesaian Tugas

Jika terdapat tugas yang dikerjakan, tuliskan disini langkah pengerjaan dan hasilnya.

#### 5. Kesimpulan

Berikan kesimpulan yang didapatkan setelah anda menyelesaikan praktikum