|  |  |  |
| --- | --- | --- |
| **Nama:**  **Fariz Ubaidillah**  **NIM: 065002100016** | C:\Users\RPL-SI 02\Pictures\288px-Trisakti_Logo.svg.png | **MODUL 3**  **Nama Dosen:**  **Anung B. Ariwibowo, M. Kom** |
| **Hari/Tanggal:**  **Minggu,10-10-2021** | **Praktikum Algoritma & Pemrograman** | **Nama Asisten Labratorium:**   1. **Azhar Rizki Zulma 065001900001** 2. **Nadiya Amanda Rizkania 064001900003** |

**Struktur Kendali (Control Structure)**

1. **Teori Singkat**

Ekspresi Boolean

Eskpresi Boolean merupakan ekspresi yang mengembalikan nilai True atau False, menggunakan operator relasional/operator perbandingan, dan juga operator logika. Selain itu Ekspresi Boolean juga dapat menggunakan operator keanggotaan (*membership operator)* dan juga operator identitas dalam beberapa kasus.

Operator Perbandingan

Operator Perbandingan adalah operator yang melakukan perbandingan antara dua buah nilai. Operator ini juga dikenal dengan operator relasional dan sering digunakan untuk membuat sebuah logika atau kondisi. Berikut ini adalah daftar Operator Aritmatika dalam Python:

|  |  |
| --- | --- |
| **Operator** | **Simbol** |
| Lebih Besar | > |
| Lebih Kecil | < |
| Sama Dengan | == |
| Tidak Sama Dengan | != |
| Lebih Besar Sama Dengan | >= |
| Lebih Kecil Sama Dengan | <= |

Operator Logika

Operator Logika merupakan sebuah operator yang digunakan untuk membuat logika dalam program yang kita buat. Operator logika juga sering disebut juga sebagai Operator Aljabar Boolean, biasanya operator logika ini digunakan untuk membuat operasi percabangan pada program. Operator Logika diantaranya seperti logika AND, OR, dan NOT.

Operator logika terdiri dari:

|  |  |
| --- | --- |
| **Operator** | **Simbol** |
| Logika AND | and |
| Logika OR | or |
| Logika Negasi/Kebalikan | not |

Konstruksi Percabangan & Blok Program

Konstruksi Percabangan adalah sebuah program yang ketika dijalankan akan menimbulkan percabangan kedalam sub cabangnya yang berisi sebuah blok program sesuai dengan kondisi dan logika yang diminta. Umumnya kostruksi percabangan dalam Bahasa pemrograman Python sendiri dapat dibuat dengan memanggil keyword *if/elif/else.* Berikut tabelnya

|  |  |
| --- | --- |
| **Keterangan** | **Keyword** |
| Terdapat 1 pilihan keputusan | if |
| Terdapat 2 pilihan keputusan | if/else |
| Terdapat lebih dari 2 pilihan keputusan | if/elif/else |

Blok program berisi sekumpulan ekpresi dan statement untuk dikerjakan oleh komputer. Dalam Bahasa pemrograman Python blok program sendiri dapat diidentifikasikan dengan tanda *colon* (“:”) setelah pendeklarasian konstruksi *if/elif/else*, *for*, *while* ataupun ketika melakukan definisi fungsi.

Blok program yang terdapat pada kondisi *if* sendiri akan dijalankan jika kondisi yang diminta bernilai *true.*

Blok program yang terdapat pada kondisi kondisi *elif* sendiri yang merupakan kepanjangan dari else if yang berarti jika tidak sesuai dengan kondisi sebelumnya maka akan disesuaikan dengan kondisi lainnya yang dapat bernilai *true.*

Blok program yang terdapat pada kondisi *else* akan dijalankan ketika nilai dari kondisi sebelumnya yaitu *if/elif* bernilai *false.*

Berikut ini adalah contoh sederhana program konstruksi percabangan yang menggunakan operator perbandingan:

Source Code

|  |
| --- |
|  |

Output

|  |
| --- |
|  |

1. **Alat dan Bahan**

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

1. **Elemen Kompetensi**
   1. Latihan pertama

Sebuah segitiga dibangun dari tiga garis lurus. Berdasarkan panjang dari sisi-sisinya, segitiga dapat dibedakan menjadi tiga jenis. Ada segitiga sama sisi, segitiga sama kaki, segitiga siku-siku, atau segitiga sembarang Buatlah sebuah program yang menerima tiga bilangan yang merupakan panjang dari sisi-sisi sebuah segitiga. Berdasarkan panjang yang diberikan, program anda akan mencetak jenis segitiganya (sama sisi, sama kaki, atau sembarang). Hati-hati: Tidak semua kombinasi tiga bilangan dapat membentuk segitiga. Contoh: 1, 2, 3 tidak mungkin membentuk segitiga.

Source Code

|  |
| --- |
| print(“Isi dengan Source Code kalian ya…”)  """  Created on Mon Oct 4 13:16:09 2021  @NIM : 065.00.21.00.016  @author: FARIZ UBAIDILLAH  """  print ("membuat program mengetahui jenis segitiga")  A = float(input("masukan sisi A = "))  B = float(input("masukan sisi B = "))  C = float(input("masukan sisi C = "))  if (A==B==C) :  print("merupakan jenis segitiga sama sisi ")  elif (A==B or B==C or C==A) :  print ("merupakan jenis segitiga samakaki ")  elif ((A + B <=C) or (A + C <=B) or (B + C <= A)) :  print ("bukan merupakan segitiga ")  else :  print("merupakan segitiga sembarang ") |

Output

|  |
| --- |
|  |

* 1. Latihan Kedua

Buatlah program untuk mencari Akar Persamaan Kuadrat dan Determinan

Source Code

|  |
| --- |
| import math  A = float(input("Masukkan Nilai A = "))  B = float(input("Masukkan Nilai B = "))  C = float(input("Masukkan Nilai C = "))  D = (B\*\*2) - (4\*A\*C)  if (A==0) :  print ("Bukan persamaan kuadrat")  elif (D==0) :  x1 = (-B/(2\*A))  print("persamaan kuadratnya adalah = ",A ,"x^2",+B,"x",+C)  print("Determinannya adalah =" ,D)  print("Akar = ",x1)  elif (D<0) :  print("persamaan kuadratnya adalah = ",A ,"x^2",+B,"x",+C)  print("Deeterminan nya adalah = ",D)  print(-B,"+Akar",D,"/2\*",A)  print(-B,"-Akar",D,"/2\*",A)  elif (D>0) :  x1 = (-B+math.sqrt(D))/(2\*A)  x2 = (-B-math.sqrt(D))/(2\*A)  print("Persamaan kuadratnya adalah = ",A ,"x^2",+B,"x",+C)  print("Determinan nya adalah = ",D)  print("Akar x1 = ",x1)  print("Akar x2 = ",x2) |

Output

|  |
| --- |
|  |

1. **Soal Latihan**

Soal:

1. Dalam sebuah kasus program, terdapat sebuah kondisi percabangan *if/else*. Jika program yang dijalankan pada kondisi *if* tidak sesuai dengan kondisinya, maka itu akan menghasilkan status nilai *false* pada percabangan *if* tersebut, dan program tersebut akan masuk ke kondisi *else*, apakah status yang diberikan kondisi *else* tersebut? Jelaskan dan berikan alasannya serta deskripsikan kelanjutan dari program tersebut!
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. Fungsi else digunakan Ketika fungsi IF tidak terpenuhi,di sini fungsi ELSE sebagai pengganti dari IF,namun kita bisa saja tidak menggunakan ELSE jika program tersebut tidak memungkinkan menggunakan fungsi ELSE maka kita dapat menggunakan fungsi ELIF yang mana merupakan percabangan yang tidak hanya menentukan 1 kondisi tetapi dapat menentukan suatu kondisi yang tidak sesuai
2. Pertama kita harus menginput suata key word yaitu import math yang mana nanti nya kita akan menggunakan suatu rumus,kemudian kita memasukan nilai masing” variable guna mengetahui hasil akhirnya.Lalu masukan rumus D = (B\*\*2) - (4\*A\*C),yaitu guna mengetahui nilai mana saja yang akan masuk dalam kriteria yang kita inginkan,kita juga menggunakan keyword if dan elif karna yang kita jalankan lebih dari 1.setelah kita menyelesaikan source kode nya,kemudian kita Run guna mengetahui apakah program yang kita buat sudah benar atau tidak.
3. **Kesimpulan**
   1. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
   2. Kita dapat mengetahui fungsi lebih jauh dari pemograman python,dan kita sekali lagi diwajibkan lebih teliti dalam menginputkan suatu fungsi agar Ketika di Run tidak terjadi eror atau keluaran dari hasil yang kita inputkan tidak dapat terbaca oleh program.
4. **Cek List (✓)**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Elemen Kompetensi** | **Penyelesaian** | |
| **Selesai** | **Tidak Selesai** |
| **1.** | Latihan Pertama | **(✓)** |  |
| **2.** | Latihan Kedua | **(✓)** |  |

1. **Formulir Umpan Balik**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Elemen Kompetensi** | **Waktu Pengerjaan** | **Kriteria** |
| **1.** | Latihan Pertama | 45 Menit | Kita diharuskan membuat program membedakan 3 jenis segitiga |
| **2.** | Latihan Kedua | 60 Menit | Membuat program mencari akar persamaan kuadrat dan determinan |

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang