**LAPORAN PRAKTIKUM  
PEMROGRAMAN 1**

**MODUL 4**

Sebuah gambar berisi logo

Deskripsi dibuat secara otomatis

Oleh:

SATRIA RAMADHAN

2311104026

S1SE07A

**PRODI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

1. **DASAR TEORI**

Dalam pembuatan program, terkadang kita harus melakukan pengulangan suatu  aksi misalnya untuk melakukan perhitungan berulang dengan menggunakan formula  yang sama. Sebagai contoh, misalnya kita ingin membuat program yang dapat  menampilkan sebuah teks “Saya sedang belajar python” sebanyak 5 kali, maka kita tidak  perlu untuk menuliskan 5 buah statement melainkan kita hanya tinggal menempatkan  satu buah statement ke dalam suatu struktur pengulangan. Dengan demikian program  kita akan lebih efisien.

Sebagai gambaran bagi anda untuk dapat lebih memahami konsep perulangan, coba  perhatikan kode program di bawah ini :

print('Saya sedang belajar python')

print('Saya sedang belajar python')

print('Saya sedang belajar python')

print('Saya sedang belajar python')

print('Saya sedang belajar python')

1. **STRUKTUR FOR**

Struktur for ini digunakan untuk menuliskan jenis perulangan yang banyaknya sudah pasti  atau telah diketahui sebelumnya. Oleh karena itu, disini kita harus melakukan inisialisasi  nilai untuk kondisi awal pengulangan dan juga harus menuliskan kondisi untuk  mengentikan proses pengulangan. Adapun bentuk umum dari pendefinisian struktur for  untuk satu statement dalam bahasa java adalah sebagai berikut :

for(ekpresi1; ekpresi2; ekpresi3)

statement;

Sedangkan for untuk dua statement atau lebih dalam bahasa java adalah sebagai berikut :

for(ekpresi1; ekpresi2; ekpresi3)

{

statement1;

statement2;

................

}

Keterangan :

**Ekspresi 1 →** digunakan sebagai proses inisialisasi variable yang akan dijadikansebagai  pencacah(counter) dari proses pengulangan, dengan kata lain ekspresi ini akan dijadikan  sebagai kondisi awal.

**Ekspresi 2 →** digunakan sebagai kondisi akhir, yaitu kondisi dimana proses pengulangan  harus dihentikan. Perlu untuk diketahui bahwa pengulangan masih akan dilakukan selama  kondisi akhir bernilai benar.

**Ekspresi 3 →** digunakan untuk menaikkan (increment) atau menurunkan(decremenet)  nilai variable yang digunakan sebagai pencacah. Apabila pengulangan yang kita lakukan  bersifat menaik, maka kita akan menggunakan statement increment, sedangkan apabila  pengulangan yang akan kita gunakan bersifat menurun maka kita harus menggunakan  statement decrement.

Secara umum contoh untuk mengilustrasikan struktur pengulangan for adalah sebagai  berikut:

for(int i=0; i<5; i++)

{

statement yang akan di ulang

……………………………………..

}

Pada sintaks di atas, mula – mula kita menginisialisasi variable i dengan nilai 0, kemudian  karena ekspresi (0<5) bernilai benar maka program akan melakukan statement untuk  pertama kalinya. Setelah itu variable i akan dinaikkan nilainya sebesar 1 melalui  statement increment i++ sehingga nilai i sekarang menjadi 1. Sampai disni program akan  mengecek ekspres (i<5). Oleh karena ekspresi (2<5) bernilai benar, maka program akan  melakukan

1. **STRUKTUR WHILE**

Pada struktur pengulangan jenis ini kondisi akan diperiksa di bagian awal. Hal ini tentu  menyebabkan kemungkinan bahwa apabila ternyata kondisi yang kita definisikan tidak  terpenuhi (bernilai salah), maka proses pengulangan pun tidak akan pernah dilakukan. Bentuk umum dari struktur while :

while(ekspresi){

statement yang akan di ulang 1;

statement yang akan di ulang 1;

.........................

}

Sebagai pembanding dengan struktur pengulangan for diatas, maka disini dituliskan  kembali program yang akan menampilkan teks “Saya sedang belajar Python” dengan  menggunakan struktur while. Adapun sintaksnya adalah sebagai berikut.

i = 0

while i <= 5:

print('Saya sedang belajar Python')

i+=1

1. **STRUKTUR DO-WHILE**

Berbeda dengan struktur while dimana kondisinya terletak di awal blok pengulangan,  pada struktur do-while kondisi diletakkan di akhir blok pengulangan. Hal ini menyebabkan  bahwa statement yang terdapat di dalam blok pengulangan ini pasti akan dieksekusi  minimal satu kali, walaupun kondisinya bernilai salah sekalipun. Maka dari itu struktur do while ini banyak digunakan untuk kasus – kasus pengulangan yang tidak mempedulikan  benar atau salahnya kondisi pada saat memulai proses pengulangan. Adapaun bentuk  umum dari struktur pengulangan do- while adalah seperti yang tertulis di bawah ini :

do{

statement yang akan di ulang;

..............

} while(ekspresi)

Mungkin bagi anda yang baru mengenal bahasa pemrograman Python akan merasa  bingung untuk membedakan struktur pengulangan while dan do-while. Dimana cara  penggunaan do-while pada python sedikit berbeda dengan bahasa pemrograman lain.  Untuk memahami perbedaanya perhatikan dua buah contoh program berikut ini.

Kode program pada javascript:

var quota = 9;

// perulangan while-do

while(quota > 0){

console.log("Masih ada quota, bisa");

quota--;

}

// perulangan do-while

do{

console.log("Quota habis!");

quota--;

} while(quota > 0)

**Kapankah Waktu yang Tepat Menggunakan while dan do/while?**

Tergantung dari kasusnya. Bila syarat perulangannya tidak berkaitan dengan hasil hitung  pada blok kode yang diulang, maka pakailah while. Tetapi, bila syarat perulangannya  berkaitan dengan hasil perhitungan di blok kode yang diulang, maka pakailah do/while.

1. **GUIDED**

**PERULANGAN**

1. **Perulangan Menggunakan For**
2. Penggunaan for

for i in range (5):

    print("Hello World")

program di atas akan mengoutputkan sebuah text “Hello World” sebanyak 5 kali

1. Pejumlahan

count = 5

for i in range (10):

    print(count)

    count += 5

Program di atas akan mengoutputkan hasil dari pejumlahan dari variable count + 5 sebanyak 10 kali sesuai dengan perulangannya.

1. Range pada for

for i in range (2, 10):

    print(i)

For selalu dimulai dari 0, kita dapat mengatur mau mulai dari mana menggunakan range. Misalnya pada program diatas. Ini akan membuat I memulai perulangan dari 2, 3, 4, 5, 6…

1. **Perulangan Menggunakan While**
2. Penggunaan while

angka = 4

while angka < 10:

    print(angka)

perulangan ini tidak akan berhenti karena suatu kondisi selalu tercapai / bernilai true

1. Membuat break pada while

quota = -1

while(True):

    print("MEWING")

    quota = quota-1

    if  (quota < 0):

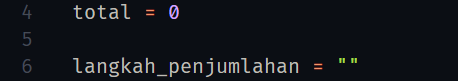
        break

kondisi pertama adalah True, maka perulangan ini akan selalu berjalan, disini kita membuat sebuah kondisi menggunakan IF lalu memberikan break. Perulangan ini akan berhenti Ketika kondisi IF tercapai.

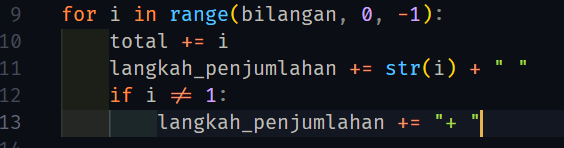
1. **UNGUIDED**
2. **Menghitung total nilai dari suatu bilangan yang diinputkan oleh user**
3. Buat input an dari user



1. Buat sebuah variable yang berguna untuk menyimpan hasil dari pejumlahan dan buat juga variable yang akan berguna untuk menyimpan Langkah-langkah dari perhitungan yang akan di format menjadi string.



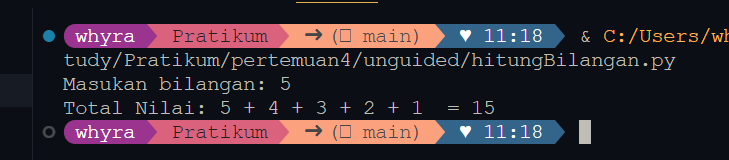
1. Buat sebuah perulangan dengan nilai range dari inputan user



1. Lalu print variable Langkah\_pejumlahan dan total



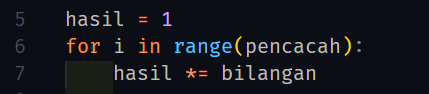
1. Jalankan program dan isi bilangan



1. Program Pencacah
2. Siapkan variable bilangan dan pencacah dengan berisikan inputan dari user



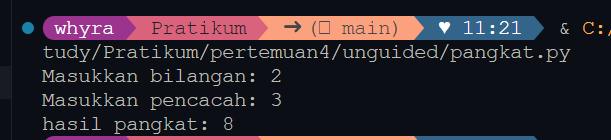
1. Buat variable untuk melakukan perkalian yang dimulai dari 1 dan buat looping untuk loop perkalian sampai range pencacah



1. Print variable hasil



1. Jalankan program



1. Program KPK
2. Buat 2 variable yang berisikan inputan dari user



1. Tentukan bilangan yang paling besar dari variable inputan user



1. Buat looping untuk menentukan KPK



1. Print hasil kpk dan berikan break pada akhir line untuk menghentikan looping



1. Jalankan programnya

