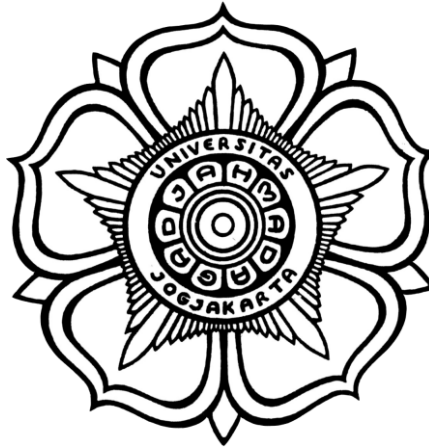


**LAPORAN PRAKTIKUM
PEMOGRAMAN KOMPUTER
PERTEMUAN V
PERULANGAN**



Disusun Oleh:

Nama : Noni Cindy Klaudia Matatar
NIM : 24/545671/25729
Kelas : B1
Dosen Pengampu : Yuris Mulya Saputra, S.T.,M.Sc.,Ph

**PROGRAM STUDI D-IV TEKNOLOGI REKAYASA INTERNET
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2024**

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
BAB II PEMBAHASAN	2
2.1 TUGAS 1	2
2.2 TUGAS 2	3
2.3 TUGAS 3	4
2.4 TUGAS 4	5
2.5 TUGAS 5	6
2.6 TUGAS 6	7
2.7 TUGAS 7	8
BAB III PENUTUP	9
3.1 Kesimpulan	9
DAFTAR PUSTAKA	10

DAFTAR GAMBAR

Gambar 2.1 TUGAS 1.....	2
Gambar 2.2 TUGAS 2.....	3
Gambar 2.3 TUGAS 3.....	4
Gambar 2.4 TUGAS 4.....	5
Gambar 2.5 TUGAS 5.....	6
Gambar 2.6 TUGAS 6.....	7
Gambar 2.7 TUGAS 7.....	8

BAB I

PENDAHULUAN

1.1 Latar Belakang

Looping atau perulangan merupakan sebuah struktur perintah yang sering digunakan pada sebuah aplikasi atau program. Perulangan juga sangat berguna pada program karena dapat kita pakai dalam struktur-struktur rumit yang besar yang memiliki kumpulan algoritma yang kompleks. 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.

Perulangan adalah apa yang disebut dalam pemrograman sebagai looping, yaitu sebuah ide dasar untuk mengatur serangkaian instruksi berulang hingga syarat tertentu terpenuhi. Dasar-dasar dan kepentingan yang mendasari penggunaan perulangan adalah sebagai berikut:

Efisiensi Kode: Bila sebuah tugas harus diulangi berkali-kali tanpa perulangan, then programmer harus menulis kode yang sama kali berulang. Dengan perulangan, tugas tersebut bisa dilakukan secara otomatis dan merupakan penemuan bertempat. Ada hadde hingga gagal.

Recurring Problems the :c Many problems faced in programming, such: iterasi data pada list atau array; pencarian data,;atau melakukan penghitungan matematis, memerlukan pemrosesan yang berulang. Perulangan memungkinkan kawa processing dari larikan-larikan dalam struktur data demikian tanpa harus dijangkau satu-persatu MAUY secara manual.

Manajemen logika kondisional: perulangan sering dikombinasikan dengan logika kondisional, seperti membuat pernyataan if, menentukan kapan perulangan harus berhenti.

Ini memungkinkan pengulangan yang dinamis berdasarkan suatu kondisi tertentu alih-alih hanya harus dilakukan jumlah iterasinya tetap.

Jenis Perulangan yang Diberikan:

For loop: Kita menggunakannya ketika jumlah iterasi sudah diketahui sebelumnya. Misalnya, untuk melakukan sebuah blok kode untuk setiap elemen di dalam array.

While loop: Kita menggunakannya ketika jumlah iterasi tidak diketahui, tapi perulangan harus berhenti saat suatu kondisi terpenuhi.

Do-while loop: Mirip dengan while loop, tetapi ada jaminan bahwa blok kode ini akan berjalan paling tidak satu kali.

Diterapkan Perulangan di Dunia Nyata:

Pemrosesan Data: Misalnya, membaca atau memanipulasi data dalam file.

Interaksi dengan Pengguna: Misalnya, meminta masukan pengguna sampai dia menerima masukan yang valid.

1.2 Rumusan Masalah

1. Efisiensi dalam Proses Berulang: Bagaimana membuat program yang mampu mengotomatisasi tugas berulang secara efisien tanpa harus menulis kode yang sama berulang kali?
2. Pengendalian Perulangan: Bagaimana menentukan kondisi awal, kondisi akhir, dan kondisi kelanjutan yang tepat agar perulangan berjalan sesuai tujuan dan berhenti di waktu yang tepat?
3. Pengelolaan Sumber Daya: Bagaimana menghindari penggunaan sumber daya berlebih, seperti memori dan CPU, selama perulangan?
4. Penanganan Infinite Loop: Bagaimana pengendalian yang dimungkinkan yang perlu ditambahkan ke dalam proses perulangan, jadi dalam kondisi apa pun perulangan tidak akan berakhir?
5. Pengaturan Output yang Konsisten: Bagaimana memastikan hasil dari proses perulangan tetap konsisten dan sesuai dengan yang diharapkan?

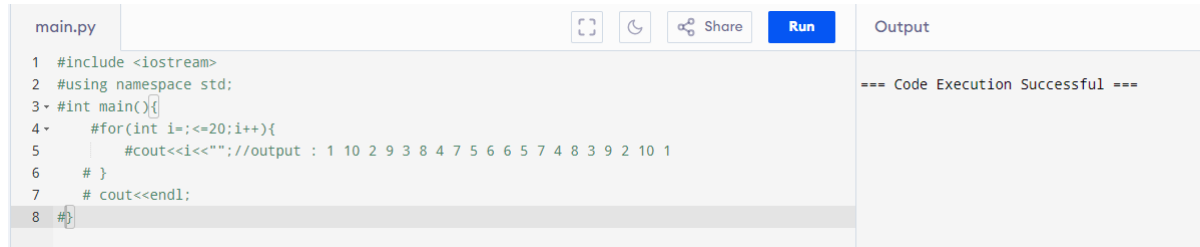
1.3 Tujuan

1. Oleh karena itu, tujuan dari perulangan dalam pemrograman bagi mahasiswa yang baru belajar dasar-dasar pemrograman antara lain sebagai berikut: 1. Efisiensi Kode. Perulangan memungkinkan penulisan kode yang lebih singkat dan efisien jika ada sesuatu yang harus dilakukan berkali-kali. Sebagai contoh, kepada mahasiswa tidak efisien jika harus menulis instruksi yang sama sebanyak 10 kali.
2. Dengan menggunakan loop, proses penulisan kode tersebut dapat diotomatiskan. 2. Pemahaman Konsep Dasar Algoritma. Dengan belajar perulangan, mahasiswa dapat memahami konsep dasar pada algoritma yaitu bagaimana sebuah proses yang sama dieksekusi beberapa kali sebelum kondisi tertentu tercapai.
3. Penggunaan struktur data: perulangan membantu mahasiswa memahami cara mengakses dan memanipulasi struktur data seperti array, list, atau matriks. Masalah-lah yang seringkali memerlukan pengulangan elemen-elemen yang sudah ada di dalamnya.. Problems Solving : Mahasiswa belajar memecahkan masalah dengan lebih mudah, karena banyak masalah pemrograman yang solusinya adalah menggunakan perulangan untuk mencari solusi secara bertahap.
4. Optimalisasi waktu eksekusi: menggunakan perulangan dengan baik sehingga mahasiswa bisa menulis program yang lebih cepat dan optimal dari segi waktu eksekusi. Hanya sangat berguna untuk masalah yang melibatkan perhitungan yang besar.

BAB II

PEMBAHASAN

2.1 TUGAS 1



```
main.py
1 #include <iostream>
2 #using namespace std;
3 int main(){
4     #for(int i=1; i<=20; i++){
5         #cout<<i<<" "; //output : 1 10 2 9 3 8 4 7 5 6 6 5 7 4 8 3 9 2 10 1
6     # }
7     # cout<<endl;
8 }
```

Output

=== Code Execution Successful ===

Analisis:

Baris 1-2<iostream> : Header () dan namespace () yang diperlukan stddisertakan sehingga program dapat menggunakan objek input-output standar seperti cout.

Baris 3 : Fungsi utama int main()didefinisikan, yang merupakan titik masuk program.

Baris 4-6 (dikomentari) :

Loop yang diberi komentar forberjalan dari i = 1ke i = 20.

Di dalam loop, ia mencetak nilai if<<"(meskipun

Output yang diharapkan (seperti terlihat dalam komentar) adalah sebuah pola: 1 10 2 9 3 8 4 7 5 6 5 7 4 8 3 9 2 10 1.

Baris 7 :cout << endl;, yang

Penjelasan

Outputnya berupa pola angka naik dan turun secara bergantian. Dimulai dengan: 1, 10, 2, 9, 3, 8, 4, 7, 5, 6, 5, 7, 4, 8, 3, 9

Polanya berganti-ganti antara angka rendah (mulai dari 1) dan angka tinggi (mulai dari 10), menambah angka rendah dan mengurangi angka tinggi hingga ke tengah urutan, lalu berbalik lagi.

2.2 TUGAS 2

main.py	Output
<pre>1 awal = int(input("Masukan saldo awal\t:")) 2 sisa = awal #bila tidak ada pengeluaran 3- while (True): 4 #pengeluaran = int(input("Masukan pengeluaran hari ini (-1 untuk 5 #keluar): ")) 6- if pengeluaran == -1: #untuk berhenti 7 #print("Sisa saldo = ", sisa) #sisa bila di berhentikan 8 break 9 #sisa = sisa - pengeluaran 10- #if sisa <0: 11 #print("Saldo tidak cukup") 12 #print("Sisa saldo ", sisa+pengeluaran) 13 break</pre>	Masukan saldo awal :

main.py	Output
<pre>1 awal = int(input("Masukan saldo awal\t:")) 2 sisa = awal #bila tidak ada pengeluaran 3- while (True): 4 pengeluaran = int(input("Masukan pengeluaran hari ini (-1 untuk 5 keluar): ")) 6- if pengeluaran == -1: #untuk berhenti 7 print("Sisa saldo = ", sisa) #sisa bila di berhentikan 8 break 9 sisa = sisa - pengeluaran 10- if sisa <0: 11 print("Saldo tidak cukup") 12 print("Sisa saldo ", sisa+pengeluaran) 13 break</pre>	<pre>ERROR! Traceback (most recent call last): File "<main.py>", line 4 pengeluaran = int(input("Masukan pengeluaran hari ini (-1 untuk ^ SyntaxError: invalid character "'" (U+201C) === Code Exited With Errors ===</pre>

Analisis:

Baris 1: awal = int(input("Masukan saldo awal \t:"))

Saldo awal diminta user agar memasukannya dan disimpan dalam variabel awal setelah diubah menjadi integer.

Baris 2: sisa = awal

Variabel sisa dengan nilai awal yang sama dengan saldo awal.sisa ini dibuat agar mengikuti sisa saldo.

Baris 3: while(True):

Program memasuki loop yang tak terbatas (infinite loop). Menyatakan bahwa program akan terjaln terus menerus selama tidak ada kondisi yang membuatnya terhenti.

Baris 4: pengeluaran = int(input("Masukan pengeluaran hari ini (-1 untuk keluar): "))

Pengeluaran diminta agar pengeluaran user memasukannya. Jika user memasukkan -1, maka program menyatakan user ingin keluar.

Baris 5-8: if pengeluaran == -1:

Jika input user sama dengan -1, maka break.

Baris 9-13 (Komentar dan Kode yang Di-Comment):




Ada beberapa baris kode yang di-comment (dimulai dengan tanda #), yang kemungkinan besar digunakan untuk menghitung dan menampilkan sisa saldo jika program dilanjutkan:

Mengurangi saldo dengan pengeluaran.

Mengecek apakah saldo cukup atau tidak.

Mencetak pesan saldo tidak cukup jika saldo menjadi negatif.

2.3 TUGAS 3

main.py	   Share	Run	Output
<pre>1 num = int(input("Multiplication table of: ")) 2 for i in range(1, 11): 3 print(num, 'x', i, '=', num*i)</pre>			Multiplication table of:

Analisis:

Output Pengguna:

Baris pertama dari kode output yang akan diminta oleh program adalah sebagai berikut:

```
num = int (input("Multiplication table of: "))
```

Program akan meminta pengguna untuk memberikan nomor ke dalam fungsi input() yang akan dimakukan oleh pengguna untuk memberikan angka dari angka ganjil yang diinputkan oleh pengguna. Fungsi int() diperlukan untuk mengonversi input yang dilakukan oleh pengguna sebagai integer. Variabel num akan berpengaruh dengan angka dalam bentuk input().

Looping:

Baris kedua dari kode looping yang akan diperintahkan oleh program adalah:

```
for i in range(1, 11):
```

Program akan melakukan looping yang menghitung pangkat angka dari 1 sampai 10 menggunakan sintak looping for. range(1,11) membuat angka yang ada dari 1 hingga 10 karena batas atas angka tidak dimasukan dan diakhiri dengan 11.

Output Table Multiplication:

Baris ketiga dari output dari program looping yang akan diproses oleh perintah sebagai berikut:

```
print(num, 'x', i, '=', num*i)
```

Fungsi print() berfungsi untuk menampilkan perkalian. Setiap looping iterasi akan menampilkan angka dengan num x i = resultan dari perhitungan i dengan num.

2.4 TUGAS 4

```
main.py
1 #include<iostream.h>
2 #include<conio.h>
3
4 void main() {
5
6     # float angka, total = 0, rata;
7     # int jumlah;
8
9     #cout << "1. Mencari Rata-rata. (jumlah angka ditentukan)" << endl;
10    #cout << endl;
11    #cout << "Masukkan jumlah angka      : "; cin >> jumlah;
12    #cout << endl;
13
14    #for (int i = 1; i <= jumlah; i++) {
15    #cout << "Masukkan angka              : "; cin >> angka;
16    #total = total + angka;
17
18    #cout << endl;
19    #cout << "Total                          : " << total << endl;
20    #rata = total / jumlah;
21    #cout << "Rata-rata                      : " << rata;
22    #getch();
23 }
```

Output

=== Code Execution Successful ===

Analisis:

input dari Pengguna:

Baris pertama: Enter the number you want to generate multiplication table for:

Pengguna diminta untuk memasukkan angka melalui fungsi input().

Int() berfungsi untuk mengubah input menjadi angka bulat.

Num akan menyimpan angka yang diinput oleh pengguna.

Perulangan (Looping):

Baris kedua: i = 1 while i < 11:

Untuk menghitung perkalian angka dari 1 hingga 10, program dapat menggunakan perulangan while.

membuat urutan angka dari 1 hingga 10 (batas atas ditetapkan sebagai angka sebelumnya, yaitu 11).

```
Int i = 1; while(i <= jumlah) { // Code here i++; }
```

```
cout <<
```

```
cout <<
```

```
cout
```

```
"Masukkan angka : "Masukkan angka: " ; scanf("%d", &angka);
```

```
total += angka;
```

```
}
```

```
total += angka;
```

```
total += angka
```

```
total
```

```
Aforlingkarantotal.
```

Menampilkan Total dan Menghitung Rata-rata:Menampilkan Total dan Menghitung Rata-rata:

```
cout << "Total : "Total: " << total << endl;
```

Bagi hasil total dengan jumlah untuk mendapatkan nilai rata-rata.

```
cout <<
```

rata-ratanya adalah total dibagi oleh jumlahnya.

```
cout <<
```

```
nilai rata-rata = total / jumlah;
```

```
cout
```

Rata-rata dapat dihitung dengan membagi total oleh jumlah.

rata-ratanya adalah total dibagi dengan jumlah tersebut.

```
rata = total /
```

```
rata = total
```

```
rata =
```

```
"Rata-rata : "Papagayo"
```

Setelah perulangan, program akan menampilkan nilai total dan melakukan perhitungan rata-rata ($\text{rata-rata} = \text{total} / \text{jumlah data}$).

Hentikan Output:

Salin kode

```
getch();
```

Itu getch() fungsi

2.5 TUGAS 5

main.py	Output
<pre>1 #Deklarasi variabel 2 var_nilai = 0 3 var_i = 1 4 #Perulangan FOR 5 for var_nilai in range (0,10) : 6 print("Perulangan pertama Ke ",var_nilai) 7 for var_i in range (1,3) : 8 print(" Perulangan ke ", var_nilai," ",var_i) 9 #diluarkan perulangan var_i 10 var_i = 1 11 #diluarkan perulangan var_nilai 12 print("var_nilai = ",int(var_nilai)+1," = 10. Bernilai False")</pre>	<pre>Perulangan pertama Ke 0 Perulangan ke 0 , 1 Perulangan ke 0 , 2 Perulangan pertama Ke 1 Perulangan ke 1 , 1 Perulangan ke 1 , 2 Perulangan pertama Ke 2 Perulangan ke 2 , 1 Perulangan ke 2 , 2 Perulangan pertama Ke 3 Perulangan ke 3 , 1 Perulangan ke 3 , 2 Perulangan pertama Ke 4 Perulangan ke 4 , 1 Perulangan ke 4 , 2 Perulangan pertama Ke 5 Perulangan ke 5 , 1 Perulangan ke 5 , 2 Perulangan pertama Ke 6 Perulangan ke 6 , 1 Perulangan ke 6 , 2 Perulangan pertama Ke 7 Perulangan ke 7 , 1 Perulangan ke 7 , 2 Perulangan pertama Ke 8 Perulangan ke 8 , 1 Perulangan ke 8 , 2 Perulangan pertama Ke 9 Perulangan ke 9 , 1 Perulangan ke 9 , 2 var_nilai = 10 = 10. Bernilai False === Code Execution Successful ===</pre>

Analisis:

Program tersebut menggunakan struktur perulangan FOR dan memiliki dua variabel utama: nilai_var dan i_var. Berikut adalah cara membaca program tersebut: Berikut adalah cara membaca program tersebut:

Deklarasi variabel:

var_nilai = 0: nilai variabel ini diberikan awal 0.

var_i = 1: nilai awal var_i adalah 1.

Perulangan FOR pertama:

Loop ini akan berjalan dari 0 hingga 9 (for var_nilai in range(0,10)): Could you please provide the original sentence that you would like an alternative for?

Tiap kali melakukan perulangan, program akan mencetak "Perulangan ke-" dan disusul dengan nilai var_nilai pada saat itu.

Perulangan FOR kedua (nested):

Di dalam perulangan pertama, ada perulangan kedua yang berjalan dari 1 hingga 2 (for var_i in range(1,3)): Can you please provide the sentence for me to create an alternative version?

Dalam perulangan ini, kita mencetak "Perulangan ke -" diikuti dengan nilai var_nilai dan var_i.

Setelah perulangan:

Walaupun nilai var_i telah diubah menjadi 1 (var_i = 1), tetapi tidak ada perubahan yang terjadi karena nilainya sudah 1 sebelumnya.

Kemudian, program akan mencetak hasil evaluasi logika pada baris terakhir untuk mengecek apakah ekspresi (var_nilai - int(var_nilai)) benar atau salah. = Pernyataan no. 10 dapat memiliki nilai yang benar atau salah.

Secara umum, program ini mencetak urutan perulangan dan kemudian mengevaluasi kondisi logika ketika perulangan selesai.

Hasil setiap perulangan yang terjadi sesuai dengan struktur di atas ditampilkan di sebelah kanan.

2.6 TUGAS 6

[illegible]

Analysis:

Dalam gambar tersebut, perulangan menggunakan while untuk mengulang proses hingga nilai variabel var nilai mencapai 10.

Penetapan variabel: var nilai dimulai dengan nilai 0. var i diinisialisasi dengan 1.

pertama (while var_nilai < 10): Program akan terus berjalan selama nilai var_nilai kurang dari 10. Dalam iterasi ini, program akan mencetak "Iterasi pertama adalah 0", yang menunjukkan nilai var_nilai. Perulangan kedua (while var_i < 3): Dalam perulangan (selama var_i < 3), mencetak "Perulangan ke-0,1" dan terus menambah nilai var_i hingga mencapai angka 3. Setelah keluar dari perulangan dalam: nilai var_i diset ulang menjadi 1, dan kemudian ditambahkan nilai 1 ke variabel var_nilai. Cek akhir: Setelah keluar dari perulangan luar, program akan mencetak kalimat "var_nilai = .. Bernilai Bukan True" yang menandakan bahwa nilai var_nilai sudah melebihi atau sama dengan 10. Hal ini menyebabkan kondisi while var_nilai < 10 menjadi salah (False). Jadi, secara keseluruhan, program ini melakukan perulangan bersarang: dalam perulangan pertama, ulang sampai var_nilai mencapai 10; dalam perulangan tersebut terdapat perulangan kedua yang dijalankan sebanyak 3 kali.

2.7 TUGAS 7

[illegible]

Analysis:

Program ini memanfaatkan bahasa pemrograman Python. Berikut penjelasan baris per baris untuk memahami program tersebut: Berikut penjelasan baris per baris untuk memahami program tersebut:

Deklarasi variabel:

nilai perulangan = 0 → Nilai ini akan digunakan untuk menyimpan perulangan.

1 digunakan sebagai variabel dalam perulangan while.

Perulangan FOR:

for var nilai in range(0,10): → Perulangan for ini menjalankan iterasi dari 0 hingga 9.

```
print("Cetak nilai perulangan pertama Ke -", var nilai)
```

Perulangan WHILE:

while(var i < 3): → Selama nilai var i kurang dari 3, perulangan while akan terus berjalan.

```
while var i > 0: print("Perulangan ke -", var nilai, ". ", var i) var i -=1
```

Setiap kali iterasi, var i akan meningkat sebesar 1.

Mengulang kembali nilai var i :

Setelah perulangan while selesai, nilai var_i akan direset kembali menjadi 1 agar dapat digunakan lagi.

Mencetak hasil akhir:

```
if int(var_nilai) >= 10: print("var_nilai telah mencapai atau melebihi angka 10.") else:
    print("var_nilai belum mencapai angka 10.")
```

Di sebelah kanan output, hasil dari setiap iterasi perulangan for maupun while terlihat.

BAB III

PENUTUP

3.1 Kesimpulan

Dengan demikian, inti dari perulangan dalam pemrograman adalah: sangat efisien. Perulangan terutama memungkinkan pengulangan otomatis dari blok kode tertentu. Ini mungkin dilakukan tanpa menulis kode yang sama berulang kali. Juga, perulangan adalah berbagai tipe. Dalam definisi lain, ada metode ayun lain, seperti for loop, while loop, dan do-while loop. For loop menggunakan jumlah iterasi terutama, while loop digunakan dengan kebutuhan kondisi khusus yang harus dipenuhi sebelum dilanjutkan perulangan, do-while loop yang serupa dengan while loop tetapi kondisi memeriksa sesudah menjalankan kode. Serta, karena ada pengulangan, tidak perlu duplikasi kode yang belum dihilangkan akan menambahkan waktu dan upaya pada pengembangan yang tidak perlu dan risiko awal menghilangkan kesalahan.

Fleksibilitas: Perulangan bisa dikombinasikan dengan suatu logika kondisi jika memerlukan dalam loop menghasilkan kontrol lebih banyak.

Secara keseluruhan, perulangan memudahkan proses pemrograman sederhana yang melibatkan tugas-tugas berulang; serentak menunjukkan fleksibilitas dan performa kode.

DAFTAR PUSTAKA

[1] <https://www.programiz.com/python-programming/while-loop> –diakses tanggal 17 Mei 2021 [2] <https://pythonbasics.org/while-loop/> – diakses tanggal 17 Mei 2021 [3] Nurul Huda, 22 January 2021, “Mempelajari Perulangan For Python”

<https://repository.unikom.ac.id/61515/1/Materi%208%20perulangan%20dalam%20java%20script-kulon.pdf>

<https://www.petanikode.com/python-perulangan/>

<https://jagongoding.com/python/dasar/perulangan-for/>