LAPORAN RESMI
MODUL II
(PENYELEKSIAN KONDISI)
ALGORITMA PEMROGRAMAN

NAMA	**:**	**FEBRYAN SETYA WIBISONO**
N.R.P	**:**	**240441100134**
DOSEN	**:**	**FITRI DAMAYANTI. S.kom, M.kom**
ASISTEN	**:**	**ISMA' YAFA NUR ZAMZAMI RAMADHANI**
TGL PRAKTIKUM	**:**	**02 OKTOBER 2024**

Disetujui : 08-10-2024
Asisten

ISMA' YAFA NUR ZAMZAMI RAMADHANI
23.04.411.00081

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

BAB I
PENDAHULUAN

1.1 Latar Belakang

Penyeleksian kondisi dalam program merupakan konsep dasar yang sangat penting karena memungkinkan program membuat keputusan berdasarkan data atau input yang diterima. Proses ini memainkan peran penting dalam pengambilan keputusan, sehingga program dapat merespons berbagai situasi yang mungkin terjadi. Tanpa adanya penyeleksian kondisi, program akan kesulitan untuk beradaptasi dengan variasi input pengguna dan hanya akan berjalan dengan cara yang sama setiap saat.

Selain itu, penyeleksian kondisi juga bertugas mengatur alur eksekusi program. Artinya, langkah-langkah yang dijalankan oleh program bisa berbeda tergantung pada kondisi yang diberikan. Misalnya, dalam aplikasi e-commerce, penyeleksian kondisi digunakan untuk memeriksa apakah stok barang tersedia. Jika stok ada, pengguna dapat melanjutkan ke tahap pembayaran; jika stok habis, program akan menampilkan pesan bahwa produk tidak tersedia.

Struktur kontrol yang sering digunakan dalam penyeleksian kondisi meliputi 'if', 'else if', dan 'else'. Sebagai contoh, dalam kode sederhana yang memberikan grade berdasarkan nilai ujian, kita dapat melihat bagaimana penyeleksian kondisi digunakan untuk menentukan hasil yang tepat. Tanpa penyeleksian kondisi, program akan kehilangan fleksibilitas dan tidak mampu menyesuaikan diri dengan kebutuhan pengguna, sehingga pengalaman pengguna menjadi monoton dan kurang dinamis.

1.2 Tujuan

- Mampu memahahami konsep Penyeleksian Kondisi
- Mampu menerapkan Penyeleksian Kondisi dalam Program
- Mampu untuk penyeleksian kondisi juga dapat meningkatkan efisiensi dalam menjalankan program

BAB II
DASAR TEORI

2.1 Pengertian Penyeleksian Kondisi

Penyeleksian kondisi adalah konsep dalam pemrograman yang memungkinkan kode untuk membuat keputusan berdasarkan kondisi tertentu. Dengan penyeleksian kondisi, program dapat menjalankan blok kode yang berbeda tergantung pada apakah suatu kondisi terpenuhi (benar) atau tidak (salah). Biasanya, penyeleksian kondisi menggunakan struktur seperti :

1. Perintah if
2. Perintah if – else
3. Perintah if – elif – else
4. Perintah if bersarang

2.2 Perintah If

Bentuk umum perintah if :

If (kondisi)

Statemen

Pernyataan If digunakan untuk memeriksa suatu kondisi di mana jika kondisi tersebut bernilai benar (True), program akan menjalankan blok kode yang berada di bawahnya. Di Python, pernyataan kondisi dan blok kode yang dieksekusi dipisahkan oleh tanda titik dua (`:`). Sebagai ilustrasi, berikut contoh penerapannya:

```
# program perintah if
n = "python"
if n == "python":
    print("hello " + n)
```

Setiap kali menulis pernyataan `if`, setelah menentukan kondisi, harus diikuti dengan tanda titik dua (`:`). Tanda ini menunjukkan bahwa jika kondisi tersebut benar, maka baris-baris kode yang mengikuti tanda titik dua akan dieksekusi oleh program.

2.3 Perintah If_Else

Pernyataan if – else digunakan untuk menyeleksi kondisi di mana jika kondisi tersebut bernilai benar, program akan menjalankan statemen 1. Namun, jika kondisi

bernilai salah, maka statemen 2 yang akan dijalankan oleh program. Ini memungkinkan program merespons kedua kemungkinan hasil dari kondisi yang diuji.

Bentuk umum perintah if – else

```
If (kondisi) :
        Statemen 1
Else :
        Statemen 2
```

```
# perintah if else
key = "python"
password = input ("masukkan password : ")
if password == key :
    print("password benar")
else:
    print("password salah")
```

2.4 Perintah If – Elif – Else

Pernyataan `if – else – elif` digunakan untuk melakukan penyeleksian kondisi ketika terdapat lebih dari satu kondisi yang perlu dievaluasi. Jika kondisi pertama bernilai benar, maka program akan menjalankan blok kode yang terkait. Jika tidak, program akan melanjutkan untuk memeriksa kondisi kedua, dan seterusnya. Berikut adalah bentuk umum dari pernyataan `if – else – elif`:

```
If (kondisi 1) :
    Statemen
Elif (kondisi 2) :
    statemen
  Else:
    Statemen
```

```
#perintah if - elif - else
angka = int(input("masukkan sebuah bilangan : "))
if angka > 0:
print("angka adalah bilangan positif")
```

```
elif angka > 0:
print("angka adalah bilangan negatif")
else:
print("angka adalah 0")
```

2.5 If Bersarang

Kondisi bersarang adalah kondisi yang terletak di dalam kondisi lain. Jika terdapat dua cabang kondisi, salah satu dari cabang tersebut dapat diisi dengan kondisi tambahan. Dengan cara ini, program dapat mengevaluasi lebih dari satu tingkat kondisi secara hierarkis. Sebagai contoh, berikut adalah penerapan kondisi bersarang:

```
# if bersarang
x = int (input("masukkan nilai x="))
y = int(input("masukkan nilai y="))
if x == y:
     print("niiai", x , "dan" , y , "mempunyai nilai yang
sama")
else :
    if x > y :
        print(x, "lebih besar dari", y)
if x < y :
    print(x, "lebih kecil dari", y)
```

2.6 If Ternary

Dalam Python, operator ternary berfungsi untuk mengevaluasi apakah suatu kondisi bernilai benar atau salah, dan kemudian mengembalikan nilai yang sesuai sebagai hasilnya. Operator ini sangat berguna ketika kita perlu menetapkan nilai pada variabel dengan cara yang lebih ringkas. Dengan menggunakan operator ternary, kita dapat menghindari penulisan beberapa baris kode yang diperlukan untuk kondisi `if-else` yang sederhana.

Pada operator ini, jika kondisi yang diberikan dalam ekspresi bernilai True, maka true value akan dieksekusi. Sebaliknya, jika kondisi tersebut mengembalikan False, maka false value yang akan dijalankan..

Contoh:

a = 10

b = 20

#contoh mengembalikan nilai pada ternary

Opet = ”opet harus belajar” if a < b else “opet tidak 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	Mengebalikan 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 Praktikum

2.7.1 Latihan 1

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

```
# Latihan 1
nomor_acak = 7
print("Tebak nomor acak dari 1 - 10")
tebakan = int(input("Tebakan anda (bil bulat): "))
if tebakan == nomor_acak:
    print("Selamat! tebakan anda benar")
```

```
    print("tapi tidak ada hadiah untuk anda ")
elif tebakan < nomor_acak:
    print("Tebakan anda terlalul kecil")
else :
    print("Tebakan anda terlalu besar")
    print("selesai")
```

2. Contoh program penyeleksian kondisi

```
a = int(input("Masukkan Umur: "))
if a <= 15 :
    print("Muda")
elif a <= 20 :
    print("Remaja")
else :
    print("Tua")
```

3. Menentukan Ganjil Genap

```
#ganjil dan genap
nilai = int(input("Masukkan angka :"))
if nilai % 2 :
    print("Bilangan Ganjil")
else :
    print("Bilangan Genap")
```

2.7.2 Latihan 2

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

```
#angka ke kata

a = int(input("Masukkan angka (0-9): "))
if a==0:
    print("angka anda nol")
elif a==1:
    print("angka anda satu")
elif a==2:
    print("angka anda dua")
elif a==3:
    print("angka anda tiga")
elif a==4:
    print("angka anda empat")
elif a==5:
    print("angka anda lima")
elif a==6:
    print("angka anda enam ")
elif a==7:
    print("angka anda tujuh")
elif a==8:
    print("angka anda delapan")
elif a==9:
    print("angka anda sembilan")
else:
    print("angka anda tidak ditemukan")
```

Isma

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 scriptnya.
4. Jelaskan fungsi dari Penyeleksian kondisi.
5. Jelaskan perbedaan antar seleksi kondisi dan operator ternary.

3.2 Jawaban

1. Pada umumnya dalam membuat program, selalu ada seleksi dimana diperlukan pengecekan suatu kondisi untuk mengarahkan program agar berjalan sesuai keinginan.
2. - Perintah if
   - Perintah if - else
   - perintah if - elif - else
   - Perintah if bersarang
3. Contoh soal : Buatlah program yang meminta pengguna untuk memasukkan usia. Jika usia di bawah 18, tampilkan "Anda masih di bawah umur". Jika usia antara 18 sampai 65, tampilkan "Anda adalah orang dewasa". Jika usia diatas 65, tampilkan "Anda adalah lansia".

script :
```
usia = int (input ("masukkan usia anda:"))
if usia<18 :
    print ("Anda masih di bawah umur")
elif 18 <= usia <= 65 :
    print ("Anda adalah orang dewasa")
else :
    print ("Anda adalah lansia")
```

Penjelasan : 1. Input usia : program meminta pengguna untuk memasukkan usia dan mengonversinya ke tipe data int (bilangan bulat)

2. Penyeleksian kondisi
   - if usia <18 : memeriksa apakah usia kurang dari 18. Jika benar, program mencetak "Anda masih di bawah umur"

- elif 18 <= usia <= 65 : memeriksa apakah usia antara 18 dan 65. Jika benar, program mencetak "Anda adalah orang dewasa"
- else : Jika kedua kondisi sebelumnya tidak terpenuhi (berarti usia diatas 65), program mencetak " Anda adalah lansia"

4. 1. Pengambilan keputusan : memungkinkan program untuk mengambil keputusan berdasarkan kondisi tertentu.
   2. Kontrol Alur Ekskusi : Dengan ini programer dapat mengatur /mengontrol alur eksekusi program, mengarahkan ke bagian tertentu dari hasil evaluasi kondisi
   3. Pembuatan Logika Program : Dasar untuk membangun program dengan logika kompleks
   4. Meningkatkan Interaktivitas : program dapat berinteraksi dengan pengguna secara lebih efektif
   5. Pengolahan Data yang Fleksibel : memungkinkan program untuk memproses data yang berbeda dengan cara yang sesuai, meningkatkan fleksibilitas dan efisiensi

5. Seleksi kondisi lebih baik untuk logika kompleks dengan banyak cabang, sedangkan Operator ternary efektif untuk keputusan sederhana dalam satu baris.

BAB IV
IMPLEMENTASI

4.1 Soal Praktikum

4.1.1 Soal No. 1

Buatlah Script Penyeleksian kondisi secara dinamis seperti gambar di bawah ini, dengan ketentuan :

100 – 81 = A

80 – 71 = B

70 – 61 = C

60 – 41 = D

40 – 0 = E

4.1.2 Soal No. 2

soal, jaka dan ida merupakan Mahasiswa sistem informasi, salah satu dari mereka mahasiswayang berprestasi, jaka memiliki skor sebanyak 1100 dan IPKnya sebanyak 3.5 sedangkan ida memiliki skor 1000 dengan ipk 3.5. Buatlah program untuk mengetahui siapa yang lulus persayratan beasiswa jika ditentukan skor yang lebih besar dari 1100 dan ipk minimal 3.0 untuk lulus persyaratan.

4.1.3 Soal No. 3

Buatkan Flowchart dan Pseudocode nya untuk soal nomor 2.

4.1.4 Soal No. 4

Buatlah program untuk menentukan tahun kabisat menggunakan inputan dan seleksi kondisi (carilah rumus yang ada di google dan aplikasikan pada program).

4.1.5 Soal No. 5

Buatlah program Diskon pembelian minuman di sebuah bar. Jika pembeli memiliki kartu member akan mendapatkan diskon 15%, jika total belanja lebih dari Rp500.000 akan mendapat diskon 10%. Lalu tampilkan nama Pembeli, Diskon yang didapatkan, total harga sebelum diskon dan total harga setelah diskon. Tetapi ika usia pembeli dibawah 18 tahun program akan berhenti dan menampilkan "Maaf usia anda belum mencukupi" pada program.

4.2 Tugas

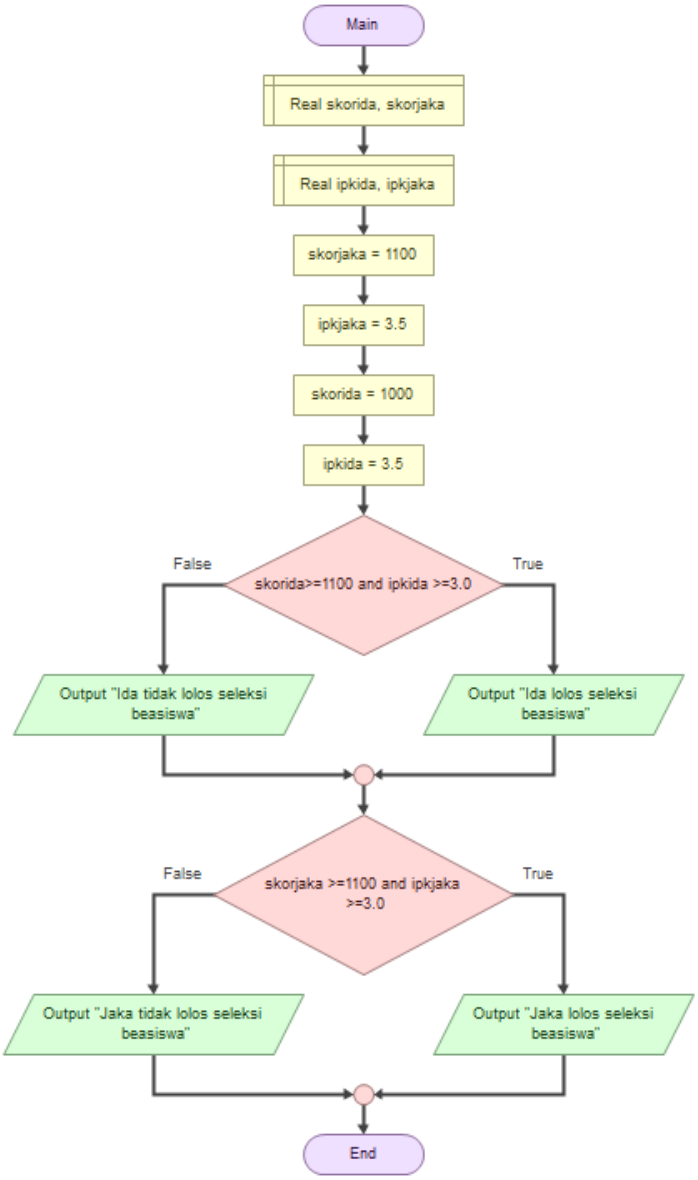
4.2.1Tugas Praktikum Soal No. 1

```
nama = input("masukkan Nama anda : ")
nim = int(input("masukkan NIM : "))
nilai_uts = int(input("masukkan nilai UTS : "))
nilai_uas = int(input("masukkan nilai UAS : "))
nilai_rata_rata = (nilai_uts + nilai_uas) / 2

print("Nama anda", nama)
print("NIM anda", nim)
print(f"nilai rata-rata {nama} adalah {int(nilai_rata_rata)}")

if nilai_rata_rata <= 100 and nilai_rata_rata >=81 :
    print("anda mendapatkan nilai A")
elif nilai_rata_rata <= 80 and nilai_rata_rata >= 71 :
    print("anda mendapatkan nilai B")
elif nilai_rata_rata <= 70 and nilai_rata_rata >= 61 :
    print("anda mendapatkan nilai C")
elif nilai_rata_rata <= 60 and nilai_rata_rata >= 41 :
    print("anda mendapatkan nilai D")
elif nilai_rata_rata <= 40 and nilai_rata_rata >= 0 :
    print("anda mendapatkan nilai E")
else :
    print("nilai yang dimasukkan salah")
```

4.2.2 Tugas Praktikum Soal No. 2

```
q = 1100
p = 1000
r = 3,5
s = 3,0
min_ipk =3,0
min_skor=1100
syarat_lulus_beasiswa=min_ipk and min_skor
if syarat_lulus_beasiswa == q and r :
    print("Jaka lulus beasiswa")
else :
    print("Jaka tidak lulus beasiswa")
if syarat_lulus_beasiswa == p and s :
    print("Ida lulus beasiswa")
else :
    print("Ida tidak lulus beasiswa")
```

4.2.3 Tugas Praktikum Soal No. 3



4.2.4 Tugas Praktikum Soal No. 4

```
tahun = int(input("Masukkan tahun: "))
if tahun % 4 == 0:
    if tahun % 100 == 0:
        if tahun % 400 == 0:
            print(f"Tahun {tahun} merupakan tahun kabisat")
        else:
            print(f"Tahun {tahun} bukan merupakan tahun kabisat")
    else:
        print(f"Tahun {tahun} merupakan tahun kabisat")
else:
    print(f"Tahun {tahun} bukan merupakan tahun kabisat")
```

4.2.5 Tugas Praktikum Soal No. 5

```
# Input untuk nama pembeli dan usia
nama_pembeli = input("Masukkan nama pembeli: ")
usia = int(input("Masukkan usia pembeli: "))

# Cek usia
if usia < 18:
    print("Maaf, usia anda belum mencukupi.")
else:
    # Input total harga
    harga = int(input("Masukkan total belanja: Rp "))
    member = input("Apakah anda memiliki kartu member (ya/tidak)? ")

    # Cek diskon
    diskon = 0
    if member == 'ya' and harga >= 500000:
        diskon += 25
    elif member == 'ya' and harga < 500000:
        diskon += 15
    elif member != 'ya' and harga >= 500000:
        diskon += 10
    else:
        print("Tidak dapat diskon")

    # Hitung total harga setelah diskon
    total_setelah_diskon = harga - (harga * diskon / 100)

    # Tampilkan hasil
    print("Nama Pembeli: ",nama_pembeli)
    print("Diskon yang didapatkan: %",diskon)
    print("Total harga sebelum diskon: Rp ",harga)
    print("Total harga setelah diskon: Rp ",total_setelah_diskon)
```

4.3 Hasil

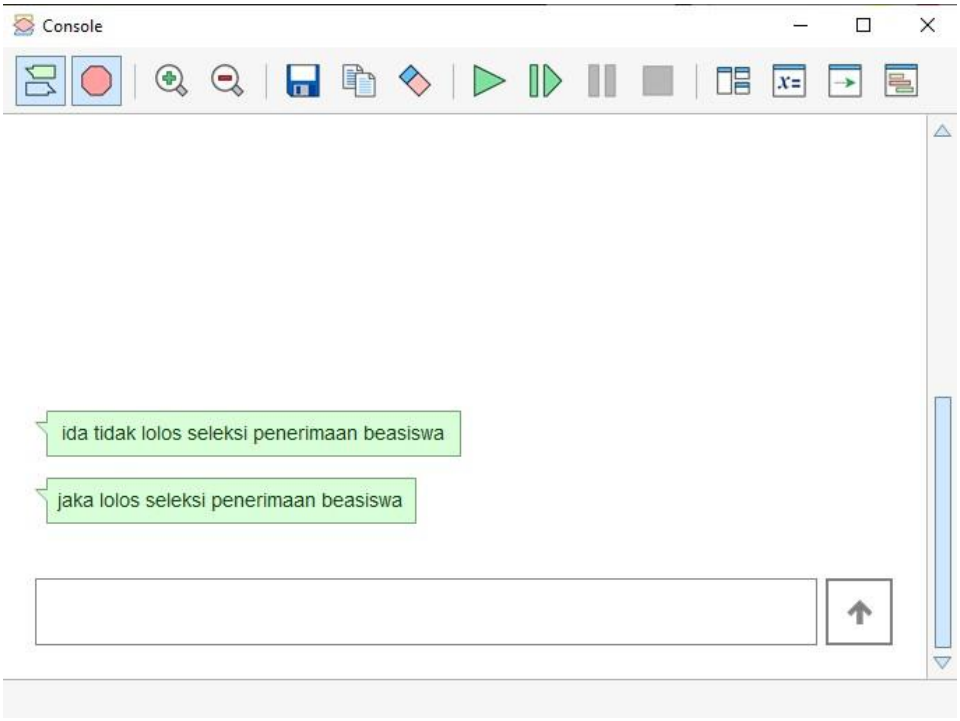
4.3.1 Hasil Praktikum Soal No. 1

```
masukkan Nama anda : Febryan
masukkan NIM : 240441100134
masukkan nilai UTS : 89
masukkan nilai UAS : 89
Nama anda Febryan
NIM anda 240441100134
nilai rata-rata Febryan adalah 89
anda mendapatkan nilai A
```

4.3.2 Hasil Praktikum Soal No. 2

```
Jaka lulus beasiswa
Ida tidak lulus beasiswa
```

4.3.3 Hasil Praktikum Soal No. 3



4.3.4 Hasil Praktikum Soal No. 4

```
Masukkan tahun: 2021
Tahun 2021 bukan merupakan tahun kabisat
```

4.3.5 Hasil Praktikum Soal No. 5

```
Masukkan nama pembeli: Febryan
Masukkan usia pembeli: 19
Masukkan total belanja: Rp 550000
Apakah anda memiliki kartu member (ya/tidak)? ya
Nama Pembeli:  Febryan
Diskon yang didapatkan: % 25
Total harga sebelum diskon: Rp  550000
Total harga setelah diskon: Rp  412500.0
```

4.4 Penjelasan

4.4.1 Penjelasan Praktikum Soal No. 1

Program ini dirancang untuk meminta pengguna memasukkan nama, NIM, serta nilai Ujian Tengah Semester (UTS) dan Ujian Akhir Semester (UAS). Setelah menerima input, program menghitung nilai rata-rata dengan menjumlahkan nilai UTS dan UAS, lalu membaginya dengan dua. Hasil perhitungan ini kemudian ditampilkan bersamaan dengan nama dan NIM pengguna. Selanjutnya, program

memberikan grade berdasarkan nilai rata-rata yang telah dihitung: pengguna akan mendapatkan nilai "A" jika rata-ratanya berada di antara 81 hingga 100, "B" untuk nilai 71 hingga 80, "C" untuk nilai 61 hingga 70, "D" untuk nilai 41 hingga 60, dan "E" untuk nilai di bawah 40. Jika nilai rata-rata berada di luar rentang yang ditentukan, program akan menampilkan pesan bahwa nilai yang dimasukkan salah. Namun, penting untuk dicatat bahwa program ini tidak melakukan validasi untuk memastikan bahwa nilai UTS dan UAS berada dalam rentang 0 hingga 100.

4.4.2 Penjelasan Praktikum Soal No. 2

Program Python ini dimulai dengan mendefinisikan skor dan IPK Jaka serta Ida, serta syarat minimal untuk lulus beasiswa, yaitu IPK minimal 3.0 dan skor minimal 1100. Variabel `syarat_lulus_beasiswa` menyimpan syarat ini. Program kemudian mengecek apakah Jaka dan Ida memenuhi syarat tersebut. Jika skor dan IPK Jaka memenuhi, program akan mencetak "Jaka lulus beasiswa", jika tidak, mencetak "Jaka tidak lulus beasiswa". Hal yang sama berlaku untuk Ida. Jika skor dan IPK Ida memenuhi, program mencetak "Ida lulus beasiswa", jika tidak, mencetak "Ida tidak lulus beasiswa".

4.4.3 Penjelasan Praktikum Soal No. 3

Jika nilai skor dan IPK seseorang sama dengan atau lebih tinggi dari nilai minimum, maka orang tersebut dinyatakan memenuhi syarat untuk mendapatkan beasiswa. Dalam contoh ini, hanya Jaka yang memenuhi syarat karena nilai skornya mencapai 1100, sedangkan Ida tidak memenuhi syarat karena nilai skornya hanya 1000, meskipun IPK mereka sama. Singkatnya, program ini berfungsi seperti sebuah mesin penilai yang secara otomatis menentukan siapa yang berhak mendapatkan beasiswa berdasarkan data yang diberikan. (Sama seperti no. 2 bedanya yang no. 3 menggunakan flogarithm)

4.4.4 Penjelasan Praktikum Soal No. 4

Program Python ini digunakan untuk mengecek apakah sebuah tahun merupakan tahun kabisat atau bukan. Pertama, program meminta input dari pengguna berupa tahun, yang kemudian disimpan dalam variabel tahun sebagai bilangan bulat. Setelah itu, program melakukan pengecekan dengan menggunakan kondisi:

Syarat kabisat yaitu : Sebuah tahun dianggap kabisat jika habis dibagi 4 (tahun % 4 == 0), **dan** tidak habis dibagi 100 (tahun % 100 != 0), **atau** tahun tersebut habis dibagi 400 (tahun % 400 == 0). Jika salah satu kondisi ini terpenuhi, maka program akan mencetak bahwa tahun tersebut adalah tahun kabisat. Jika tidak, program mencetak bahwa tahun tersebut bukan tahun kabisat. Misalnya, tahun 2020 akan dikenali sebagai tahun kabisat karena memenuhi syarat pertama (habis dibagi 4 dan tidak habis dibagi 100). Sebaliknya, tahun 1900 bukan tahun kabisat karena meskipun habis dibagi 4, tahun tersebut juga habis dibagi 100 dan tidak habis dibagi 400.

4.4.5 Penjelasan Praktikum Soal No. 5

Program Python ini meminta pengguna untuk memasukkan nama dan umur. Jika umur 18 tahun atau lebih, program mencetak "baik"; jika tidak, program mencetak "maaf umur anda belum cukup" dan keluar. Selanjutnya, program menanyakan apakah pengguna memiliki kartu member dan total belanja mereka. Diskon diberikan berdasarkan dua kriteria: 15% untuk pemegang kartu member dan 10% untuk belanja di atas 500.000. Total diskon dihitung dan kemudian dikurangkan dari jumlah belanja untuk mendapatkan total harga setelah diskon. Program kemudian mencetak nama pengguna, total harga sebelum diskon, dan total diskon yang diterima.

BAB V
PENUTUP

5.1 Analisa

Penyelesaian kondisi dalam pemrograman merupakan metode untuk mengarahkan jalannya program berdasarkan syarat-syarat yang telah ditentukan. Dengan menggunakan struktur seperti if-else, kita dapat memeriksa kondisi tertentu dan mengambil keputusan berbeda sesuai dengan hasil yang didapat. Ini membuat kode kita lebih fleksibel dan responsif terhadap berbagai input yang diterima.

Penyeleksian kondisi juga memungkinkan kita untuk membangun logika program yang lebih kompleks. Dengan menggabungkan beberapa kondisi menggunakan operator logika seperti AND dan OR, kita dapat menciptakan alur program yang lebih mendalam. Misalnya, dalam program yang mengelola data mahasiswa, kita bisa mengecek kehadiran dan prestasi akademik secara bersamaan.

Memahami dan memanfaatkan penyeleksian kondisi dengan baik sangat penting. Ini tidak hanya meningkatkan efektivitas program, tetapi juga memudahkan proses debugging. Ketika terjadi kesalahan, kita dapat dengan cepat menemukan kondisi mana yang tidak terpenuhi. Dengan keterampilan ini, kita bisa menciptakan aplikasi atau sistem yang lebih canggih dan berkualitas tinggi.

5.2 Kesimpulan

1. Penyeleksian kondisi adalah elemen vital dalam pemrograman yang mengatur jalannya program berdasarkan syarat-syarat tertentu.
2. Dengan memanfaatkan struktur if-else dan operator logika, program menjadi lebih fleksibel dan dapat menangani berbagai situasi.
3. Kemampuan dalam penyeleksian kondisi tidak hanya membantu dalam mengembangkan logika yang kompleks tetapi juga meningkatkan efisiensi program secara keseluruhan.