**Struktur Kontrol Kondisional dan Perulangan**

Dalam pemrograman, terdapat dua konsep dasar yang sangat penting untuk mengendalikan alur eksekusi program, yaitu struktur kontrol kondisional (conditional control structure) dan struktur perulangan (looping structure). Kedua konsep ini memungkinkan program untuk membuat keputusan dan melakukan tindakan secara berulang-ulang berdasarkan kondisi tertentu.

✅ 1. Struktur Kontrol Kondisional

Struktur kontrol kondisional adalah struktur yang digunakan untuk menentukan jalur eksekusi kode berdasarkan kondisi tertentu. Kondisi ini dievaluasi sebagai true (benar) atau false (salah). Jika kondisi tersebut terpenuhi (true), maka blok kode tertentu akan dijalankan. Jika tidak (false), maka program dapat melakukan tindakan lain atau mengabaikannya.

Struktur kondisional memungkinkan program untuk mengambil keputusan secara dinamis, sehingga program dapat beradaptasi terhadap input atau situasi yang berbeda.

Contoh Sederhana:

Bayangkan sebuah aplikasi yang menentukan apakah seseorang lulus atau tidak berdasarkan nilai ujian. Jika nilai lebih besar atau sama dengan 75, maka siswa tersebut dinyatakan lulus. Jika tidak, maka siswa dinyatakan tidak lulus.

Dalam bahasa pemrograman Python, struktur ini dapat direpresentasikan sebagai berikut:

nilai = 80

if nilai >= 75:

print("Lulus")

else:

print("Tidak Lulus")

Pada contoh di atas, jika nilai adalah 80, maka program akan mencetak "Lulus". Jika nilai kurang dari 75, maka program akan mencetak "Tidak Lulus".

**Jenis-Jenis Struktur Kondisional:**

1. **If Statement:**
   * Hanya mengevaluasi satu kondisi. Jika kondisi tersebut benar, maka blok kode akan dieksekusi.

umur = 18

if umur >= 17:

print("Anda berhak mendapatkan SIM")

1. **If-Else Statement:**
   * Digunakan untuk menangani dua kemungkinan kondisi. Jika kondisi benar, jalankan blok pertama; jika salah, jalankan blok kedua.
2. **If-Elif-Else Statement:**
   * Digunakan untuk menangani lebih dari dua kondisi.

nilai = 85

if nilai >= 90:

print("A")

elif nilai >= 75:

print("B")

elif nilai >= 60:

print("C")

else:

print("D")

1. **Nested If:**
   * Struktur if yang berada di dalam struktur if lain, digunakan untuk mengecek beberapa kondisi secara bertingkat.

umur = 16

genre\_film = "horor"

if umur >= 17:

if genre\_film == "horor":

print("Boleh menonton film horor")

else:

print("Boleh menonton film umum")

else:

if genre\_film == "horor":

print("Tidak boleh menonton film horor")

else:

print("Boleh menonton film umum")

✅ 2. Struktur Perulangan (Looping Structure)

Struktur perulangan digunakan untuk mengulang eksekusi blok kode selama kondisi tertentu terpenuhi atau hingga jumlah iterasi yang telah ditentukan selesai. Ini sangat berguna untuk melakukan tugas-tugas yang berulang secara efisien tanpa perlu menulis kode berulang kali.

Misalnya, jika kita ingin mencetak angka dari 1 hingga 5, kita dapat menggunakan perulangan daripada menulis perintah print() lima kali.

Contoh Sederhana:

for i in range(1, 6):

print(i)

Pada contoh di atas, perintah print(i) akan dieksekusi sebanyak lima kali dengan nilai i mulai dari 1 hingga 5.

Jenis-Jenis Struktur Perulangan:

For Loop:

Digunakan untuk mengulang blok kode dengan jumlah iterasi tertentu.

Biasanya digunakan untuk iterasi pada daftar (list), string, atau range angka.

Contoh:

for i in range(3):

print("Hello")

**output**

Hello

Hello

Hello

**While Loop:**

* Melakukan perulangan selama kondisi tertentu bernilai **true**.
* Jika kondisi tidak pernah menjadi false, maka akan terjadi **infinite loop** (perulangan tak terbatas).

**Contoh:**

count = 1

while count <= 3:

print("Perulangan ke-", count)

count += 1

**Nested Loop:**

* Perulangan di dalam perulangan lainnya.
* Cocok untuk mengakses struktur data dua dimensi seperti matriks atau tabel.

**Contoh:**

for i in range(1, 3):

for j in range(1, 3):

print(f"i: {i}, j: {j}")

output

i: 1, j: 1

i: 1, j: 2

i: 2, j: 1

i: 2, j: 2

**✅ 3. Kontrol Perulangan (Loop Control Statements)**

Untuk mengontrol alur perulangan, terdapat beberapa pernyataan khusus, yaitu:

1. **Break:**
   * Menghentikan perulangan sebelum mencapai batas akhir.

**Contoh:**

for i in range(1, 6):

if i == 3:

break

print(i)

outputnya

1

2

1. **Continue:**

* Melompati iterasi tertentu dan **melanjutkan ke iterasi berikutnya**.

**Contoh:**

for i in range(1, 6):

if i == 3:

continue

print(i)

output

1

2

4

5

1. **Pass:**

* Tidak melakukan apa-apa. Biasanya digunakan sebagai placeholder.

Contoh

for i in range(1, 4):

if i == 2:

pass

print(i)

output

1

2

3

**✅ Kesimpulan:**

* **Struktur kontrol kondisional** memungkinkan program untuk **mengambil keputusan** berdasarkan kondisi tertentu, seperti if, if-else, dan if-elif-else.
* **Struktur perulangan** memungkinkan program untuk **mengulang eksekusi kode** selama kondisi tertentu terpenuhi, seperti for dan while.
* **Kontrol perulangan** (break, continue, pass) digunakan untuk **mengontrol alur perulangan** agar lebih fleksibel.

Dengan memahami kedua struktur ini, seorang programmer dapat membuat program yang lebih dinamis, efisien, dan mampu menangani berbagai kondisi secara efektif.