

LAPORAN RESMI
MODUL II
PENYELEKSIAN KONDISI
ALGORITMA DAN DASAR PEMROGRAMAN



NAMA	: MUHAMMAD MAULIDAN HABIBI
N.R.P	: 250441100121
DOSEN	: SRI HERAWATI, S.Kom., M.Kom.
ASISTEN	: FAISAL DWI FIRMANSYAH
TGL PRAKTIKUM	: 03 OKTOBER 2025

Disetujui : 08 OKTOBER 2025
Asisten

FAISAL DWI FIRMANSYAH
24.04.411.00109



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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, kemampuan untuk membuat keputusan merupakan hal yang sangat penting agar program dapat berjalan sesuai dengan logika yang diinginkan. Setiap program tidak hanya mengeksekusi perintah secara berurutan, tetapi juga harus mampu menyesuaikan tindakan berdasarkan kondisi tertentu yang terjadi selama program dijalankan. Salah satu konsep dasar yang digunakan untuk mencapai hal tersebut adalah penyeleksian kondisi (conditional statement). Dengan menggunakan penyeleksian kondisi, program dapat memilih jalur eksekusi yang berbeda berdasarkan hasil evaluasi dari suatu pernyataan logika apakah bernilai benar (True) atau salah (False).

Dalam bahasa pemrograman Python, penyeleksian kondisi dilakukan menggunakan perintah `if`, `elif`, dan `else`. Struktur ini memungkinkan pengembang untuk mengatur alur program menjadi lebih fleksibel, efisien, dan interaktif sesuai kebutuhan logika yang diinginkan. Melalui praktikum ini, diharapkan mahasiswa atau peserta mampu memahami konsep dasar penyeleksian kondisi, cara penggunaannya dalam Python, serta mampu mengimplementasikannya dalam program sederhana yang melibatkan pengambilan keputusan.

1.2 Tujuan

- Mampu Memahami konsep dasar penyeleksian kondisi dalam bahasa pemrograman Python.
- Mengetahui fungsi dan perbedaan antara pernyataan `if`, `elif`, dan `else`.
- Mampu membuat program yang dapat mengambil keputusan berdasarkan hasil evaluasi suatu kondisi.
- Mampu mengimplementasikan penyeleksian kondisi pada kasus nyata untuk menghasilkan output yang sesuai dengan logika program.
- Mampu melatih kemampuan logika dalam menyusun alur program yang menggunakan percabangan.

BAB II

DASAR TEORI

2.1 Pengertian Penyeleksian Kondisi

Secara umum, dalam proses pembuatan program selalu dibutuhkan seleksi atau pengambilan keputusan, yaitu proses untuk memeriksa suatu kondisi agar program dapat berjalan sesuai dengan logika atau tujuan yang diinginkan, Dalam bahasa pemrograman Python, pengecekan kondisi ini dilakukan menggunakan tiga jenis pernyataan utama, yaitu:.

- A. Perintah if
- B. Perintah if-else
- C. Perintah if-elif-else
- D. Perintah If Bersarang

2.2 Perintah If

Pernyataan if digunakan untuk melakukan seleksi kondisi, di mana jika kondisi tersebut bernilai benar (True), maka program akan menjalankan perintah atau statemen yang berada di bawahnya, Dalam bahasa Python, penulisan antara kondisi dan blok perintah dipisahkan dengan tanda titik dua (:).

```
bahasa = "python"
if bahasa == "python":
    print("Hai,", bahasa)
```

2.3 Perintah If-else

Variabel adalah lokasi memori yang dicadangkan untuk menyimpan nilai- nil Statemen if – else digunakan untuk melakukan penyeleksian kondisi kondisi bernilai benar maka program akan mengeksekusi statemen 1. Namun,jika nilai kondisi bernilai salah maka statemen 2 yang akan dieksekusi.Penulisan variabel Python sendiri juga memiliki aturan tertentu, yaitu :

```
# Bentuk umum perintah if - else:
# if (kondisi):
#     statemen1
# else:
```

```
# statemen2
# Program pengecekan warna lampu lalu lintas
lampu = input("Masukkan warna lampu (merah/kuning/hijau): ")

if lampu == "hijau":
    print("Silakan jalan")
else:
    print("Berhenti dulu ya!")
```

2.4 Perintah if-elif-else

Pernyataan if – elif – else digunakan ketika program perlu melakukan pengecekan terhadap lebih dari satu kondisi. Struktur ini memungkinkan program untuk mengevaluasi beberapa kemungkinan secara berurutan.

Apabila kondisi pertama bernilai benar (True), maka perintah yang terkait akan dijalankan dan proses seleksi berhenti. Namun, jika kondisi pertama salah (False), program akan melanjutkan pengecekan ke kondisi berikutnya hingga menemukan kondisi yang benar, atau menjalankan bagian else jika tidak ada kondisi yang terpenuhi.

```
# Bentuk umum perintah if - elif - else:
# if (kondisi 1):
#     statemen
# elif (kondisi 2):
#     statemen
# else:
#     statemen
# Program pengecekan suhu air
suhu = int(input("Masukkan suhu air (°C): "))

if suhu > 100:
    print("Air dalam keadaan mendidih.")
elif suhu >= 0:
    print("Air dalam bentuk cair.")
else:
    print("Air membeku.")
```

2.5 If Bersarang

Kondisi bersarang (nested condition) adalah kondisi yang ditempatkan di dalam kondisi lain. Artinya, di dalam salah satu cabang dari sebuah pernyataan kondisi, dapat terdapat pengujian kondisi tambahan.

Dengan kata lain, ketika terdapat dua atau lebih percabangan, salah satu cabangnya dapat berisi pernyataan if lain untuk melakukan pengecekan lebih lanjut terhadap situasi tertentu.

```
# Program If Bersarang
umur_A = int(input("Masukkan umur orang pertama: "))
umur_B = int(input("Masukkan umur orang kedua: "))

if umur_A == umur_B:
    print("Keduanya memiliki umur yang sama.")
else:
    if umur_A > umur_B:
        print("Orang pertama lebih tua dari orang kedua.")
    if umur_A < umur_B:
        print("Orang pertama lebih muda dari orang kedua.")
```

2.6 If Ternary

Dalam Python, Operator Ternary menentukan apakah suatu kondisi benar atau salah dan kemudian mengembalikan nilai yang sesuai sebagai hasilnya. Operator ini berguna dalam kasus di mana kita perlu menetapkan nilai ke variabel agar lebih sederhana, Ini berguna ketika kita ingin menghindari penulisan beberapa baris untuk kondisi if-else sederhana. Seperti dalam if-else sederhana, true_value akan dieksekusi ketika kondisi yang diberikan dalam ekspresi adalah True. Jika kondisi mengembalikan False, maka false_value akan dieksekusi.

Contoh:

a = 10

b = 20

#contoh mengembalikan nilai pada ternary opet = "opet harus belajar" if a < b else "opet tak perlu belajar" print(opet).

Fitur	Seleksi Kondisi	Operator Ternary
Bentuk	Lebih lengkap, menggunakan if-else	Lebih singkat, menggunakan if else
Fungsi	Mengalirkan eksekusi ke blok kode	Mengembalikan nilai
Penggunaan	Untuk keputusan yang lebih kompleks, sering digunakan dalam struktur kontrol lainnya	Untuk keputusan sederhana yang menghasilkan nilai
Keluaran	Tidak langsung menghasilkan nilai, melainkan menjalankan blok kode	Langsung menghasilkan nilai

Contoh:

a = 10

b = 20

print("doni anak baik" if a == b else "Doni bukan anak yang baik")

2.7 Contoh Soal Latihan 1

a. Di bawah ini adalah contoh program Penyeleksian kondisi pada Bahasa pemrograman Python.

```
angka_rahasia = 5
print("Saya punya angka rahasia antara 1 sampai 10!")
tebakan = int(input("Coba tebak angkanya: "))
if tebakan == angka_rahasia:
    pesan = "Hebat! Tebakan kamu benar!\nTapi sayangnya tidak ada hadiahnya."
else:
```

```
    pesan = "Terlalu kecil." if tebakan < angka_rahasia else  
    "Terlalu besar."  
    pesan += " Coba lagi!"  
print(pesan)  
print("Permainan selesai, terima kasih sudah bermain!")
```

Hasil dari outputnya:

```
Saya punya angka rahasia antara 1 sampai 10!  
Coba tebak angkanya: 5  
Hebat! Tebakan kamu benar!  
Tapi sayangnya tidak ada hadiahnya.  
Permainan selesai, terima kasih sudah bermain!
```

b. Contoh program Penyeleksian kondisi

```
umur = int(input("Masukkan usia kamu: "))  
if umur < 13:  
    print("Anak-anak")  
elif umur <= 18:  
    print("Remaja")  
elif umur <= 35:  
    print("Dewasa")  
else:  
    print("Lansia")
```

Output yang dikeluarkan seperti ini

```
Masukkan usia kamu: 18  
Remaja
```

c. Menentukan ganjil genap

```
nilai = int(input("masukkan angka: "))  
if nilai%2:  
    print ("bilangan ganjil")  
else:  
    print ("bilangan genap")
```

Outputnya:

```
masukkan angka: 5
bilangan ganjil
```

2.8 Contoh Soal Latihan 2

Buatlah program jika andi memasukan nilai 1 sampai 9 , maka outputnya “angka anda satu “ – “angka anda Sembilan “ menggunakan operasi if elif dan else

```
angka = int(input("Masukkan angka antara 0 sampai 9: "))
angka_bro = {
    0: "nol",
    1: "satu",
    2: "dua",
    3: "tiga",
    4: "empat",
    5: "lima",
    6: "enam",
    7: "tujuh",
    8: "delapan",
    9: "sembilan"
}

if angka in angka_bro:
    print("Angka anda", angka_bro[angka])
else:
    print("Angka di luar jangkauan! (hanya 0-9)")
```

Outputnya:

```
Masukkan angka antara 0 sampai 9: 8
Angka anda delapan
```


BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Jelaskan pengertian Penyeleksian kondisi. menggunakan Bahasa kalian sendiri. Berikan juga analogi sederhana (misalnya dalam kehidupan sehari-hari) agar lebih mudah dipahami.
2. Apa perbedaan antara if, if-else, if-elif-else, dan if bersarang? jelaskan dengan bahasa kalian sendiri dan berikan contoh kodenya!
3. Bagaimana aturan penulisan struktur Penyeleksian kondisi pada Python?
4. Jelaskan fungsi penyeleksian kondisi dalam pemrograman! kenapa tanpa penyeleksian kondisi sebuah program akan menjadi kacau?
5. Buatlah Program dengan studi kasus berikut:
Sebuah jalan tol memberikan ketentuan pembayaran sebagai berikut: Tarif dasar tol berbeda sesuai jenis kendaraan:
 - mobil pribadi → Rp 15.000
 - truk kecil → Rp 25.000
 - truk besar → Rp 40.000
 Diskon tarif tol berlaku jika:
 - jika menggunakan e-money → diskon 10%
 - jika pembayaran menggunakan e-money dan dilakukan pada jam serpi (23.00 - 05.00) → diskon 20%
 - jika pembayaran tunai → tidak ada diskon

3.2 Jawaban

1. Penyeleksi kondisi adalah cara program mengambil keputusan berdasarkan syarat. Analogi: kalau hujan bawa payung, kalau tidak hujan tidak bawa.
2. if → mengecek satu kondisi saja
 if-else → kalau salah jalankan yang lain.
 if-elif-else → mengecek banyak kondisi berturut-turut.
 if bersarang → if di dalam if
 contoh kodenya:

```
if suhu > 30:
    print ("cuaca Panas")

if suhu > 30:
    print ("cuaca Panas")
else:
    print ("cuaca adem")
```

Jika suhu lebih dari 30 maka "cuaca Panas" jika tidak, maka tidak ada yang terjadi.

if-else adalah percabangan yang berbagai jalurnya ada dua, satu benar dan satu salah contoh:

```
if suhu > 30:
```

```
    print ("cuaca Panas")
```

```
else:
```

```
    print ("cuaca adem")
```

artinya: jika cuaca atau suhu lebih dari 30 maka "cuaca Panas" tapi jika tidak maka cuaca dingin.

if-elif-else hanya digunakan ketika ada banyak kondisi yang perlu diperiksa secara berurutan.

Contoh:

```
if suhu > 30:
```

```
    print ("cuaca Panas")
```

```
elif suhu > 20:
```

```
    print ("cuaca adem")
```

```
else:
```

```
    print ("cuaca dingin")
```

If bersarang (nested if) digunakan pada saat ingin memeriksa kondisi lebih kompleks yang bergantung pada kondisi sebelumnya, atau biasanya disebut if di dalam if.

Contoh:

```
if suhu < 23:
```

```
    if suhu ≤ 17:
```

```
        print ("cuaca sejuk sekali dan hujan")
```

```
else:
```

```
    print ("sejuk tapi tidak hujan")
```

```
else:
```

```
    print ("tidak hujan dan cerah")
```

```

3. Gunakan kata kunci: if, elif, else.
   setelah kondisi beri tanda:
   Perintah di dalam blok harus indent (spasi/tab)
   contoh:
   if x > 0:
       print ("Positif")
   elif x == 0:
       print ("Nol")
   else:
       print ("Negatif")

4. Penyeleksi kondisi membuat program bisa mengambil
   keputusan sesuai situasi. Tanpa program jadi kaku karena
   hanya berjalan lurus tanpa menyesuaikan keadaan.

5. kendaraan = "mobil"
   pembayaran = "e-money"
   jam = 23
   if kendaraan == "mobil":
       tarif = 15000
   elif kendaraan == "truk kecil":
       tarif = 25000
   elif kendaraan == "truk besar":
       tarif = 40000
   else:
       tarif = 0
       print ("jenis kendaraan tidak dikenali")
   if tarif > 0:
       if pembayaran == "e-money":
           if jam >= 23 or jam <= 5:
               diskon = 0.20
           else:
               diskon = 0.10
       else:
           diskon = 0
   total_bayar = tarif - (tarif * diskon)
   print (f"jenis kendaraan: {kendaraan}")
   print (f"tarif dasar: Rp {tarif}")
   print (f"metode pembayaran: {pembayaran}")
   print (f"jam pembayaran: {jam}.00")
   print (f"Diskon: {int(diskon * 100)}%")
   print (f"total bayar: Rp {int(total_bayar)}")
  
```

BAB IV

IMPLEMENTASI

4.1 Tugas Praktikum

4.1.1 Seorang dosen ingin membuat program sederhana untuk menentukan nilai huruf mahasiswa berdasarkan skor ujian (0–100) dengan ketentuan sebagai berikut :

Keterangan	Nilai
Jika nilai > 85	A
70 – 84	B
60 - 69	C
50 - 59	D
< 50	E

Kemudian program juga memiliki fitur tambahan yaitu jika nilai mahasiswa ≥ 85 dan nilai kehadiran $\geq 90\%$, maka diberikan status "Lulus dengan Pujian". Buatlah program dengan menggunakan Flowchart yang dimana program dapat menerima input nilai ujian dan persentase kehadiran, lalu cetak nilai huruf dan status kelulusan.

4.1.2 Sebuah bioskop memiliki aturan harga tiket:

Kategori	Harga/Diskon
Tiket normal	Rp.50.000
anak-anak < 12 tahun	Diskon 50%
Diskon 30%	Diskon 30%
penonton hari Selasa (promo)	Diskon 20%

Jika ada lebih dari satu kondisi diskon yang berlaku, gunakan diskon terbesar. Buatlah program python yang dimana program meminta input usia, status pelajar (ya/tidak), dan hari, lalu cetak harga tiket yang harus dibayar.

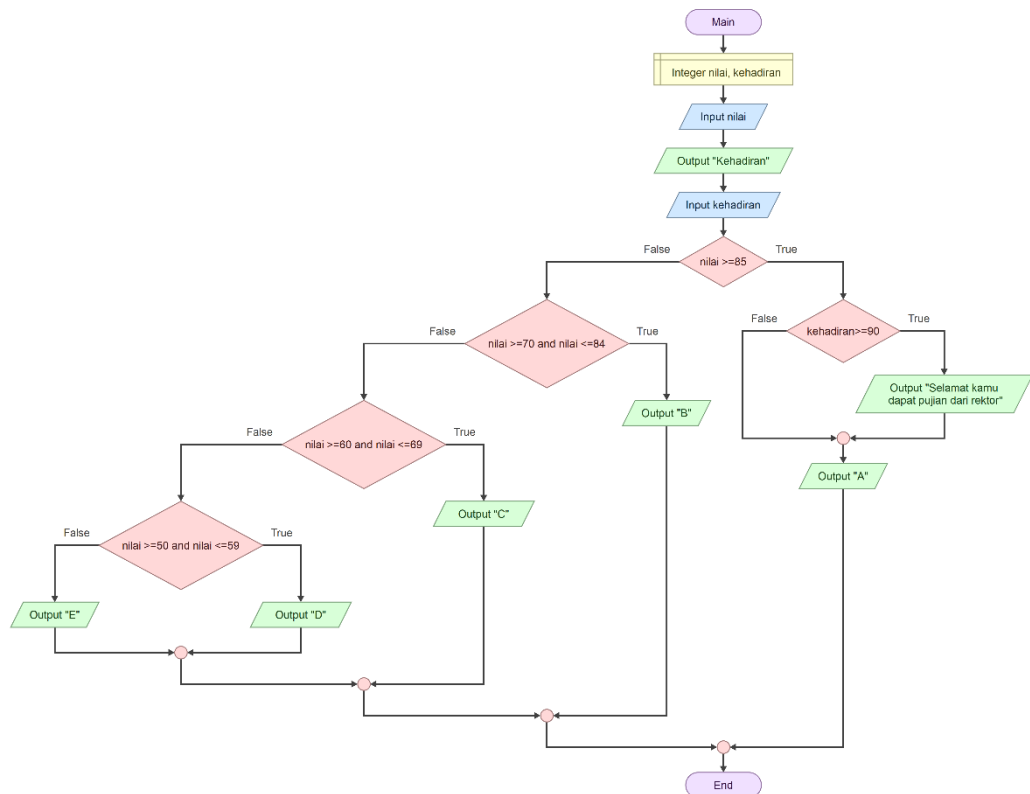
4.1.3 Sebuah mall menerapkan tarif parkir dengan ketentuan sebagai berikut

- a. 2 jam pertama : Rp5.000
- b. Setiap jam berikutnya : Rp3.000
- c. Maksimal tarif parkir per hari : Rp20.000

Kemudian mall ini juga memiliki aturan tambahan yaitu jika pengunjung adalah member VIP, biaya parkir langsung Rp0. Buatlah program python yang menerima input lama parkir (jam) dan status VIP, lalu hitung total biaya parkir.

4.2 Source Code

4.2.1 Tugas Praktikum Soal No. 1



4.2.2 Tugas Praktikum Soal No. 2

```

harga_tiket = 50000
usia = int(input("Masukkan usia Anda: "))
pelajar = input("Apakah Anda pelajar SMA dengan kartu pelajar? (ya/tidak): ")
hari = input("Masukkan hari: ")
diskon = 0
if usia < 12:
    diskon = 50
if pelajar == "ya" or pelajar == "Ya":
    if diskon < 30:
        diskon = 30
  
```



```

if hari == "Selasa" or hari == "selasa":
    if diskon < 20:
        diskon = 20
harga_akhir = harga_tiket - (harga_tiket * diskon / 100)
print("Usia Anda:", usia, "tahun")
print("Status Pelajar:", pelajar)
print("Hari:", hari)
print("Diskon:", diskon, "%")
print("Total yang harus dibayar: Rp", int(harga_akhir))

```

4.2.3 Tugas Praktikum Soal No.3

```

harga_tiket = 50000
usia = int(input("Masukkan usia Anda: "))
pelajar = input("Apakah Anda pelajar SMA dengan kartu pelajar? (ya/tidak): ")
hari = input("Masukkan hari: ")
diskon = 0
if usia < 12:
    diskon = 50
if pelajar == "ya" or pelajar == "Ya":
    if diskon < 30:
        diskon = 30
if hari == "Selasa" or hari == "selasa":
    if diskon < 20:
        diskon = 20
harga_akhir = harga_tiket - (harga_tiket * diskon / 100)
print("Usia Anda:", usia, "tahun")
print("Status Pelajar:", pelajar)
print("Hari:", hari)
print("Diskon:", diskon, "%")
print("Total yang harus dibayar: Rp", int(harga_akhir))

```

4.3 Hasil

4.3.1 Tugas Praktikum Soal No. 1



4.3.2 Tugas Praktikum Soal No. 2

```
masukkan usia Anda: 19
Apakah Anda pelajar SMA dengan kartu pelajar? (ya/tidak): ya
Masukkan hari: 8
Usia Anda: 19 tahun
Status Pelajar: ya
Hari: 8
Diskon: 30 %
Total yang harus dibayar: Rp 35000
```

4.3.3 Tugas Praktikum Soal No. 3

```
lama_parkir = int(input("Masukkan lama parkir (jam): "))
vip = input("Apakah Anda member VIP? (ya/tidak): ")
if vip == "ya" or vip == "Ya":
    total = 0
    print("Anda member VIP, biaya parkir gratis.")
elif vip == "tidak" or vip == "Tidak":
    if lama_parkir <= 2:
        total = 5000
    else:
        total = 5000 + (lama_parkir - 2) * 3000
        if total > 20000:
            total = 20000
```

```
else:
    total = 0
    print("Input tidak valid. Harap masukkan 'ya' atau 'tidak'.")
print("Total biaya parkir: Rp", total)
```

4.4 Penjelasan

4.4.1 Tugas Praktikum Soal No. 1

Sebuah program sederhana untuk menentukan nilai huruf mahasiswa berdasarkan skor ujian yang diperoleh. Program dimulai dengan simbol Main yang merupakan titik awal eksekusi, kemudian mendeklarasikan variabel bertipe Integer dengan nama "nilai". Setelah deklarasi, program meminta input nilai dari pengguna melalui simbol Input. Proses penentuan nilai huruf dilakukan melalui serangkaian keputusan bersarang (nested decision) yang mengevaluasi skor nilai secara bertahap dari nilai tertinggi hingga terendah.

Pengecekan pertama memeriksa apakah nilai lebih besar atau sama dengan 85. Jika kondisi ini bernilai True, maka program langsung mengoutput "A" dan melanjutkan ke output "Lulus dengan pujian", kemudian semua alur digabungkan menuju simbol End. Namun jika kondisi bernilai False, program akan melanjutkan ke pengecekan berikutnya untuk mengevaluasi rentang nilai yang lebih rendah.

Pengecekan selanjutnya dilakukan secara berurutan dengan menggunakan kondisi AND untuk menentukan rentang nilai. Pengecekan kedua memeriksa apakah nilai berada dalam rentang 70 hingga 84 dengan kondisi "nilai >=70 and nilai <=84" yang mengoutput "B" jika True. Pengecekan ketiga memeriksa rentang 60-69 dengan kondisi "nilai >=60 and nilai <=69" yang mengoutput "C" jika True. Pengecekan terakhir memeriksa rentang 50-59 dengan kondisi "nilai >=50 and nilai <=59" yang mengoutput "D" jika True.

Jika semua kondisi sebelumnya bernilai False, maka secara otomatis program akan mengoutput "E" karena nilai dianggap di bawah 50. Semua alur output dari B, C, D, dan E akan bertemu di junction node (titik penghubung kecil berbentuk lingkaran) dan mengalir bersama-sama menuju simbol End yang

menandakan program selesai. Struktur bersarang ini memastikan bahwa setiap nilai yang diinput akan mendapat tepat satu kategori huruf yang sesuai dengan ketentuan grading yang telah ditetapkan, dengan alur eksekusi yang efisien dan logis.

4.4.2 Tugas Praktikum Soal No. 2

Program ini berfungsi untuk menghitung harga tiket bioskop berdasarkan tiga kondisi, yaitu usia penonton, status pelajar, dan hari menonton. Harga dasar tiket ditetapkan sebesar Rp50.000, kemudian pengguna diminta memasukkan usia, apakah ia pelajar SMA dengan kartu pelajar, serta hari menonton. Nilai diskon awal diatur 0, dan program akan memeriksa setiap kondisi secara berurutan menggunakan struktur percabangan if. Jika usia kurang dari 12 tahun maka mendapat diskon 50%, jika pelajar SMA mendapat 30%, dan jika menonton pada hari Selasa mendapat 20%.

Program memastikan hanya diskon terbesar yang diterapkan meskipun beberapa kondisi terpenuhi. Setelah menentukan diskon, program menghitung harga akhir dengan rumus $\text{harga_akhir} = \text{harga_tiket} - (\text{harga_tiket} * \text{diskon} / 100)$. Hasil akhirnya menampilkan usia, status pelajar, hari menonton, besar diskon, dan total harga yang harus dibayar. Program ini menunjukkan penerapan logika percabangan untuk pengambilan keputusan dan perhitungan sederhana dalam bahasa Python.

4.4.3 Tugas Praktikum Soal No. 3

Program di atas berfungsi menghitung total biaya parkir di mall berdasarkan lama parkir dan status keanggotaan VIP. Pengguna diminta memasukkan lama parkir (jam) serta menjawab apakah merupakan member VIP (“ya” atau “tidak”). Jika VIP, biaya parkir gratis (Rp0); jika bukan, tarifnya Rp5.000 untuk dua jam pertama dan Rp3.000 untuk setiap jam berikutnya, dengan batas maksimal Rp20.000 per hari. Jika total melebihi batas tersebut, nilainya diset Rp20.000. Apabila input status tidak valid, program menampilkan pesan kesalahan dan biaya parkir ditetapkan Rp0, lalu hasil akhir ditampilkan di layar.

BAB V

PENUTUP

5.1 Analisa

Dalam praktikum ini, program dibuat menggunakan struktur percabangan if-elif-else untuk mengubah angka menjadi bentuk kata. Setiap kondisi if digunakan untuk memeriksa nilai input pengguna dari 0 hingga 9, kemudian menampilkan hasil dalam bentuk teks. Program juga menggunakan bagian else untuk menangani input yang tidak sesuai (misalnya angka di luar rentang 0–9). Dari hasil percobaan, dapat disimpulkan bahwa logika percabangan bekerja dengan baik dalam membedakan kondisi yang berbeda dan menampilkan output yang sesuai berdasarkan nilai yang dimasukkan.

5.2 Kesimpulan

Dapat disimpulkan bahwa penggunaan struktur percabangan (if-elif-else) sangat efektif untuk membuat program yang membutuhkan pengambilan keputusan berdasarkan kondisi tertentu. Melalui praktikum ini, kita dapat memahami cara kerja logika seleksi dalam Python, bagaimana input pengguna diproses, serta bagaimana setiap kondisi menghasilkan output yang berbeda sesuai dengan nilai yang dimasukkan. Operator aritmatika adalah fondasi dasar dalam pemrograman untuk operasi matematika.

- Program menggunakan struktur if-elif-else untuk menentukan output berdasarkan nilai yang dimasukkan pengguna. Penanganan kasus khusus seperti pembagian nol penting untuk mencegah error program.
- Setiap kondisi if mewakili angka tertentu (0–9) dan menghasilkan teks sesuai angka tersebut. Pondasi dasar python menggunakan operator aritmatika dan implementasinya.
- Penggunaan fungsi input() memungkinkan program menerima data. Bagian else berfungsi untuk menampilkan pesan kesalahan jika angka yang dimasukkan berada di luar jangkauan dari pengguna.
- Praktikum ini membuktikan bahwa percabangan logika sangat penting dalam pengambilan keputusan pada program Python.