

## Pendahuluan

Dalam pemrograman, pengulangan atau loop adalah salah satu konsep fundamental yang memungkinkan eksekusi berulang dari sekumpulan instruksi atau blok kode. Python, sebagai salah satu bahasa pemrograman yang populer dan digunakan secara luas, menyediakan dua jenis loop utama: for dan while. Kedua jenis loop ini dirancang untuk mempermudah programmer dalam mengotomatiskan tugas-tugas yang repetitif.

## Tujuan dan Manfaat

### Tujuan

1. **Automatisasi Tugas Repetitif:** Salah satu tujuan utama penggunaan loop adalah untuk mengotomatiskan tugas-tugas yang bersifat repetitif. Ini memungkinkan eksekusi berulang dari blok kode tanpa perlu menulis ulang kode tersebut berkali-kali.
2. **Pengolahan Data:** Loop digunakan untuk mengiterasi elemen-elemen dalam struktur data seperti daftar, tuple, dan kamus (dictionary), sehingga memudahkan manipulasi dan pengolahan data.
3. **Efisiensi Kode:** Dengan menggunakan loop, kode menjadi lebih efisien dan lebih mudah dibaca. Loop mengurangi jumlah baris kode yang harus ditulis dan di-maintain.
4. **Penerapan Algoritma:** Banyak algoritma memerlukan pengulangan, seperti pencarian, pengurutan, dan berbagai algoritma matematika. Loop memungkinkan penerapan algoritma-algoritma tersebut secara efektif.
5. **Kontrol Alur Program:** Loop memberikan kontrol yang lebih baik atas alur eksekusi program, memungkinkan program untuk beradaptasi dengan kondisi yang berubah-ubah selama eksekusi.

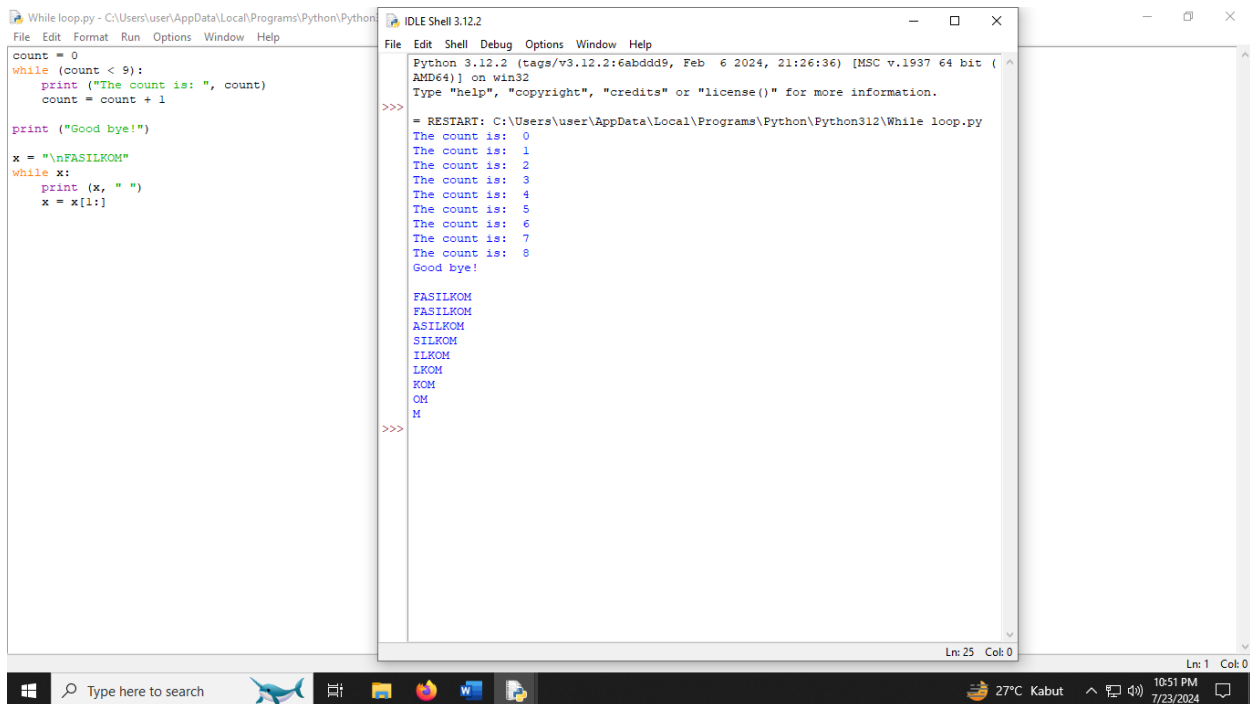
### Manfaat

1. **Menghemat Waktu dan Usaha:** Dengan mengotomatiskan tugas-tugas yang berulang, programmer dapat menghemat waktu dan usaha yang seharusnya diperlukan untuk menulis kode secara manual.
2. **Konsistensi dan Akurasi:** Penggunaan loop memastikan bahwa tugas-tugas yang berulang dilakukan dengan cara yang konsisten dan akurat, mengurangi kemungkinan kesalahan manusia.

3. **Kemudahan Pemeliharaan:** Kode yang menggunakan loop lebih mudah dipelihara dan diperbarui. Jika ada perubahan yang perlu dilakukan, cukup mengubah satu bagian dari loop daripada mengubah banyak baris kode.
4. **Fleksibilitas:** Loop memberikan fleksibilitas untuk mengerjakan berbagai tugas dengan sedikit modifikasi. Misalnya, loop yang sama dapat digunakan untuk mengiterasi berbagai jenis koleksi data.
5. **Penggunaan Sumber Daya yang Efisien:** Loop memungkinkan penggunaan sumber daya komputasi yang lebih efisien, karena loop mengurangi redundansi dan memungkinkan operasi batch.
6. **Pembelajaran dan Pengembangan:** Penggunaan loop membantu programmer memahami konsep dasar pemrograman seperti kondisi, kontrol alur, dan algoritma, yang penting untuk pengembangan keterampilan pemrograman yang lebih lanjut.

## Program

### 1.



The screenshot displays a Python IDE with two windows. The left window, titled 'While loop.py', contains the following code:

```
count = 0
while (count < 9):
    print ("The count is: ", count)
    count = count + 1

print ("Good bye!")

x = "\nFASILKOM"
while x:
    print (x, " ")
    x = x[1:]
```

The right window, titled 'IDLE Shell 3.12.2', shows the execution output:

```
>>>
= RESTART: C:\Users\user\AppData\Local\Programs\Python\Python312\While loop.py
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
Good bye!

FASILKOM
FASILKOM
ASILKOM
SILKOM
ILKOM
LKOM
KOM
OM
M
>>>
```

The Windows taskbar at the bottom shows the system clock as 10:51 PM on 7/23/2024, with a temperature of 27°C in Kabut.

```
For loop.py - D:\pemula\Python\For loop.py (3.12.2)
File Edit Format Run Options Window Help

angka = [1,2,3,4,5]
for x in angka:
    print(x)

buah = ["Nanas", "Apel", "Jeruk"]
for makanan in buah:
    print("Saya suka makan", makanan)

nama = ['budi', 'andi', 'rudi', 'sandi']
usia = [20,18,22,19]
for i in range(len(nama)):
    print(nama[i], 'berusia', usia[i], 'tahun')
```

```
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: D:\pemula\Python\For loop.py
1
2
3
4
5
Saya suka makan Nanas
Saya suka makan Apel
Saya suka makan Jeruk
budi berusia 20 tahun
andi berusia 18 tahun
rudi berusia 22 tahun
sandi berusia 19 tahun
>>>
```

```
Nested loop.py - D:\pemula\Python\Nested loop.py (3.12.2)
File Edit Format Run Options Window Help

i = 2
while(i < 100):
    j = 2
    while(j <= (i/j)):
        if not(i%j): break
        j = j+1
    if(j > i/j): print(i, " is prime")
    i = i+1

print("Good bye!")
```

```
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: D:\pemula\Python\Nested loop.py
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
13 is prime
17 is prime
19 is prime
23 is prime
29 is prime
31 is prime
37 is prime
41 is prime
43 is prime
47 is prime
53 is prime
59 is prime
61 is prime
67 is prime
71 is prime
73 is prime
79 is prime
83 is prime
89 is prime
97 is prime
Good bye!
>>> |
```

2A.

The image shows a Python IDE window titled "Latihan 2a.py" and an "IDLE Shell 3.12.2" window. The script in the IDE calculates the sum of even and odd numbers from 1 to 10. The shell shows the execution output.

```
Latihan 2a.py - D:\pemula\Python\Latihan 2a.py (3.12.2)
File Edit Format Run Options Window Help

1 = list(range(1,10))
print("list angka, ",1)

ganjil = 0
genap = 0

for j in 1:
    if j % 2 ==0:
        genap +=1
    else:
        ganjil +=1

print("Jumlah bilangan genap : ", genap)
print("Jumlah bilangan ganjil : ", ganjil)
```

```
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: D:\pemula\Python\Latihan 2a.py
list angka,  [1, 2, 3, 4, 5, 6, 7, 8, 9]
Jumlah bilangan genap : 4
Jumlah bilangan ganjil : 5
>>> |
```

Ln: 8 Cok: 0

## 2B.

The image shows a Python IDE window titled "Latihan 2b.py" and an "IDLE Shell 3.12.2" window. The script in the IDE takes an input number and calculates the sum of its products with numbers from 1 to 10. The shell shows the execution output.

```
Latihan 2b.py - D:\pemula\Python\Latihan 2b.py (3.12.2)
File Edit Format Run Options Window Help

i = int(input("Input sebuah angka : "))
for j in range(1,11):
    print(i, "x", j, "=", i*j)
```

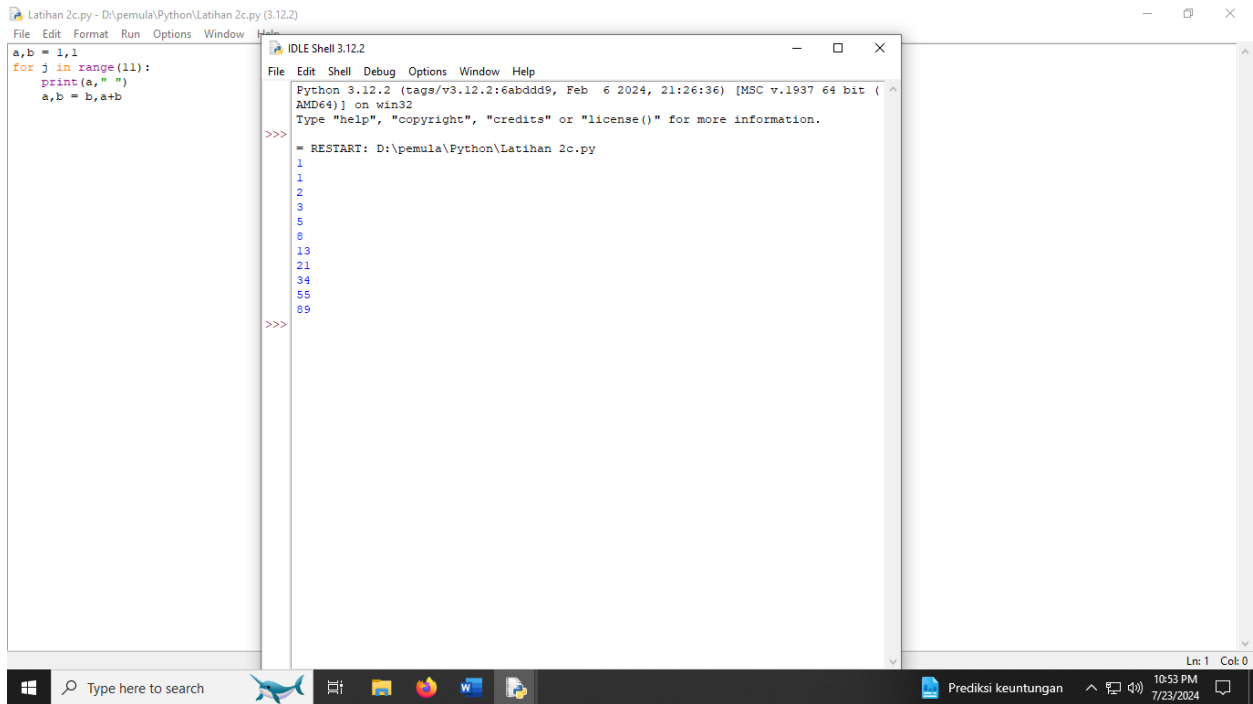
```
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: D:\pemula\Python\Latihan 2b.py
Input sebuah angka : 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
>>> |
```

Ln: 16 Cok: 0

2C.



The screenshot shows the Python IDLE 3.12.2 environment. The editor window displays a script named 'Latihan 2c.py' with the following code:

```
a,b = 1,1
for j in range(11):
    print(a, " ")
    a,b = b,a+b
```

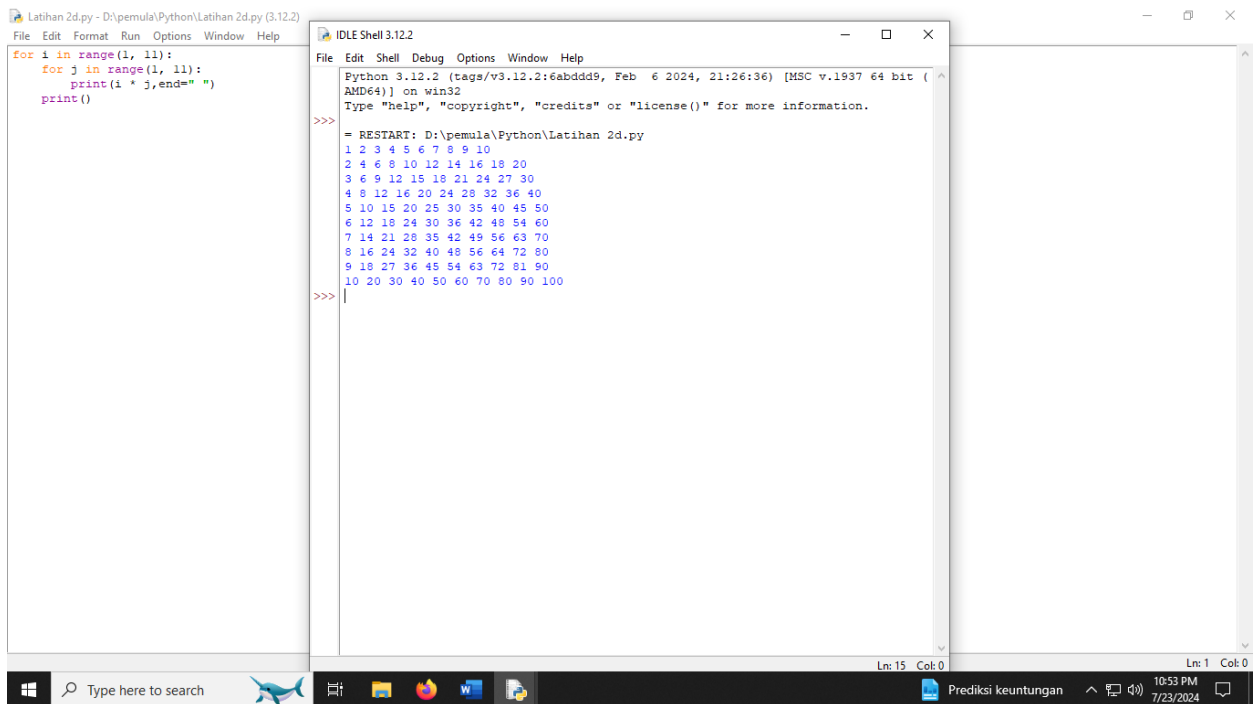
The Shell window shows the execution output:

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: D:\pemula\Python\Latihan 2c.py
1
1
2
3
5
8
13
21
34
55
89
>>>
```

The Windows taskbar at the bottom shows the search bar, task view, and several open applications including a file explorer, Firefox, and a document titled 'Prediksi keuntungan'. The system clock indicates 10:53 PM on 7/23/2024.

2D.



The screenshot shows the Python IDLE 3.12.2 environment. The editor window displays a script named 'Latihan 2d.py' with the following code:

```
for i in range(1, 11):
    for j in range(1, 11):
        print(i * j, end=" ")
    print()
```

The Shell window shows the execution output:

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: D:\pemula\Python\Latihan 2d.py
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
>>>
```

The Windows taskbar at the bottom shows the search bar, task view, and several open applications including a file explorer, Firefox, and a document titled 'Prediksi keuntungan'. The system clock indicates 10:53 PM on 7/23/2024.

## Analisis program

Penggunaan loop di Python memberikan fleksibilitas dan efisiensi dalam menulis kode yang membutuhkan eksekusi berulang. Memahami dan menguasai kedua jenis loop, for dan while, adalah langkah penting bagi setiap programmer Python untuk menyelesaikan berbagai masalah pemrograman secara efektif. Dengan loop, tugas-tugas yang sebelumnya membosankan dan repetitif dapat disederhanakan menjadi lebih terstruktur dan mudah diikuti.

## Referensi

<https://www.w3schools.com/python/>