

**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
KOMPUTER
MODUL 6
REPETITION**

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah
Praktikum Basis Data



Disusun Oleh :

AHSAN MAULANA RIZQI

104062400071

S1 BD 04 B

**PROGRAM STUDI S1 BISNIS DIGITAL
TELKOM UNIVERSITY PURWOKERTO
2025**

1. Dasar Teori

1.1. Repetition

Perulangan atau bisa juga disebut dengan looping/iteration/repetition, ada dua jenis perulangan pada python yaitu for dan while.

A. For

Secara sederhana penggunaan for ditujukan untuk:

- Counted loop (perulangan yang terhitung).
- Biasanya digunakan untuk perulangan yang sudah jelas banyaknya.
- Perulangan berdasarkan jumlah elemen.

B. While

Secara sederhana penggunaan while ditujukan untuk:

- Uncounted loop (perulangan yang tak terhitung).
- Biasanya digunakan untuk perulangan yang belum tentu banyaknya.

1.2. Control Statements

Control statements adalah sebuah statement yang bisa digunakan untuk mengubah cara eksekusi dalam percabangan dan perulangan dari cara eksekusi normal. Control statement terdiri dari:

A. Break Statement

Break statement adalah sebuah statement yang akan membuat suatu program berhenti atau keluar dari suatu blok program.

B. Continue Statement

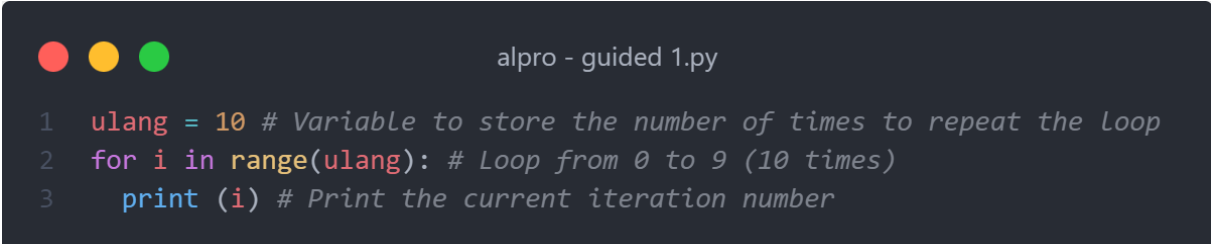
Continue statement adalah sebuah statement yang akan membuat suatu program langsung menuju step selanjutnya atau interval berikutnya dan mengabaikan (skip) baris kode di bawahnya.

C. Pass Statement

Pass statement adalah sebuah statement yang akan membuat suatu blok program tidak akan dieksekusi.

2. Pembahasan Tugas Guided

1.1 Coding repetition for



```
alpro - guided 1.py
1  ulang = 10 # Variable to store the number of times to repeat the loop
2  for i in range(ulang): # Loop from 0 to 9 (10 times)
3      print(i) # Print the current iteration number
```

Gambar 1. Coding repetition for

Variabel ulang yang memiliki value 10 untuk melakukan pengulangan value ulang. For I in range (ulang) code ini digunakan untuk pengulangan sebanyak variabel ulang yang dimulai dari 0 yang akan disimpan di variabel I.

```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 1.py"
0
1
2
3
4
5
6
7
8
9
○ kamize@server:~/web/alpro/week 6$
```

Gambar 2. Hasil dari coding repetition for

1.2 Coding repetition for untuk range 1

```
● ● ● alpro - guided 2.py
1 for i in range(1,10,2): # looping dari 1 sampai 10 dengan increment 2
2     print (i) # 1,3,5,7,9
```

Gambar 3. Coding repetition for untuk range 1

Code di gambar 3 menunjukkan formula for loop untuk melakukan pengulangan setiap 2 step. Formula for loop adalah for variabel in range(awal,akhir,step).

```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 2.py"
1
3
5
7
9
○ kamize@server:~/web/alpro/week 6$
```

Gambar 4. Hasil dari coding repetition for untuk range 1

1.3 Coding repetition for untuk range 2

```
● ● ● alpro - guided 3.py
1 # menggunakan start dan stop
2 for i in range(1, 10): # for Loop yang menggunakan range 1 sampai 9
3     print(i, end=' ')
4 print()
5 # menggunakan start dan stop default
6 for i in range(10): # for Loop yang menggunakan range 0 sampai 9
7     print(i, end=' ')
8 print()
9 # menggunakan start, stop dan step
10 for i in range(1, 10, 2): # for Loop yang menggunakan range 1 sampai 9 dengan step 2
11     print(i, end=' ')
12 print()
```

Gambar 5. Coding repetition for untuk range 2

Code di gambar 5 menunjukkan contoh penggunaan formula for loop. Formula for loop adalah for variabel in range(start,stop,step).

```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 3.py"
1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
1 3 5 7 9
○ kamize@server:~/web/alpro/week 6$
```

Gambar 6. Hasil Coding repetition for untuk range 2

1.4 Coding repetition while

```
● ● ● alpro - guided 4.py
1 jawab = "ya" # Inisialisasi variabel jawab
2 hitung = 0 # Inisialisasi variabel hitung
3 while jawab == "ya": # Looping selama jawab == "ya"
4     hitung += 1 # Increment hitung
5     jawab = input("Apakah anda ingin menghitung lagi? (ya/tidak) : ") # Input untuk melanjutkan atau tidak
6 print("Jumlah perulangan: ", hitung) # Menampilkan jumlah perulangan
```

Gambar 7. Coding repetition while

Inisialisasi variabel jawab dengan value “ya” dan variabel hitung dengan value 0. While loop untuk melakukan perulangan dengan kondisi jawab == “ya”. Variabel hitung ditambahkan dengan 1. Variabel jawab meminta input dari user. Jika user memasukan “ya” maka akan dilakukan perulangan, jika user memasukan selain “ya” maka kode akan berhenti.

```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 4.py"
Apakah anda ingin menghitung lagi? (ya/tidak) : ya
Apakah anda ingin menghitung lagi? (ya/tidak) : ya
Apakah anda ingin menghitung lagi? (ya/tidak) : tidak
Jumlah perulangan: 3
○ kamize@server:~/web/alpro/week 6$
```

Gambar 8. Hasil Coding repetition while

1.5 Coding repetition while dengan if statement

```
● ● ● alpro - guided 5.py
1 while True: # This is an infinite loop that will keep running until we break out of it
2     # The loop will keep asking for input until the user types 'stop'
3     reply = input("Enter text, [type 'stop' to quit:] ") # This line prompts the user for input and stores it in the variable 'reply'
4     # The input is expected to be a string, and the user can type anything they want
5     if reply == "stop": # This line checks if the user input is equal to the string 'stop'
6         break # If the user types 'stop', the loop will break and the program will end
7     print(reply.lower()) # This line prints the user input in lowercase, regardless of how the user typed it
```

Gambar 9. Coding repetition while dengan if statement

Pada gambar 9 menunjukkan while loop dengan infinite loop atau perulangan yang terus menerus menggunakan kondisi True. Variabel reply untuk meminta input dari user. If statement dengan kondisi jika variabel reply == “stop” maka kode akan berhenti dengan break statement. Variabel reply dengan method lower() untuk mengubah value variabel reply menjadi lowercase.

```

● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 5.py"
Enter text, [type 'stop' to quit:] www.
www.
Enter text, [type 'stop' to quit:] name
name
Enter text, [type 'stop' to quit:] stop
○ kamize@server:~/web/alpro/week 6$

```

Gambar 10. Hasil Coding repetition while dengan if statement

1.6 Coding repetition while dengan if statement dan operator

```

alpro - guided 6.py

1 x = 0 # Inisialisasi variabel x
2 while True: # Looping tanpa batas
3     nama = input("Masukkan nama anda: ") # Meminta input nama dari pengguna
4     print("Nama anda adalah: ", nama) # Menampilkan nama yang dimasukkan
5     x += 1 # Menambah nilai x setiap iterasi
6     if x == 5: # Jika x yang dimasukkan adalah 5, keluar dari loop
7         break # break statement untuk keluar dari loop

```

Gambar 11. Coding repetition while dengan if statement dan operator

Pada gambar 11 menunjukan contoh penggunaan while loop menggunakan if statement dan penggunaan operator. Inisialisasi variabel x yang memiliki value 0. While True untuk melakukan perulangan tanpa batas. Variabel nama untuk meminta input dari user. Variabel x akan ditambahkan dengan 1 setiap perulangan. If statement dengan kondisi x == 5. Jika variabel x sama dengan 5 maka jalankan break statement untuk keluar dari while loop.

```

● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 6.py"
Masukkan nama anda: AHSAN
Nama anda adalah: AHSAN
Masukkan nama anda: MAULANA
Nama anda adalah: MAULANA
Masukkan nama anda: RIZQI
Nama anda adalah: RIZQI
Masukkan nama anda: S1
Nama anda adalah: S1
Masukkan nama anda: BD
Nama anda adalah: BD
○ kamize@server:~/web/alpro/week 6$

```

Gambar 12. Coding repetition while dengan if statement dan operator

1.7 Coding break statement

```

alpro - guided 7.py

1 for i in range(30): # program meminta perulangan 30 kali
2     print(i) # Menampilkan nilai i dari 0 hingga 29
3     if i == 20: # jika i yang dimasukkan adalah 20, keluar dari loop
4         break # break statement untuk keluar dari loop

```

Gambar 13. Code break statement

Pada gambar 13 for loop akan meminta perulangan sebanyak 30 kali. if statement dengan kondisi variabel `i == 20`. Jika variabel `i` memiliki value 20 maka kode atau perulangan nya akan berhenti dengan menggunakan break statement.

```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 7.py"
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
○ kamize@server:~/web/alpro/week 6$
```

Gambar 14. Code break statement

1.8 Coding continue statement

```
● ● ● alpro - guided 8.py
1 i = 0 # Inisialisasi variabel i
2 while i < 8: # Looping selama i kurang dari 8
3     i += 1 # Menambah nilai i setiap iterasi
4     if i == 6: # Jika i yang dimasukkan adalah 6, Lanjutkan ke iterasi berikutnya
5         continue # continue statement untuk melanjutkan ke iterasi berikutnya
6     print(i, end=' ') # Menampilkan nilai i yang tidak sama dengan 6
7 print()
```


Gambar 15. Coding continue statement

Pada gambar 15 terdapat variabel `i` dengan value 0. While loop dengan kondisi variabel `i` kurang dari 8 yang artinya akan melakukan perulangan selama value variabel `i` kurang dari 8. Variabel `i` akan ditambahkan dengan 1 di setiap perulangan. If statement dengan kondisi variabel `i` sama dengan 6. Jika value variabel `i` sama dengan 6 maka akan dilanjutkan ke perulangan selanjutnya menggunakan continue statement.

```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 8.py"
1 2 3 4 5 7 8
○ kamize@server:~/web/alpro/week 6$
```

Gambar 16. Hasil dari Coding continue statement

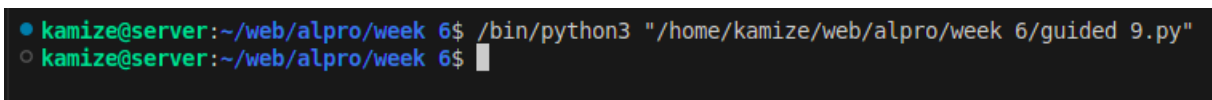
1.9 Coding pass statement



```
alpro - guided 9.py
1 data = [1, 2, 3, 4, 5] # List data
2 for i in data: # Looping data
3     pass # tidak ada yang dilakukan pada setiap iterasi karena perintah pass adalah perintah kosong
```

Gambar 17. Code pass statement

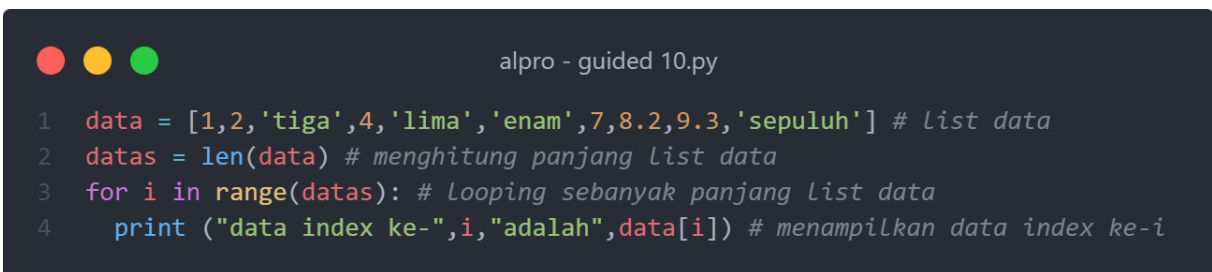
Pada gambar 17 terdapat list data dengan value [1,2,3,4,5]. For I in data untuk melakukan perulangan melalui setiap item dalam list data. Pass statement akan membuat program tidak akan dieksekusi



```
kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 9.py"
kamize@server:~/web/alpro/week 6$
```

Gambar 18. Hasil Code pass statement

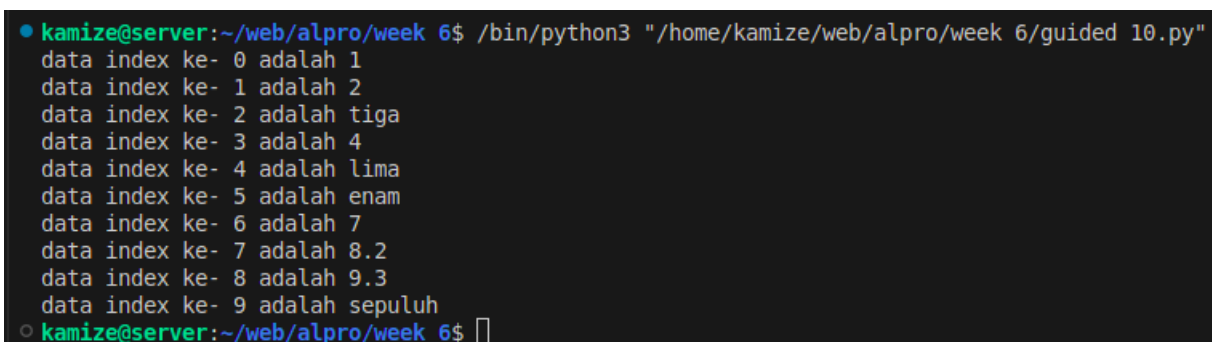
1.10 Coding repetition for dengan list



```
alpro - guided 10.py
1 data = [1,2,'tiga',4,'lima','enam',7,8.2,9.3,'sepuluh'] # List data
2 datas = len(data) # menghitung panjang list data
3 for i in range(datas): # looping sebanyak panjang list data
4     print ("data index ke-",i,"adalah",data[i]) # menampilkan data index ke-i
```

Gambar 19. Code repetition for dengan list

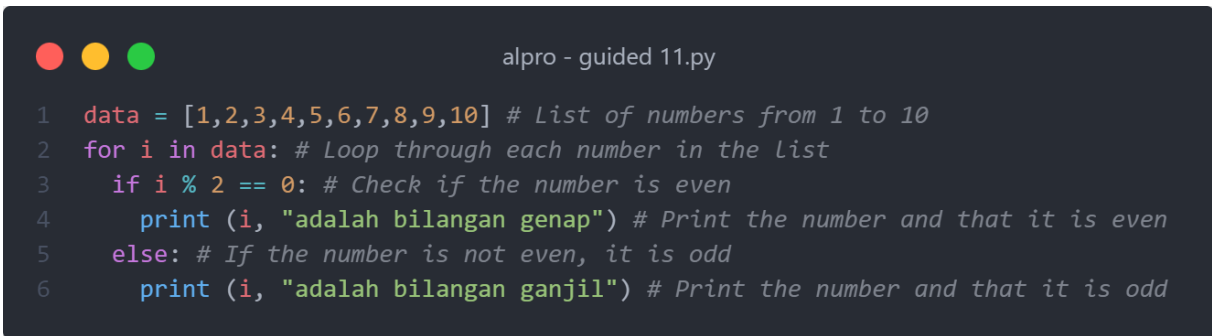
Pada gambar 17 terdapat variabel data yang bertipe data list dengan value [1, 2, tiga, 4, lima, enam, 7, 8.2, 9.3, sepuluh]. Variabel datas dengan value panjang dari variabel list menggunakan fungsi len (data). Perulangan for dengan perulangan sebanyak value panjang.



```
kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 10.py"
data index ke- 0 adalah 1
data index ke- 1 adalah 2
data index ke- 2 adalah tiga
data index ke- 3 adalah 4
data index ke- 4 adalah lima
data index ke- 5 adalah enam
data index ke- 6 adalah 7
data index ke- 7 adalah 8.2
data index ke- 8 adalah 9.3
data index ke- 9 adalah sepuluh
kamize@server:~/web/alpro/week 6$
```

Gambar 20. Code repetition for dengan list

1.11 Coding repetition for dengan list dan if else statement



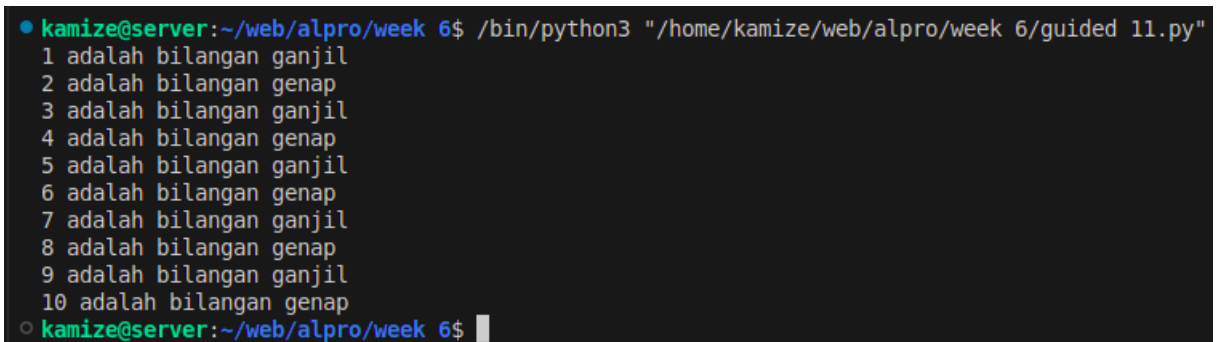
```

alpro - guided 11.py
1 data = [1,2,3,4,5,6,7,8,9,10] # List of numbers from 1 to 10
2 for i in data: # Loop through each number in the list
3     if i % 2 == 0: # Check if the number is even
4         print(i, "adalah bilangan genap") # Print the number and that it is even
5     else: # If the number is not even, it is odd
6         print(i, "adalah bilangan ganjil") # Print the number and that it is odd

```

Gambar 21. Code repetition for dengan list dan if else statement

Pada gambar 17 terdapat variabel data yang bertipe data list dengan value [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. For I in data untuk melakukan perulangan setiap item yang ada di dalam list data. If statement dengan kondisi value variabel I dimoduluskan 2 sama dengan 0, maka munculkan value variabel I dan string adalah bilangan genap. Jika kondisi statement nya False maka munculkan alue variabel I dan string adalah bilangan ganjil.



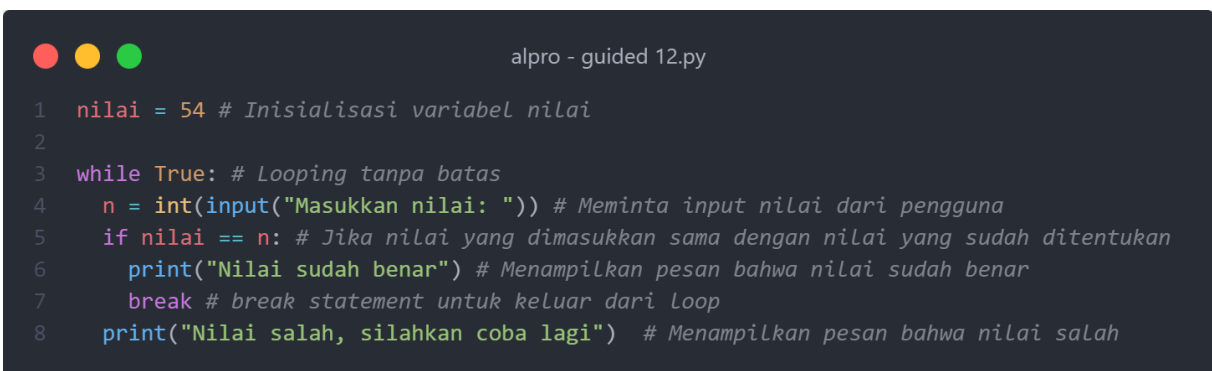
```

kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 11.py"
1 adalah bilangan ganjil
2 adalah bilangan genap
3 adalah bilangan ganjil
4 adalah bilangan genap
5 adalah bilangan ganjil
6 adalah bilangan genap
7 adalah bilangan ganjil
8 adalah bilangan genap
9 adalah bilangan ganjil
10 adalah bilangan genap
kamize@server:~/web/alpro/week 6$

```

Gambar 22. Hasil Code repetition for dengan list dan if else statement

1.12 Coding repetition while dengan if dan break statement



```

alpro - guided 12.py
1 nilai = 54 # Inisialisasi variabel nilai
2
3 while True: # Looping tanpa batas
4     n = int(input("Masukkan nilai: ")) # Meminta input nilai dari pengguna
5     if nilai == n: # Jika nilai yang dimasukkan sama dengan nilai yang sudah ditentukan
6         print("Nilai sudah benar") # Menampilkan pesan bahwa nilai sudah benar
7         break # break statement untuk keluar dari loop
8     print("Nilai salah, silahkan coba lagi") # Menampilkan pesan bahwa nilai salah

```

Gambar 23. Code repetition while dengan if dan break statement

Pada gambar 23 menunjukan contoh penggunaan while loop degan if dan break statement. Pada code terdapat variabel nilai dengan value 54 dan variabel n untuk menyimpan value dari input user. While True adalah perulangan tanpa batas dengan while untuk perulanagn dan True untuk kondisi atau statement. If statement dengan

kondisi value variabel nilai sama dengan value variabel n dan menghasilkan value True, maka jalankan break statement dan munculkan string nilai sudah benar.

```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 12.py"
Masukkan nilai: 75
Nilai salah, silahkan coba lagi
Masukkan nilai: 54
Nilai sudah benar
○ kamize@server:~/web/alpro/week 6$
```

Gambar 24. Hasil Code repetition while dengan if dan break statement

1.13 Coding repetition while dan for

```
alpro - guided 13.py
1 lines = list() # Digunakan untuk menyimpan baris teks yang dimasukkan oleh pengguna
2 testAnswer = input("Apakah Anda ingin memasukan baris ke list? (y): ") # jika pengguna ingin memasukan baris ke list ketik "y"
3 while testAnswer == "y": # jika pengguna mengetik "y" maka program akan meminta pengguna untuk memasukan baris teks
4     line = input("Masukkan baris teks: ") # meminta pengguna untuk memasukan baris teks
5     lines.append(line) # digunakan untuk menambahkan item ke akhir sebuah list
6     testAnswer = input("Apakah Anda ingin menambahkan baris lagi? (y): ")
7 print ("Baris yang telah anda masukan adalah: ") # jika pengguna mengetik selain "y"
8 for line in lines: # Ketika pengguna mengetik selain "y", semua baris yang telah dimasukkan akan ditampilkan
9     print(line) # menampilkan semua baris yang telah dimasukkan
```

Gambar 25. Code repetition while dan for

Pada gambar 25 menunjukkan contoh pernggunaan while loop dan for loop. While loop digunakan untuk menambahkan isi ke dalam sebuah list berdasarkan input dari user dan for loop digunakan untuk menampilkan semua isi yang ada di dalam list.

1.13.1 Inisialisasi variabel

A. `lines = list()`

variabel ini digunakan untuk menyimpan list kosong menggunakan fungsi `list()`.

B. `testAnswer = input("Apakah Anda ingin memasukan baris ke list? (y): ")`

variabel ini digunakan untuk meminta input dari user dan memunculkan string “Apakah Anda ingin memasukan baris ke list? (y):” ke console atau layar.

1.13.2 Proses code

A. `While testAnswer == y.`

While adalah kata kunci yang digunakan untuk memulai loop. Jika variabel `testAnswer` sama dengan “y”, maka proses perulangan berjalan atau mulai. Pada proses ini code while akan di ulang terus menerus sampai user memasukan value selain “y”.

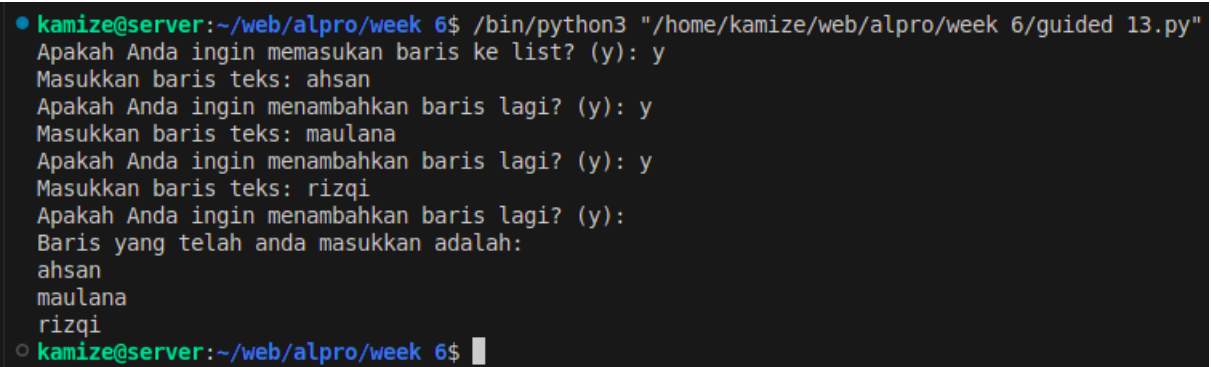
B. `line = input("Masukkan baris teks: ")`

variabel ini digunakan untuk meminta input dari user dan memunculkan string “Masukkan baris teks: ” ke console atau layar.

C. `Lines.append(line)`

Lines sebagai list kosong. Method `append()` untuk menambahkan isi atau item kedalam list `lines`. Line sebagai value input user. Jadi variabel `lines` diisi sesuai dengan value dari input user yang terdapat pada variabel `line`.

- D. `testAnswer = input("Apakah Anda ingin menambahkan baris lagi? (y): ")`
variabel ini digunakan untuk meminta input dari user dan memunculkan string “Apakah Anda ingin menambahkan baris lagi? (y): ” ke console atau layar.
- E. `print ("Baris yang telah anda masukkan adalah: ")`
proses ini untuk memunculkan string “Baris yang telah anda masukkan adalah:” ke console atau layer.
- F. `for line in lines:`
`for` adalah kata kunci yang digunakan untuk memulai loop. Line adalah variabel yang digunakan untuk menyimpan nilai elemen saat ini dari iterable atau objek yang diulang dalam setiap iterasi atau perulangan. In digunakan untuk merujuk pada iterable tempat variabel `line` akan melakukan iterasi. `lines` adalah objek yang elemen-elemennya akan diiterasi.
- G. `print(line)`
proses untuk memunculkan value variabel `line` dari setiap `for` loop.



```
● kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/guided 13.py"
Apakah Anda ingin memasukan baris ke list? (y): y
Masukkan baris teks: ahsan
Apakah Anda ingin menambahkan baris lagi? (y): y
Masukkan baris teks: maulana
Apakah Anda ingin menambahkan baris lagi? (y): y
Masukkan baris teks: rizqi
Apakah Anda ingin menambahkan baris lagi? (y):
Baris yang telah anda masukkan adalah:
ahsan
maulana
rizqi
○ kamize@server:~/web/alpro/week 6$
```

Gambar 26. Hasil Code repetition while dan for

3. Pembahasan Tugas Unguided

3.1. Unguided 1: Kalkulator

3.1.1. Rules

- A. Buatlah sebuah kalkulator sederhana untuk menghitung 2 buah bilangan dengan operator(*,/,+,-).
- B. Jika operator (*) diinputkan maka akan melakukan proses perkalian.
- C. Jika operator (/) diinputkan maka akan melakukan proses pembagian.
- D. Jika operator (+) diinputkan maka akan melakukan proses penjumlahan.
- E. Jika operator (-) diinputkan maka akan melakukan proses pengurangan.

- F. Jika operator yang diinputkan selain operator (*,/,+,-), munculkan pesan operator tidak valid.
- G. Jika operator yang diinputkan adalah bilangan 0 maka program diakhiri.
- H. Bilangan tidak boleh dibagi dengan 0.
- I. Gunakan percabangan dan perulangan

3.1.2. Jawaban

```
alpro - unguided 1.py

1 while True:
2     print("Kalkulator Sederhana")
3     print("Masukkan 0 pada operator untuk keluar dari program.")
4
5     try:
6         # Input bilangan pertama
7         bilangan1 = float(input("Masukkan bilangan pertama: "))
8
9         # Input operator
10        operator = input("Masukkan operator (*, /, +, -): ")
11
12        # Jika operator adalah 0, keluar dari program
13        if operator == "0":
14            print("Program dihentikan.")
15            break
16
17        # Input bilangan kedua
18        bilangan2 = float(input("Masukkan bilangan kedua: "))
19
20        # Proses perhitungan berdasarkan operator
21        if operator == "*":
22            hasil = bilangan1 * bilangan2
23            print(f"Hasil: {bilangan1} * {bilangan2} = {hasil}")
24        elif operator == "/":
25            if bilangan2 == 0:
26                print("Error: Tidak dapat membagi dengan 0.")
27            else:
28                hasil = bilangan1 / bilangan2
29                print(f"Hasil: {bilangan1} / {bilangan2} = {hasil}")
30        elif operator == "+":
31            hasil = bilangan1 + bilangan2
32            print(f"Hasil: {bilangan1} + {bilangan2} = {hasil}")
33        elif operator == "-":
34            hasil = bilangan1 - bilangan2
35            print(f"Hasil: {bilangan1} - {bilangan2} = {hasil}")
36        else:
37            print("Operator tidak valid. Silakan masukkan operator yang benar (*, /, +, -).")
38    except ValueError:
39        print("Input tidak valid. Masukkan angka yang benar.")
40
41    print() # Baris kosong untuk pemisah antar iterasi
```

Gambar 27. Unguided 1: Kalkulator

Program ini merupakan kalkulator sederhana berbasis teks yang berjalan dalam sebuah loop tak hingga (while True). Di awal setiap iterasi, program menampilkan judul dan pesan petunjuk bahwa pengguna dapat keluar dari program dengan

memasukkan operator 0. Selanjutnya, program meminta pengguna untuk memasukkan bilangan pertama, diikuti oleh input operator matematika yang diinginkan (*, /, +, atau -).

Jika pengguna memasukkan 0 sebagai operator, maka program akan mencetak pesan bahwa program dihentikan dan keluar dari loop menggunakan perintah break. Namun, jika operator valid dimasukkan, program melanjutkan dengan meminta bilangan kedua.

Program kemudian melakukan perhitungan berdasarkan operator yang dipilih:

- Jika operator adalah *, maka dilakukan perkalian.
- Jika operator /, maka dilakukan pembagian, tetapi program memeriksa terlebih dahulu apakah bilangan kedua adalah nol (untuk mencegah pembagian dengan nol).
- Jika operator adalah -, maka dilakukan pengurangan.
- Jika operator adalah +, maka dilakukan penjumlahan.

Setiap hasil perhitungan akan ditampilkan menggunakan print() dengan format yang jelas dan rapi. Bila operator yang dimasukkan tidak valid, program memberikan pesan kesalahan yang menyarankan operator yang benar. Program juga menangani kesalahan input (seperti pengguna memasukkan huruf alih-alih angka) menggunakan blok try-except dan akan menampilkan pesan bahwa input tidak valid jika terjadi ValueError.

Akhir setiap iterasi memberikan satu baris kosong untuk memisahkan antar perhitungan agar tampilan lebih bersih. Program ini cocok digunakan sebagai latihan logika dasar, penggunaan loop, percabangan, dan exception handling dalam Python

3.1.3. Hasil

```

• kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/unguided 1.py"
Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: 10
Masukkan operator (*, /, +, -): *
Masukkan bilangan kedua: 12
Hasil: 10.0 * 12.0 = 120.0

Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: 10
Masukkan operator (*, /, +, -): /
Masukkan bilangan kedua: 2
Hasil: 10.0 / 2.0 = 5.0

Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: 10
Masukkan operator (*, /, +, -): +
Masukkan bilangan kedua: 12
Hasil: 10.0 + 12.0 = 22.0

Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: 10
Masukkan operator (*, /, +, -): -
Masukkan bilangan kedua: 12
Hasil: 10.0 - 12.0 = -2.0

Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: 10
Masukkan operator (*, /, +, -): /
Masukkan bilangan kedua: 0
Error: Tidak dapat membagi dengan 0.

Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: b
Input tidak valid. Masukkan angka yang benar.

Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: 10
Masukkan operator (*, /, +, -): s
Masukkan bilangan kedua: 2
Operator tidak valid. Silakan masukkan operator yang benar (*, /, +, -).

Kalkulator Sederhana
Masukkan 0 pada operator untuk keluar dari program.
Masukkan bilangan pertama: 0
Masukkan operator (*, /, +, -): 0
Program dihentikan.
○ kamize@server:~/web/alpro/week 6$ █

```

Gambar 28. Unguided 1: Kalkulator

3.2. Unguided 2: Piramida Bintang

3.2.1. Rules

- A. menggunakan perulangan
- B. boleh memakai for maupun while

3.2.2. Jawaban

```
alpro - unguided 2.py

1 sisi = 20 # sisi segitiga
2 count = 1 # jumlah bintang
3 spasi = sisi // 2 # jumlah spasi
4
5 while True: # loop untuk mencetak segitiga
6     if count <= sisi: # jika jumlah bintang kurang dari atau sama dengan sisi segitiga
7         print(" " * spasi , "*" * count) # mencetak spasi dan bintang
8         count += 2 # menambah jumlah bintang
9         spasi -= 1 # mengurangi jumlah spasi
10    else: # jika jumlah bintang lebih dari sisi segitiga
11        break # keluar dari loop
```

Gambar 29. Unguided 2 Piramida bintang

pada gambar ?? menunjukkan code untuk membuat piramida bintang menggunakan while loop dan if-else statement.

A. Inisialisasi Variabel

- a. Variabel sisi yang memiliki value 20 digunakan untuk sisi segitiga.
- b. Variabel count yang memiliki value 1 digunakan untuk jumlah buntang.
- c. Variabel space yang memiliki value dibagi 2 (bulat) untuk jumlah spasi

B. Proses

a. While True

Diguakan untuk melakukan perulangan tanpa batas.

b. If count <= sisi

If statement dengan kondisi count kurang dari atau sama dengan sisi segitiga maka jalankan proses:

- `print(" " * spasi , "*" * count)`
digunakan untuk mencetak segitiga dengan “ ” sebanyak value variabel spasi dan “*” sebanyak value variabel count.
- `count += 2`
digunakan untuk menambahkan variabel count dengan 2.
- `spasi -= 1`
digunakan untuk mengurangi variabel spasi dengan 1.

c. Else

Digunakan jika if statement tidak ada yang benar atau False. Statement ini akan menjalankan proses break statement untuk menghentikan infinite loop dari while loop dan mengakhiri proses code.

3.2.3. Hasil

```
kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/unguided 2.py"  
      *  
     ***  
    *****  
   *********  
  **********  
 ****  
*****  
*****  
*****  
*****  
*****
```

kamize@server:~/web/alpro/week 6\$ █

gambar 30. Hasil Unguided 2 Piramida bintang

3.3. Unguided 3: sum_number_xxxx.py

3.3.1. Rules

- A. Buatlah sebuah list dengan nama numbers, isikan list tersebut nilai 7,8,9,10,11,12,13,6,5,4,3,2,1
- B. Jumlahkan semua nilai tersebut
- C. Gunakan for

3.3.2. Jawaban

alpro - unguided 3.py

```
1  # Membuat list dengan nama numbers
2  numbers = [7, 8, 9, 10, 11, 12, 13, 6, 5, 4, 3, 2, 1]
3
4  # Menjumlahkan semua nilai dalam list menggunakan for loop
5  total = 0
6  for number in numbers:
7      total += number
8
9  # Menampilkan hasil penjumlahan
10 print("Jumlah semua nilai dalam list:", total)
```

Gambar 31. Unguided 3

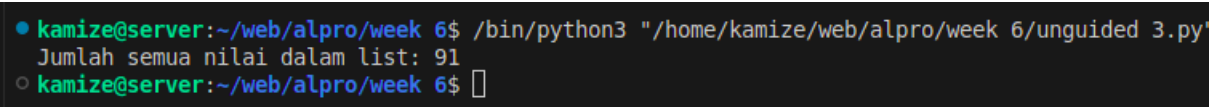
Program Python pada gambar tersebut bertujuan untuk menjumlahkan semua nilai yang terdapat dalam sebuah list. Pertama, program mendeklarasikan sebuah list bernama `numbers` yang berisi sejumlah bilangan bulat, yaitu `[7, 8, 9, 10, 11, 12, 13, 6, 5, 4, 3, 2, 1]`. Setelah itu, program menginisialisasi variabel `total` dengan nilai awal `0`, yang akan digunakan untuk menyimpan hasil akumulasi penjumlahan.

Selanjutnya, program menggunakan perulangan `for` untuk mengiterasi setiap elemen dalam `list numbers`. Dalam setiap iterasi, nilai dari elemen saat ini

(number) akan ditambahkan ke dalam variabel total. Dengan cara ini, program akan menjumlahkan seluruh elemen dalam list satu per satu hingga proses perulangan selesai.

Terakhir, program mencetak hasil penjumlahan dengan menggunakan fungsi `print()`, yang akan menampilkan teks "Jumlah semua nilai dalam list:" diikuti oleh nilai total yang telah dihitung. Dengan demikian, program ini merupakan contoh dasar penggunaan list, perulangan, dan operasi penjumlahan dalam bahasa pemrograman Python.

3.3.3. Hasil



```
• kamize@server:~/web/alpro/week 6$ /bin/python3 "/home/kamize/web/alpro/week 6/unguided 3.py"
Jumlah semua nilai dalam list: 91
○ kamize@server:~/web/alpro/week 6$
```

Gambar 31. Hasil Unguided 3

4. Ringkasan

Dalam laporan praktikum Algoritma dan Pemrograman Komputer Modul 6 dengan topik utama Repetition atau perulangan dalam bahasa pemrograman Python. Ada dua jenis utama perulangan yang dibahas: `for` dan `while`. Perulangan `for` digunakan saat jumlah iterasi sudah pasti (counted loop), sedangkan `while` digunakan untuk perulangan yang belum pasti (uncounted loop).

Selain itu, laporan juga membahas penggunaan control statements seperti `break`, `continue`, dan `pass`. `Break` digunakan untuk menghentikan perulangan, `continue` untuk melewati iterasi tertentu dan langsung ke iterasi selanjutnya, dan `pass` untuk mengosongkan blok kode agar tidak dieksekusi.

Beberapa contoh program diberikan dalam laporan. Salah satunya adalah kalkulator sederhana berbasis teks yang berjalan dalam perulangan tak hingga (`while True`). Pengguna dapat melakukan operasi matematika dasar (+, -, *, /) dan keluar dari program dengan memasukkan operator "0". Program ini juga menangani kesalahan input dengan `try-except`.

Contoh lainnya adalah program membuat piramida bintang, yang memanfaatkan `while` loop dan `if-else`. Program mencetak pola segitiga simetris dengan bintang, menggunakan manipulasi variabel spasi dan jumlah bintang untuk setiap baris.

Contoh terakhir adalah program Python yang menjumlahkan semua nilai dalam list [7, 8, 9, 10, 11, 12, 13, 6, 5, 4, 3, 2, 1] menggunakan perulangan `for`. Program mendeklarasikan variabel total sebagai penampung hasil penjumlahan, lalu menambahkan elemen satu per satu ke dalam total, dan mencetak hasil akhirnya.

Secara keseluruhan, laporan ini menunjukkan pemahaman yang baik tentang dasar-dasar perulangan, penerapan logika pemrograman, serta penanganan kondisi khusus dalam

Python. Ini menjadi dasar penting bagi pengembangan program yang lebih kompleks di masa depan, terutama dalam struktur kontrol dan logika algoritma.