

LAPORAN RESMI
MODUL 3
LOOPING (PERULANGAN)
ALGORITMA PEMROGRAMAN



NAMA	: IKTIAR RAMADANI
N.R.P	: 230441100053
DOSEN	: MOHAMMAD SYARIEF, ST., M.Cs
ASISTEN	: KUKUH COKRO WIBOWO
TGL PRAKTIKUM	: 06 OKTOBER 2023

Disetujui : 10 Oktober 2023
Asisten

KUKUH COKRO WIBOWO
21.04.411.00102



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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perulangan atau Looping adalah sebuah proses program dimana setiap perintah/ instruksi dapat dijalankan secara berulang dengan kondisi tertentu (Wolfman, 2013). Pengulangan yaitu satu atau lebih pernyataan yang akan diulang sesuai dengan jumlah tertentu yang diinginkan. Ada beberapa statement atau pernyataan dalam sistem perulangan yaitu Statement For, Statement While dan Statement Do-While (Latif et al., 2018). Dalam pemrograman, looping harus berhenti pada suatu keadaan tertentu, jika tidak maka akan terjadi hang pada komputer karena mengerjakan sesuatu yang tidak pernah berhenti (Basuki, 2019). Hang yaitu suatu sistem pada komputer yang tidak dapat merespon/ menanggapi segala perintah yang telah dimasukkan, hal ini tentunya dapat mengarah pada kerusakan komputer.

Untuk mencacah berapa kali perulangan dilakukan, diperlukan suatu variabel pencacah yang bertipe integer. Variabel tersebut akan bertambah nilainya setiap kali perulangan dilakukan. Konstruksi while digunakan untuk melakukan perulangan terhadap baris kode tertentu selama suatu kondisi terpenuhi. Jika kondisi sudah tidak terpenuhi, maka program akan keluar dari perulangan. Konstruksi repeat digunakan untuk melakukan perulangan sampai suatu kondisi terpenuhi. Hampir setiap program mutlak memerlukan suatu perulangan dan percabangan. Tujuan perulangan disini adalah untuk mengulang statement berulang kali sesuai jumlah yang ditentukan pemakai. Program perulangan terbagi atas beberapa bagian yang digunakan sesuai dengan kebutuhan. Setiap perulangan akan memberikan output sesuai dengan algoritma yang dikehendaki.

Dengan Python, Anda dapat menggunakan perulangan while untuk menjalankan tugas yang sama beberapa kali dan perulangan for untuk mengulang sekali di atas data daftar. Dalam modul ini, Anda akan mempelajari tentang dua jenis perulangan dan kapan harus menerapkan masing-masing dari perulangan tersebut. Perulangan atau disebut Looping adalah bahasa pemrograman dan algoritma yang kegunaannya untuk mengulang sebuah perintah/intruksi yang

dibuat dalam script sesuai dengan jumlah yang telah ditentukan. Perulangan bertujuan untuk mempersingkat waktu pernyataan program yang harus ditulis dalam jumlah yang banyak.

1.2 Tujuan

- Mampu memahami apa yang dimaksud dengan perulangan atau looping
- Mampu menjelaskan macam-macam jenis perulangan
- Mampu membuat contoh dengan menggunakan rumus perulangan

BAB II

DASAR TEORI

2.1 Pengertian Perulangan Atau Looping

Perintah perulangan adalah perintah yang digunakan untuk mengulang pengeksekusian statemen-statemen hingga berkali-kali agar sesuai dengan iterasi yang diinginkan pembuat. Dalam bahasa python, perintah untuk pengulangan atau loop ada dua macam yaitu while dan for. Selain itu, ada juga perintah lain seperti break, continue, dan pass. Perulangan bertujuan untuk mempersingkat waktu pernyataan yang harus ditulis dalam jumlah banyak.

2.2 Macam-macam Bentuk Perulangan

- a) Statemen while : Perintah while pada python merupakan perintah yang paling umum digunakan untuk proses iterasi. Konsep sederhana dari perintah while adalah ia akan mengulang mengeksekusi statemen dalam blok while selama nilai kondisinya benar. Dan ia akan keluar atau tidak melakukan eksekusi blok statemen jika nilai kondisinya salah.
- b) Statemen for : Perintah for dalam python mempunyai ciri khas tersendiri dibandingkan dengan bahasa pemrograman lain. Tidak hanya mengulang bilangan-bilangan sebuah ekspresi aritmatik, atau memberikan keleluasaan dalam mendefinisikan iterasi perulangan dan menghentikan perulangan pada saat kondisi tertentu. Dalam python, statemen for bekerja mengulang berbagai macam tipe data sekuensial seperti List, String, dan Tuple.
- c) Statemen continue : Statemen continue menyebabkan alur program kembali ke perintah looping. Jadi jika dalam sebuah perulangan terdapat statemen continue, maka program akan kembali ke perintah looping untuk iterasi selanjutnya.
- d) Statemen break : Perintah break digunakan untuk menghentikan jalannya proses iterasi pada statemen for atau while. Statemen yang berada di bawah break tidak akan di eksekusi dan program akan keluar dari proses looping.

- e) Statemen pass: Statemen pass mengakibatkan program tidak melakukan tindakan apa-apa. Perintah pass biasanya digunakan untuk mengabaikan suatu blok statemen perulangan, pengkondisian, class, dan fungsi yang belum didefinisikan badan programnya agar tidak terjadi error ketika proses kompilasi.

BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Jelaskan pengertian dan fungsi dari continue, break dan perintah pass
2. buatlah contoh soal sederhana dengan menggunakan perulangan for dan while.
3. Ada berapa macam bentuk perulangan, apa bedanya?

3.2 Jawaban

1. a. break adalah berhenti, jadi jika ada perintah break, maka program akan menghentikan jalannya dan tidak akan melanjutkan.
b. continue adalah lanjut, jadi jika ada perintah continue, maka program akan melanjutkan jalannya sesuai dengan perintah awal yang telah tersedia.
c. pass adalah perintah yang memerintahkan program untuk tidak melakukan apapun, digunakan untuk mengabaikan suatu blok statemen dan fungsi yang belum didefinisikan

2. for :	while
angka = 5	i = 1
for i in range(1, 6):	while i <= 5:
print("x", end=" ")	print(i)
	i += 1
output :	output
x x x x x	1
	2
	3
	4
	5

3. Ada 2 macam yaitu while dan for, perbedaan adalah jika while digunakan untuk melakukan pengulangan selama kondisi itu benar, sedangkan for digunakan untuk melakukan iterasi terhadap urutan seperti list, string, dan tuple

BAB IV

IMPLEMENTASI

4.1 Membuat Program Penyeleksian dan Menyatakan Umur

4.1.1 Source Code

Buatlah program dengan bentuk angka NIM terakhir kalian!

```
user = input("Masukkan NIM anda: ")

for angka in user:
    if angka == '0':
        print("*****")
        print("*      *")
        print("*      *")
        print("*      *")
        print("*      *")
        print("*      *")
        print("*****")
        print("")
    elif angka == '1':
        print("  *  ")
        print("* * ")
        print("  *  ")
        print("  *  ")
        print("  *  ")
        print("  *  ")
        print("*****")
        print("")
    elif angka == '2':
        print("*****")
        print("      *")
        print("      *")
        print("*****")
        print("*      ")
        print("*      ")
        print("*****")
        print("")
    elif angka == '3':
        print("*****")
        print("      *")
        print("      *")
        print("*****")
        print("      *")
        print("      *")
        print("*****")
```

```

        print("")
    elif angka == '4':
        print("*      *")
        print("*      *")
        print("*      *")
        print("*****")
        print("      *")
        print("      *")
        print("      *")
        print("")
    elif angka == '5':
        print("*****")
        print("*      ")
        print("*      ")
        print("*****")
        print("      *")
        print("      *")
        print("*****")
        print("")
    elif angka == '6':
        print("*****")
        print("*      ")
        print("*      ")
        print("*****")
        print("*      *")
        print("*      *")
        print("*****")
        print("")
    elif angka == '7':
        print("*****")
        print("      *")
        print("      *")
        print("      *")
        print("      *")
        print("      *")
        print("      *")
        print("")
    elif angka == '8':
        print("*****")
        print("*      *")
        print("*      *")
        print("*****")
        print("*      *")
        print("*      *")
        print("*****")
        print("")
    elif angka == '9':

```



```

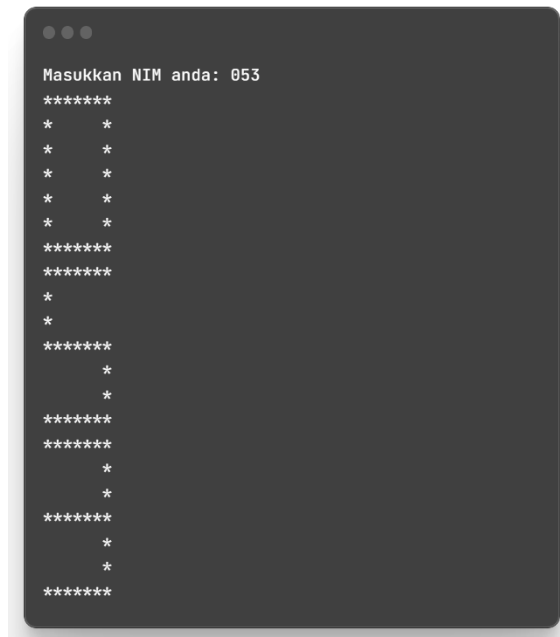
print("*****")
print("*      *")
print("*      *")
print("*****")
print("      *")
print("      *")
print("*****")
print("")
else :
    print( angka, " Tidak termasuk angka, cobalah menggunakan
angka atau gunakan angka lebih dari 0")

```

4.1.2 Penjelasan

User diminta memasukkan NIM nya masing-masing berupa angka. Setelah itu, program akan mencetak angka yang dimasukkan berupa gambar atau pola berbentuk angka. Program akan mengecek data yang telah dimasukkan tersebut dengan menggunakan rumus *for*, yang biasa digunakan apabila data yang dimasukkan tersebut berupa data list, tuple, string dan lainnya. Pada program diatas, saya menggunakan variabel bebas berupa *angka* untuk menyatakan rumus selanjutnya. Saya juga menggunakan fungsi *if*, *elif*, dan *else* agar lebih mudah mencetak pola. Pada program saya mencetak angka pokok dari 0 sampai 9. Ketika user memasukkan data berupa nomor, maka program akan mencetak sesuai dengan data yang dimasukkan. Karena data yang dimasukkan berupa bentuk list, maka program akan menggunakan *range* sebagai patokan. Jadi, jika user memasukkan nomor semisal 053, maka nomor tersebut memiliki range per kata sebanyak 3, begitupun jika kita memasukkan banyak angka, program akan menghitung range yang ada lalu mencetaknya, sehingga akan muncul hasil yang sesuai dengan data tersebut. Namun, apabila user memasukkan data selain angka, komputer akan membacanya bernilai salah dan akan mencetak hasil bahwa data yang dimasukkan salah.

4.1.3 Hasil



```
Masukkan NIM anda: 053
*****
*   *
*   *
*   *
*   *
*   *
*****
*****
*
*
*****
*
*
*****
*
*
*****
*
*
*****
*
*
*****
*****
```

4.2. Membuat Program Perulangan Bilangan Prima

4.2.1. Source Code

Buatlah program perulangan bilangan prima!

```
print("===== MENCARI BILANGAN PRIMA =====")
print("")

while True:
    user = int(input("Masukkan angka awal : "))

    if user <= 1:
        print("Angka awal harus lebih dari 1")
        ulangi = input("Apakah ingin melanjutkan? (y/t): ")
        if ulangi.lower() != "y":
            break
        else:
            continue
    batas = int(input("Masukkan angka akhir : "))

    while user <= batas:
        if user == 2 or user == 3 or user == 5 or user == 7:
            print(user, end=" ")
        elif user % 2 != 0 and user % 3 != 0 and user % 5 != 0 and
user % 7 != 0:
            print(user, end=" ")
        user += 1

    ulangi = input("\nApakah ingin mengulang pertanyaan lagi? (y/t): ")
    )
```

```
if ulangi.lower() != "y":  
    break  
  
print("Terima kasih telah menggunakan program ini")
```

4.2.2. Penjelasan

User diminta untuk memasukkan angka awal terlebih dahulu. Untuk angka awal harus lebih dari satu, jika memasukkan angka kurang dari atau sama dengan satu maka program tidak akan melanjutkan dan menanyakan apakah ingin melanjutkan. Namun jika benar maka program akan melanjutkan dengan menyuruh untuk memberikan batas akhirnya, setelah itu user melakukan run dan akan tampil jawaban berupa bilangan prima yang benar diantara bilangan awal hingga bilangan akhir yang dimasukkan. Pada program diatas, pertama-tama user memasukkan 2 data yang akan dicari. Data tersebut berupa nilai awal, nilai yang digunakan sebagai acuan mulainya penghitungan dan data yang kedua berupa nilai batas yang merupakan batasan dari nilai yang akan dihitung. Hal ini dimaksudkan agar user dapat menentukan data yang memang ingin dicari. User dapat memasukkan angka berapapun karena program ini dibuat tanpa batasan, namun dengan syarat nilai awal harus lebih dari 1. Pada program ini, saya menggunakan looping berupa while, yang digunakan agar selama kondisi benar, maka program akan terus berjalan, namun apabila kondisi bernilai salah maka program akan berhenti. Disini saya juga menggunakan program ulangi, yang berfungsi apabila user memasukkan data yang salah, program akan menanyakan apakah program tersebut akan dilanjutkan atau tidak. Apabila user memilih untuk melanjutkan, program akan mengulang pertanyaan tersebut dari depan lagi, dan begitu seterusnya sampai user memilih untuk berhenti, maka program akan menghentikannya dan menampilkan output terima kasih. Untuk penghitungannya menggunakan rumus yang sudah ada, jadi untuk bilangan prima adalah bilangan yang ketika dibagi hanya dapat dibagi oleh 1 dan dirinya sendiri. Pada program, menggunakan acuan dari nilai awal yaitu 2, 3, 5 dan 7. Untuk seterusnya menggunakan pembagian dari angka-angka tersebut. Agar nilai yang dihasilkan menampilkan hasil secara horizontal maka program ditambah rumus python berupa `end=" "` agar hasilnya horizontal.

4.2.3 Hasil

```
===== MENCARI BILANGAN PRIMA =====

(Syarat: Angka awal harus lebih dari 1)

Masukkan angka awal : 2
Masukkan angka akhir : 10
2 3 5 7
```

4.3 Membuat Program Penghitungan Denda

4.3.1. Source Code

Sebuah perpustakaan melayani peminjaman dan pengembalian buku. Setiap buku hanya boleh dipinjam selama 1 minggu (7 hari). Jika lebih dari batas waktu tersebut, maka petugas perpustakaan akan memberikan denda sebanyak Rp2000 per harinya. Petugas akan memberikan denda tambahan sebesar Rp5000 setiap minggunya jika buku belum juga dikembalikan. Bantulah petugas perpustakaan untuk membuat program penghitungan denda tersebut dengan input berupa lama peminjaman buku (dalam satuan hari) dan output denda yang harus dibayar. Setelah program selesai dijalankan, tampilkan pertanyaan menghitung lagi atau tidak. Jika dijawab "ya", maka program akan mengulangi dari awal.

```
print("===== MENGHITUNG DENDA =====")
print("")
def hitung_denda(lama_meminjam) :
    dendahari = 2000
    dendaminggu = 5000
    total = 0

    if lama_meminjam > 7:
        total += (lama_meminjam - 7) * dendahari

    if lama_meminjam > 7:
        total += (lama_meminjam - 7) // 7 * dendaminggu

    return total

while True:
    lama_meminjam = int(input("Masukkan lama peminjaman buku (dalam hari): "))
```

```

if lama_meminjam <= 1:
    print("salah, Harus memasukkan minimal 1 atau lebih")
elif 1 <= lama_meminjam <= 7:
    print("Anda tidak perlu membayar denda karena meminjam
kurang dari 7 hari.")

else:
    denda = hitung_denda(lama_meminjam)
    print("Denda yang harus dibayar adalah Rp", denda)

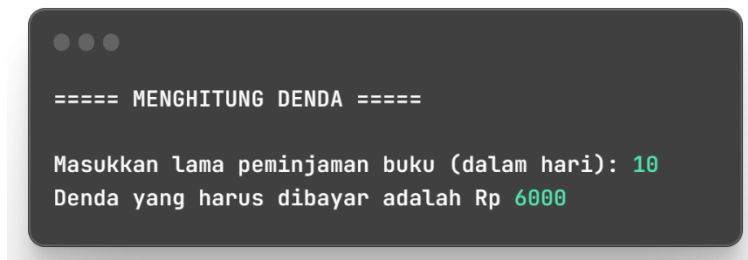
ulangi = input("Apakah ingin menghitung lagi? (y/t): ")
if ulangi.lower() != "y":
    break

```

4.3.2. Penjelasan

Pada program kali ini user diminta untuk memasukkan berapa hari buku yang dipinjam. Pada saat user memasukkan angka 0 atau kurang, program akan menampilkan hasil salah. Pada saat memasukkan angka sampai 7 maka hasilnya user tidak perlu membayar denda. Dan saat memasukkan angka lebih dari 7 maka program akan menghitungnya dan akan menampilkan hasil perhitungannya. Pada program ini, menggunakan fungsi berupa *def* atau definition yang berfungsi agar fungsi tersebut dapat dipanggil berulang-ulang dan lebih menghemat rumus agar tidak membuat rumus berulang-ulang. Jadi apabila kita membuat fungsi *def* maka fungsi tersebut berisi parameter yang berisi rumus. Apabila kita membuat program yang panjang, kita hanya menulis fungsi *def* tersebut agar dapat menghitung. Di program tersebut juga menggunakan rumus ulangi yang digunakan apabila user memasukkan data salah, maka program akan berhenti dan menanyakan apakah ingin di ulang atau tidak. Jika ya, maka akan mengulang dari depan, jika tidak akan berhenti.

4.3.3. Hasil



```

==== MENGHITUNG DENDA ====

Masukkan lama peminjaman buku (dalam hari): 10
Denda yang harus dibayar adalah Rp 6000

```

BAB V

PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa dengan menggunakan menggunakan kita bisa menghitung sesuatu tanpa menggunakan cara satu demi satu. Hal ini akan sangat mempermudah karena hanya dengan menggunakan rumus perulangan, kita tidak perlu membuat rumus yang panjang. Pada modul dan pembelajaran ini, kita belajar bahwa dengan menggunakan perulangan berupa while atau for, kita lebih mudah membuat program.

5.2 Kesimpulan

1. Kita jadi bisa mengerti dengan pentingnya penggunaan perulangan di sebuah program agar program tersebut dapat berjalan sesuai dengan apa yang kita mau.
2. Perintah perulangan adalah perintah yang digunakan untuk mengulang pengeksekusian statemen-statemen hingga berkali-kali agar sesuai dengan iterasi yang diinginkan pembuat
3. Ada 2 bentuk perintah perulangan yaitu while dan for. Selain itu ada juga perintah break, continue, dan pass
4. Bisa membuat program dengan menggunakan perintah perulangan