

**LAPORAN RESMI**  
**MODUL 2**  
**PENYELEKSIAN KONDISI**  
**ALGORITMA PEMROGRAMAN**



<b>NAMA</b>	<b>: IKTIAR RAMADANI</b>
<b>N.R.P</b>	<b>: 230441100053</b>
<b>DOSEN</b>	<b>: MOHAMMAD SYARIEF, ST., M.Cs</b>
<b>ASISTEN</b>	<b>: KUKUH COKRO WIBOWO</b>
<b>TGL PRAKTIKUM</b>	<b>: 29 SEPTEMBER 2023</b>

Disetujui : .. ..... 2023  
Asisten

**KUKUH COKRO WIBOWO**  
**21.04.411.00102**



**LABORATORIUM BISNIS INTELIJEN SISTEM**  
**PRODI SISTEM INFORMASI**  
**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Seleksi kondisi atau yang biasa disebut juga percabangan adalah proses menentukan apa yang akan dilakukan berdasarkan terpenuhi atau tidaknya suatu kondisi dengan banyak kemungkinan terjadi. Dengan seleksi kondisi, kita bisa memastikan bahwa program menjalankan suatu perintah tertentu jika memenuhi kondisi yang kita inginkan dan akan melanjutkan ke kondisi berikutnya apabila kondisi pertama tidak terpenuhi. Pada umumnya dalam membuat program, selalu ada seleksi dimana diperlukan pengecekan suatu kondisi untuk mengarahkan program agar berjalan sesuai keinginan dan dapat digunakan. Pada Python untuk melakukan suatu pengecekan kondisi, terdapat beberapa macam statemen, Antara lain adalah perintah `if`, perintah `if-else`, perintah `if-elif-else`, dan perintah `if bersarang`.

Latar belakang penyeleksian kondisi, dalam konteks pemrograman komputer, adalah konsep yang mendasari penggunaan pernyataan kondisional (seperti `"if"`, `"else"`, dan `"elif"`) untuk mengontrol alur eksekusi program berdasarkan kondisi tertentu. Ini adalah elemen dasar dari logika pemrograman yang memungkinkan program untuk membuat keputusan dan berperilaku sesuai dengan kondisi yang diberikan. Ada beberapa latar belakang ini terkait erat dengan konsep berikut seperti yang pertama logika kondisional yang pada dasar dari penyeleksian kondisi. Kedua adalah pernyataan `"if"` yang merupakan pernyataan dasar dalam penyeleksian kondisi benar. Ketiga adalah pernyataan `"else"` yang digunakan bersamaan dengan pernyataan `"if"` untuk menentukan tindakan yang akan diambil jika kondisi dalam pernyataan `"if"` adalah salah (`false`). Pernyataan keempat adalah `"elif"` (`else if`) yang digunakan untuk menguji kondisi tambahan setelah `"if"` dan sebelum `"else"` untuk mengambil tindakan alternatif jika kondisi dalam `"if"` salah (`false`). Kelima ada kontrol alur program yang menyeleksi kondisi adalah alat yang kuat untuk mengontrol alur eksekusi program. Dan keenam yaitu logika alur eksekusi adalah latar belakang ini juga mencakup bagaimana alur eksekusi program berubah berdasarkan hasil pengecekan kondisi.

Dengan latar belakang penyeleksian kondisi ini, pengembang dapat membuat program yang lebih dinamis, adaptif, dan mampu mengambil keputusan berdasarkan data yang ada. Ini adalah konsep kunci dalam pemrograman yang diterapkan dalam berbagai bahasa pemrograman dan digunakan dalam hampir setiap jenis aplikasi perangkat lunak.

## **1.2 Tujuan**

- Mampu memahami apa yang dimaksud dengan penyeleksian kondisi
- Mampu membedakan macam-macam penyeleksian kondisi
- Mampu menjelaskan fungsi penyeleksian kondisi
- Mampu membuat contoh dengan menggunakan program penyeleksian kondisi

## **BAB II**

### **DASAR TEORI**

#### **2.1 Pengertian Penyeleksian Kondisi**

Penyeleksian kondisi adalah konsep dasar dalam pemrograman yang digunakan untuk mengontrol alur eksekusi program berdasarkan kondisi-kondisi tertentu. Ini memungkinkan program untuk membuat keputusan tentang tindakan apa yang harus diambil berdasarkan apakah kondisi tertentu benar atau salah. Penyeleksian kondisi biasanya diimplementasikan dengan menggunakan pernyataan "if" dan "else" dalam banyak bahasa pemrograman.

Seleksi kondisi adalah suatu blok kode yang dieksekusi hanya ketika kriteria yang ditentukan terpenuhi. Teknik seleksi kondisi banyak digunakan untuk kontrol alur program. Dengan seleksi kondisi, kita bisa memastikan bahwa program menjalankan suatu perintah tertentu jika memenuhi kondisi yang kita inginkan. Penyeleksian kondisi adalah suatu proses dimana program akan memilih atau mengevaluasi suatu kondisi tertentu untuk kemudian melakukan aksi yang sesuai berdasarkan hasil evaluasi tersebut.

#### **2.2 Macam-macam Penyeleksian Kondisi**

- a) Statemen if : Digunakan untuk melakukan penyeleksian dimana jika kondisi bernilai benar maka program akan mengeksekusi statemen dibawahnya.
- b) Statemen if-else : Digunakan untuk melakukan penyeleksian kondisi dimana jika kondisi bernilai benar maka program akan mengeksekusi statemen 1. Namun, jika nilai kondisi bernilai salah maka statemen 2 yang akan dieksekusi.
- c) Statemen if-else – elif : Digunakan untuk melakukan penyeleksian kondisi dimana kondisi yang diberikan lebih dari 1 kondisi atau memiliki beberapa kondisi. Jika kondisi pertama bernilai benar maka lakukan seleksi kondisi ke-dua dan seterusnya.

d) Kondisi bersarang : Suatu kondisi di dalam kondisi tertentu, Jika terdapat 2 cabang kondisi maka di dalam salah satu cabang kondisi tersebut dapat pula di isi suatu kondisi tertentu.

## BAB III TUGAS PENDAHULUAN

### 3.1 Soal

1. Jelaskan pengertian penyeleksian kondisi
2. Sebutkan macam-macam penyeleksian kondisi
3. Tuliskan contoh soal dengan menggunakan penyeleksian kondisi dan jelaskan script nya
4. Jelaskan fungsi dari penyeleksian kondisi

### 3.2 Jawaban

1. Penyeleksian kondisi adalah suatu proses yang digunakan untuk menentukan apa yang akan dilakukan dan tidaknya suatu kondisi di sebuah program dengan banyak cara atau kemungkinan yang akan terjadi.
2. a. Perintah if : penyeleksian dimana jika kondisi tersebut bernilai benar.  
b. Perintah if - else : penyeleksian dimana jika kondisi satu benar, maka kondisi yang satunya bernilai salah  
c. Perintah if - elif - else : penyeleksian dimana terdapat lebih dari 2 kondisi yang ditemukan  
d. Perintah if bertingkat : penyeleksian suatu kondisi jika terdapat 2 cabang dalam salah satu cabang kondisi.

### 3. Contoh soal

```
angka = int(input("Masukkan: "))  
  
if angka < 100:  
    print("puluhan")  
elif 100 > angka < 999:  
    print("ratusan")  
else:  
    print("negatif")
```

### Penjelasan:

Pada program diatas, user harus memasukkan sebuah angka, misalnya 250. Setelah itu, user melakukan run dan program akan menghitungnya. pada contoh diatas kita memasukkan angka 250. pada tahap pertama, program akan menggunakan perintah "if". Dikarenakan pada perintah "if" hanya mempunyai perintah  $< 100$ , maka program akan lanjut ke tahap "elif" yang terdapat perintah  $100 < \text{angka} < 999$ . karena angka 250 ada di antaranya, maka program akan menampilkan hasil tersebut.

4. fungsi dari penyeleksian kondisi adalah untuk mengatur atau mengarahkan program untuk mencari nilai yang benar, memungkinkan program menjalankan perintah apakah kondisi tersebut benar atau salah, memvalidasi input, menghindari kesalahan eksekusi dan mengatur program agar berjalan lancar.

## BAB IV

### IMPLEMENTASI

#### 4.1 Membuat Program Penyeleksian dan Menyatakan Umur

##### 4.1.1 Source Code

Buatlah Script Penyeleksian kondisi dengan menyatakan umur. Jika umur di atas 50 = Tua, Jika umur di atas 25 = Dewasa, dan umur diatas 17 Muda serta umur diatas 7 Anak-anak!

```
print("===== MENYELEKSI BERDASAR UMUR =====")
user = int(input("MASUKKAN UMUR ANDA : "))

if user >= 50 :
    print("Karena umur anda", user , "Maka anda dapat dikatakan Tua")
elif user >= 25 :
    print("Karena umur anda", user , "Maka anda dapat dikatakan Dewasa")
elif user >= 17 :
    print("Karena umur anda", user , "Maka anda dapat dikatakan Muda")
elif user >= 7 :
    print("Karena umur anda", user , "Maka anda dapat dikatakan Anak")
else :
    print("bukan salah satu dari jenis diatas")
    # if user >= 6 :
    #     print("Karena umur anda", user , "Maka anda dapat dikatakan Anak Prasekolah")
    # elif user >= 2 :
    #     print("Karena umur anda", user , "Maka anda dapat dikatakan Balita")
    # else :
    #     print("Karena umur anda", user , "Maka anda dapat dikatakan Bayi")
```

##### 4.1.2 Penjelasan

User disuruh untuk memasukkan umur mereka. Setelah memasukkan user kemudian melakukan run dan program akan menghitung umur mereka dan menentukan umur tersebut masuk ke dalam kategori apa.

##### 4.1.3 Hasil

```
===== MENYELEKSI BERDASAR UMUR =====
MASUKKAN UMUR ANDA : 19
Karena umur anda 19 Maka anda dapat dikatakan Muda
```

#### 4.2. Membuat Program Data Nilai Mahasiswa

##### 4.2.1. Source Code

Buatlah Script Penyeleksian kondisi secara dinamis sebuah program yang menentukan nilai mahasiswa dengan memasukkan nama, nilai tugas, nilai uts, nilai uas, dan nilai tugas akhir. Tampilkan rata-rata nilai dan tentukan nilai dengan ketentuan :

$$100 - 80 = A$$

$$80 - 70 = B$$

$$70 - 60 = C$$

$$60 - 40 = D$$

$$40 - 0 = E$$



```

print("MENENTUKAN NAMA, RATA-RATA, DAN POINT MAHASISWA")

print(" ")

nama = str(input("MASUKKAN NAMA ANDA          : "))
nilai_tugas = float(input("MASUKKAN NILAI TUGAS ANDA          : "))
nilai_uts = float(input("MASUKKAN NILAI UTS ANDA          : "))
nilai_uas = float(input("MASUKKAN NILAI UAS ANDA          : "))
nilai_tugas_akhir = float(input("MASUKKAN NILAI TUGAS AKHIR ANDA : "))

rata_rata = (nilai_tugas + nilai_uts + nilai_uas + nilai_tugas_akhir)/4

if not (0 <= nilai_tugas <= 100):
    print("Nilai salah")
elif not (0 <= nilai_uts <= 100):
    print("Nilai salah")
elif not (0 <= nilai_uas <= 100):
    print("Nilai salah")
elif not (0 <= nilai_tugas_akhir <= 100):
    print("Nilai salah")
else:
    rata_rata

if 80 <= rata_rata <= 100:
    print('nilai A')
elif 70 <= rata_rata < 80:
    print('nilai B')
elif 60 <= rata_rata < 70:
    print('nilai C')
elif 40 <= rata_rata < 60:
    print('nilai D')
elif 0 <= rata_rata < 40:
    print('nilai E')
else:
    print('Nilai Tidak Ditemukan')

```

#### 4.2.2. Penjelasan

User diminta untuk memasukkan nama mereka, setelah itu program akan meminta user untuk memasukkan nilai tugas, nilai UTS, nilai UAS, dan nilai tugas akhir. Setelah itu program akan menghitung rata-rata dari nilai tugas, nilai UTS, nilai UAS, dan nilai tugas akhir dengan menambahkan dan membaginya dengan angka 4. Setelah itu, program akan mencari kategori yang cocok dengan nilai yang sudah dihitung. Saat user melakukan run, akan muncul hasil berupa nama, rata-rata, dan point/nilai/kategori dari user.

#### 4.2.3 Hasil

```

MENENTUKAN NAMA, RATA-RATA, DAN POINT MAHASISWA

MASUKKAN NAMA ANDA          : iktiar
MASUKKAN NILAI TUGAS ANDA   : 85
MASUKKAN NILAI UTS ANDA    : 87
MASUKKAN NILAI UAS ANDA    : 86
MASUKKAN NILAI TUGAS AKHIR ANDA : 89
nilai A

```

### 4.3 Membuat Program Permainan Kertas Gunting Batu

#### 4.3.1. Source Code

Buat sebuah program berupa permainan (gunting/batu/kertas) yang dapat dimainkan oleh dua orang. Kedua pemain menginput pilihan masing-masing (batu/kertas/gunting). Kemudian, program melakukan perbandingan

untuk melihat siapa yang menang dan siapa yang kalah menggunakan seleksi kondisi!

```
print("==== PERMAINAN KERTAS GUNTING BATU ====")

print(" ")

usersatu= input("Pemain satu (kertas/gunting/batu):")
userdua= input("Pemain dua (kertas/gunting/batu):")

if usersatu == userdua :
    hasil = "hasil : Seri!!!"
elif usersatu == "kertas" and userdua == "gunting" :
    hasil = "Pemain dua menang"
elif usersatu == "gunting" and userdua == "batu" :
    hasil = "Pemain dua menang"
elif usersatu == "batu" and userdua == "kertas" :
    hasil = "Pemain dua menang"
elif usersatu == "gunting" and userdua == "kertas" :
    hasil = "Pemain satu menang"
elif usersatu == "batu" and userdua == "gunting" :
    hasil = "Pemain satu menang"
elif usersatu == "kertas" and userdua == "batu" :
    hasil = "Pemain satu menang"
else :
    hasil = "Kata salah"

print(" ")

print( hasil )
```

#### 4.3.2. Penjelasan

Pada program kali ini akan ada 2 pilihan untuk bermain yaitu pemain satu dan pemain dua. User satu diminta untuk memilih pilihannya, dengan memilih antara kertas, gunting, dan batu. Setelah itu user dua melakukan hal yang sama. Setelah diisi, program akan mencari jawaban yang menang dan akan menampilkan hasilnya.

#### 4.3.3. Hasil

```
==== PERMAINAN KERTAS GUNTING BATU ====

Pemain satu (kertas/gunting/batu):kertas
Pemain dua (kertas/gunting/batu):batu

Pemain satu menang
```

## **BAB V**

### **PENUTUP**

#### **5.1 Analisa**

Dari hasil praktikum, praktikan menganalisa bahwa dengan menggunakan penyeleksian kondisi, kita bisa menentukan dimanakah kondisi yang dicari sesuai dengan nilai yang telah disiapkan. Dengan adanya penyeleksian kondisi diatas, kami menjadi lebih baik dalam melakukan pemrograman, dikarenakan dengan menggunakan penyeleksian kondisi, program yang kami buat menjadi lebih tertata dan dapat digunakan, serta mudah untuk menggunakannya. Pada kali ini, kita banyak belajar tentang macam-macam penyeleksian kondisi diantaranya adalah perintah “ if “ yang digunakan untuk nilai benar, perintah “ else “ untuk nilai salah dan perintah “ elif “ untuk menentukan beberapa kemungkinan nilai. Materi ini adalah dasar dari materi logika pemrograman yang berupa penyeleksian kondisi yang terdiri dari perintah “ if “ , “ else “ , “ elif “.

#### **5.2 Kesimpulan**

1. Kita jadi bisa mengerti dengan pentingnya penggunaan penyeleksian kondisi di sebuah program agar pprogram tersebut dapat berjalan sesuai dengan apa yang kita mau.
2. Penyeleksian kondisi adalah konsep dasar dalam pemrograman yang digunakan untuk mengontrol alur eksekusi program berdasarkan kondisi-kondisi tertentu.
3. Pada Python untuk melakukan suatu pengecekan kondisi, terdapat beberapa macam statemen, Antara lain adalah perintah if, perintah if-else, perintah if-elif-else, dan perintah if bersarang.
4. Kita jadi bisa membuat sebuah program yang menggunakan penyeleksian kondisi.
5. Mengetahui, memahami, dan mampu menjelaskan tentang apa itu penyeleksian kondisi, macam-macam bentuknya, fungsi dan kegunaanya.