

Laporan Praktikum

Algoritma dan Pemrograman

Perulangan Dengan While



Asisten :

Pahril Dwi Saputra	11221056
Zaky Dio Akbar Pangestu	11221050

Disusun Oleh :

Muchammad Maulana	10241043
Eagan Ferdian	10241025
Annisa Muthia Zahra	20241010
Dinar Kanthi Amalia	20241022
Ciara Khalisha Aqilah	20241016
Yulia Puspitasari	20241088

17 September 2024

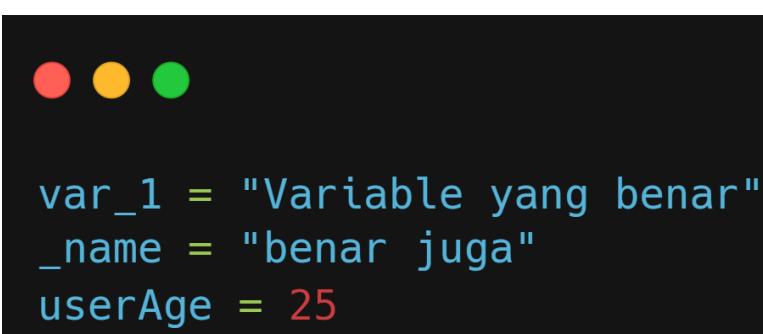
Dasar Teori

1. Variabel

Variabel adalah sebuah nama simbolik yang merujuk pada lokasi penyimpanan di memori komputer, di mana data dapat disimpan, diakses, dan dimanipulasi. Variabel memungkinkan program untuk bekerja dengan data secara dinamis.

a. Deklarasi dan Penugasan:

Dalam Python, deklarasi dan penugasan variabel dilakukan menggunakan operator penugasan (`=`). Misalnya:



```
var_1 = "Variable yang benar"
_name = "benar juga"
userAge = 25
```

- ❖ Karakter pertama harus berupa huruf atau garis bawah/underscore `_`
- ❖ Karakter selanjutnya dapat berupa huruf, garis bawah/underscore `_` atau angka
- ❖ Karakter pada nama variabel bersifat sensitif (case-sensitif). Contoh, variabel `namaDepan` dan `namadepan` adalah variabel yang berbeda.

b. Jenis Data dan Tipe Variabel:

Variabel dapat menyimpan berbagai jenis data. Tipe data dasar dalam Python meliputi:

- ❖ **Integer (int)**: Bilangan bulat, contoh: `x = 5`
- ❖ **Float (float)**: Bilangan pecahan, contoh: `y = 3.14`

-
- ❖ **String (str)**: Teks, contoh: `name = "Alice"`
 - ❖ **Boolean (bool)**: Nilai True atau False, contoh: `flag = True` Dan lain-lain.

a. Penggunaan Operator Penugasan :

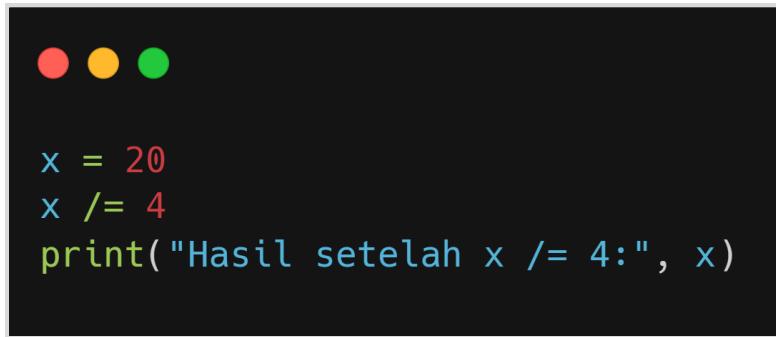
Operator penugasan tidak hanya digunakan untuk memberikan nilai awal pada variabel tetapi juga untuk melakukan operasi matematika atau bitwise secara bersamaan dengan penugasan:

- `+=` (Penambahan dan penugasan): `x += 5` setara dengan `x = x + 5`
- `-=` (Pengurangan dan penugasan): `x -= 3` setara dengan `x = x - 3`
- `*=` (Perkalian dan penugasan): `x *= 2` setara dengan `x = x * 2`
- `/=` (Pembagian dan penugasan): `x /= 4` setara dengan `x = x / 4`
- `%=` (Modulus dan penugasan): `x %= 2` setara dengan `x = x % 2`
- `**=` (Pangkat dan penugasan): `x **= 3` setara dengan `x = x ** 3`

Contoh penggunaan:



```
x = 17
x %= 5
print("Hasil setelah x %= 5:", x)
```



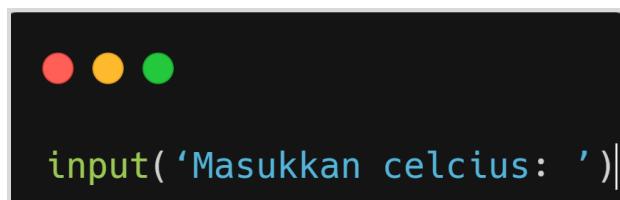
```
x = 20
x /= 4
print("Hasil setelah x /= 4:", x)
```

2. Input dan Output dalam Pemrograman

a. Input

Input adalah data atau informasi yang dimasukkan ke dalam sistem komputer atau program untuk diproses lebih lanjut. Dalam konteks pemrograman, input bisa datang dari berbagai sumber, seperti keyboard, file, sensor, atau bahkan jaringan.

Dalam dunia teknologi, input merujuk pada data, informasi, atau sinyal yang dimasukkan ke dalam suatu sistem untuk diproses lebih lanjut. Contoh penggunaan dalam pemrograman bahasa python :



```
● ● ●
input('Masukkan celcius: ')|
```

b. Output

Output adalah hasil yang diberikan oleh program setelah diproses berdasarkan input yang diterima dan logika yang telah didefinisikan dalam kode. Output bisa berupa data yang ditampilkan di layar, ditulis ke file, dikirim ke perangkat lain, atau dikirim melalui jaringan.

Output dalam bahasa pemrograman adalah hasil yang diberikan oleh program setelah dijalankan berdasarkan input dan logika yang telah didefinisikan dalam kode. Contoh penggunaan:



```
● ● ●
print ("hallow,nama saya muchammad maulana")|
```

3. Operasi Aritmatika

Operator Aritmatika adalah konstruksi dalam pemrograman yang digunakan untuk melakukan perhitungan matematis dengan operand atau nilai. Dalam python, Anda dapat menggunakan berbagai operator aritmatika untuk melakukan operasi matematika dasar. Berikut adalah penjelasan rinci mengenai setiap operator aritmatika di Pyhton.

Operator aritmatika yaitu:

Operator	Simbol	Contoh	Penjelasan
Penambahan	+	1 + 1	Menjumlahkan nilai dari operan kiri menggunakan operan kanan.
Pengurangan	-	1 - 1	Mengurangkan nilai dari operan kiri menggunakan operan kanan.
Perkalian	*	1 * 1	Mengalikan nilai dari operan kiri menggunakan operan kanan.
Pembagian	/	1 / 1	Membagi nilai dari operan kiri menggunakan operan kanan.
Modulus	%	1 % 1	Menghitung sisa bagi nilai dari operan kiri menggunakan operan kanan.
Pangkat	**	2 ** 2	Memangkatkan nilai dari operan kiri menggunakan operan kanan.
Floor	//	2 // 2	Membagi nilai dari operan kiri menggunakan operan kanan lalu dibulatkan kebawah.

Contoh penggunaan dalam pemrograman bahasa python:

```
● ● ●
a = 5
b = 3
result = a + b
print(f"Hasil penjumlahan dari a+b adalah {result}")

#Output Hasil penjumlahan dari a+b adalah 8
```

4. Konversi Tipe Data

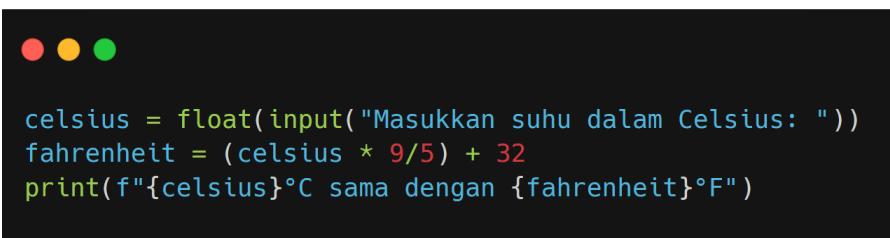
Konversi tipe data adalah proses mengubah nilai dari satu tipe data menjadi tipe data lain. Ini berguna saat anda perlu memastikan bahwa data yang diproses sesuai dengan tipe data yang diharapkan oleh operasi atau fungsi tertentu. Dalam python, konversi tipe data sering dilakukan menggunakan fungsi *built-in*.

a. Fungsi Konversi Tipe Data.

Berikut adalah fungsi konversi tipe data yang umum digunakan di Phyton :

- **int()**: Mengubah nilai menjadi tipe data integer (bilangan bulat).
- **float()**: Mengubah nilai menjadi tipe data float (bilangan pecahan).
- **str()**: Mengubah nilai menjadi tipe data string (teks).
- **bool()**: Mengubah nilai menjadi tipe data boolean (True atau False).

Contoh penggunaan salah satu tipe data dalam pemograman bahasa Phyton:



```
celsius = float(input("Masukkan suhu dalam Celsius: "))
fahrenheit = (celsius * 9/5) + 32
print(f"{celsius}°C sama dengan {fahrenheit}°F")
```

5. Eksekusi Kondisional

Eksekusi kondisional adalah struktur pemilihan yang digunakan untuk mengeksekusi perintah berdasarkan kondisi tertentu. Jika kondisi bernilai benar, perintah tersebut akan diabaikan. Stuktur kondisional juga dapat menggunakan pernyataan IF-Else. IF-Else untuk menguji kondisi dan mengekskusi blok kode yang sesuai.

a. Eksekusi Kondisional Bersarang

Eksekusi kondisional bersarang adalah bentuk dari if. Didalam If. Pada bentuk bersarang,, kode yang ada di dalam blok yang dieksekusi karena prasyarat tertentu kemudian menemui blok kondisi yang baru.

Contoh penggunaan dalam pemrograman bahasa Phyton:

```
umur = 20
kewarganegaraan = "indonesia"

if umur >= 18:
    if kewarganegaraan = "indonesia":
        print ("boleh mengikuti pemilu")
    else:
        print ("Tidak boleh mengikuti pemilu")
else:
    print ("boleh cukup umur untuk memilih")
```

b. Ekesusi Kondisional Majemuk

Ekseskusi kondisional majemuk adalah proses dimana dua atau lebih ekspresi boolean digabungkan menggunakan operator kogika seperti AND, OR, atau NOT untuk menentukan apakah suatu blok kode harus dijalankan atau tidak.

Contoh penggunaan dalam pemrograman bahasa Phyton:

```
nilai_akhir = 85
kehadiran = 75
nilai_tugas = 80

if (nilai_akhir >= 80 and nilai_tugas >= 70) or kehadiran >=
90:
    print("Anda mendapatkan penghargaan")
else:
    print("Anda tidak memenuhi syarat untuk mendapatkan
penghargaan")
```

6. Operator Komparasi

Operasi komparasi atau ada yang menyebutnya juga relasi, adalah operator yang digunakan untuk memunculkan hasil perbandingan dari kedua nilai. Disebut juga operator relasi karena operator tersebut bertujuan mencari relasi dari 2 nilai. Operator komparasi relasi biasanya digunakan untuk perumpamaan.

Contoh penggunaan dalam pemrograman bahasa Phyton:

```
● ● ●

#kelompok7

nilai = 85

if nilai >= 75:
    print("Lulus")
else:
    print("Tidak lulus")
```

7. While

Perulangan ‘while’ dalam Pyhton adalah metode yang efektif untuk mengendalikan eksekusi berulang berdasarkan kondisi tertentu. Perulangan ini akan berlangsung selama kondisi yang diberikan menghasilkan nilai “True” dan akan berhenti ketika kondisi tersebut menjadi ‘False’. Intruksi while digunakan untuk membuat statement berulang selama kondisi syarat diberikan bernilai benar. Jika kondisi bernilai syarat maka statement tidak akan diulang. Perulangan while termasuk encountered loop.

Perulangan ‘While’ sangat berkaitan dengan variasi boolean, atau logical statement. Karena penentuan kapan suatu blok kode akan diulang-ulang ditinjau dari True or False dari suatu pernyataan logika. Jika ekspresi bernilai True, maka statement dijalankan dan akan berlanjut pada eksekusi statement selanjutnya

a. Deklarasi dan Penugasan :

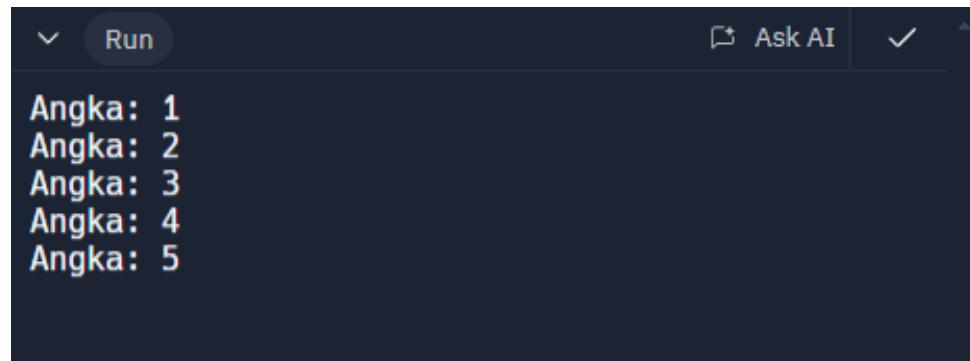
Dalam Python, deklarasi untuk perulangan ‘while’ adalah sebagai berikut :

```
● ● ●  
while kondisi:  
    # blok kode yang akan dijalankan selama kondisi  
    bernilai True
```

- Kata kunci ‘while’ diikuti oleh kondisi yang akan di evaluasi.
- Jika bernilai benar atau ‘True’, blok kode di dalam loop akan terus dijalankan.
- Jika bernilai salah atau ‘False’, loop akan berhenti.

b. Contoh implementasi sederhana:

```
● ● ●  
# Contoh perulangan while  
i = 1  
while i <= 5:  
    print("Angka:", i)  
    i += 1
```



The screenshot shows a code editor interface with a dark theme. At the top, there are three colored window control buttons (red, yellow, green). Below them is a toolbar with a dropdown arrow, a 'Run' button, an 'Ask AI' button, and a checkmark icon. The main area contains Python code and its output. The code is as follows:

```
# Contoh perulangan while
i = 1
while i <= 5:
    print("Angka:", i)
    i += 1
```

The output window displays the following text, indicating that the loop has printed the numbers 1 through 5:

```
Angka: 1
Angka: 2
Angka: 3
Angka: 4
Angka: 5
```

Source Code

Projek_bilangan_ganjil.py

```
1 bilangan = int(input("Masukkan bilangan: "))
2
3 while bilangan > 0:
4     if bilangan % 2 != 0:
5         print(bilangan, end=" ")
6         bilangan -=1
```

Projek_luas_bangun_datar.py

```
1 while True:
2     print("==== MENU ====")
3     print("1. Persegi")
4     print("2. Persegi Panjang")
5     print("3. Segitiga")
6     print("4. Keluar")
7
8     pilihan = input("Pilih menu (1-4): ")
9
10
11     if pilihan == '1':
12
13         sisi = float(input("Masukkan panjang sisi persegi: "))
14         luas = sisi * sisi
15         keliling = 4 * sisi
16         print(f"Luas Persegi: {luas}")
17         print(f"Keliling Persegi: {keliling}\n")
18
19
20     elif pilihan == '2':
21
22         panjang = float(input("Masukkan panjang persegi panjang: "))
23     )
24         lebar = float(input("Masukkan lebar persegi panjang: "))
25         luas = panjang * lebar
26         keliling = 2 * (panjang + lebar)
27         print(f"Luas Persegi Panjang: {luas}")
28         print(f"Keliling Persegi Panjang: {keliling}\n")
29
30
```

```

31 elif pilihan == '3':
32     alas = float(input("Masukkan alas segitiga: "))
33     tinggi = float(input("Masukkan tinggi segitiga: "))
34     sisi1 = float(input("Masukkan sisi pertama segitiga: "))
35     sisi2 = float(input("Masukkan sisi kedua segitiga: "))
36     sisi3 = float(input("Masukkan sisi ketiga segitiga: "))
37     luas = 0.5 * alas * tinggi
38     keliling = sisi1 + sisi2 + sisi3
39     print(f"Luas Segitiga: {luas}")
40     print(f"Keliling Segitiga: {keliling}\n")
41
42
43 elif pilihan == '4':
44     print("Terima kasih! Program selesai.")
45     break
46
47
48 else:
49     print("Pilihan tidak valid. Silakan coba lagi.\n")

```

Projek_mencari_nilai_faktorial.py

```

1  hasil = int(input("Masukkan angka: "))
2  faktorial = 1
3  penjabaran = ""
4
5  i = hasil
6  while i > 0:
7
8
9      faktorial *= i
10     if i == 1:
11         penjabaran += str(i)
12     else:
13         penjabaran += str(i) + " x "
14         i -= 1
15
16 print(f"{hasil}! = {penjabaran} = {faktorial}")

```

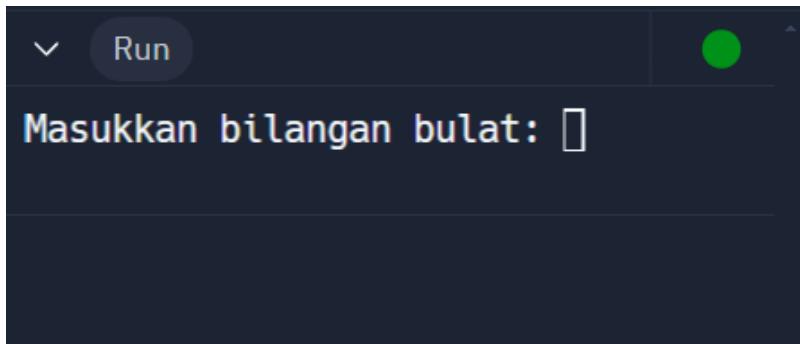
implementasi.py

```
1 hijau = "\033[92m"
2 Merah = "\033[91m"
3 Reset = "\033[0m"
4 nama_pemain = input("Masukkan nama petualangmu: ")
5 nyawa = 100
6 skor = 0
7
8 print(f"\nSelamat datang, {nama_pemain}! Kesehatanmu: {nyawa}.
9 Ayo kita mulai petualangan!")
10
11 while nyawa > 0:
12     print("\nKamu berada di sebuah gua. Ada dua jalan.")
13     print("1. Jalan ke kiri")
14     print("2. Jalan ke kanan")
15     pilihan = input("Pilih jalan (1/2): ")
16
17     if pilihan == "1":
18         print("\nKamu jalan ke kiri dan nemu harta karun!")
19         skor += 10
20         print(f"Skor kamu: {skor}")
21
22     elif pilihan == "2":
23         print("\nKamu jalan ke kanan dan ketemu monster!")
24         aksi = input("(melawan/lari): ")
25         if aksi == "melawan":
26             nyawa -= 20
27             print(f"Kesehatanmu berkurang! Kesehatan tersisa:
28 {nyawa}")
29             elif aksi == "lari":
30                 print("Kamu berhasil kabur, tapi kehilangan 5
31 poin!")
32                 skor -= 5
33                 print(f"Skor kamu: {skor}")
34             else:
35                 print("Aksi gak dikenali!")
36
37     else:
38         print("Pilihan gak valid! Coba pilih 1 atau 2.")
39
40     lanjut = input("Mau lanjut petualangan? (ya/tidak): ")
41     if lanjut.lower() != "ya":
42         break
43
44 print(f"\nPetualanganmu selesai, {nama_pemain}.")
45 print(f"Skor akhir: {skor}. Kesehatan akhir: {nyawa}.")
46 if nyawa <= 0:
47     print("Sayang sekali, kamu udah kalah!")
```

```
else:  
    print("Makasih udah main!")
```

Screenshot

Projek_bilangan_ganjil.py



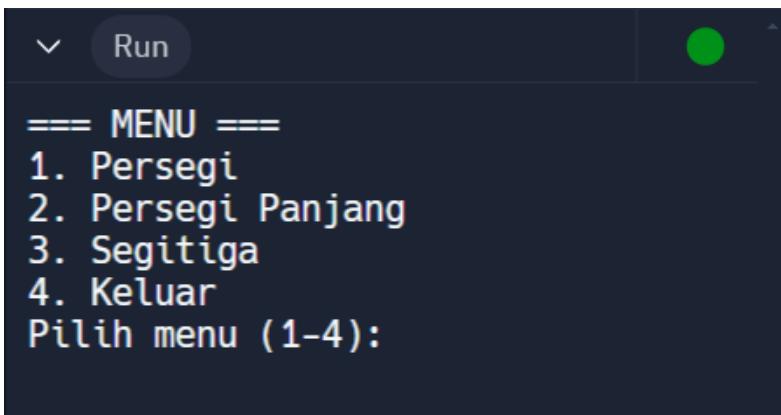
```
Masukkan bilangan bulat: 
```

Tampilan awal program, pengguna diminta untuk memasukkan bilangan bulat.

```
Masukkan bilangan bulat: 20
19 17 15 13 11 9 7 5 3 1
```

Setelah memasukkan bilangan bulat, bilangan ganjil akan ditampilkan.

Projek_luas_bangun_datar.py



```
== MENU ==
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4): 
```

Tampilan awal program, pengguna diminta untuk memilih menu bangun datar apa yang akan dihitung luasnya..

```
Run
```

```
--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4): 1
Masukkan panjang sisi persegi: 12
```

Contoh Setelah memilih bangun datar Persegi (opsi 1) , Pengguna diminta memasukkan panjang sisi persegi.

```
--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4): 1
Masukkan panjang sisi persegi: 12
Luas Persegi: 144.0
Keliling Persegi: 48.0

--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4):
```

Setelah memasukkan panjang sisi persegi , Luas dan Keliling Persegi akan di tampilkan, setelah itu pengguna akan kembali ke menu awal.

```
--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4): 2
Masukkan panjang persegi panjang: 12
Masukkan lebar persegi panjang: 20
```

Contoh Setelah memilih bangun datar Persegi Panjang (opsi 2) , Pengguna diminta memasukkan panjang sisi persegi panjang dan lebar persegi panjang.

```
--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4): 2
Masukkan panjang persegi panjang: 12
Masukkan lebar persegi panjang: 20
Luas Persegi Panjang: 240.0
Keliling Persegi Panjang: 64.0

--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4):
```

Setelah memasukkan panjang dan lebar persegi panjang, Luas dan Keliling Persegi panjang akan ditampilkan, setelah itu pengguna akan kembali ke menu awal.

```
--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4): 3
Masukkan alas segitiga: 10
Masukkan tinggi segitiga: 18
Masukkan sisi pertama segitiga: 12
Masukkan sisi kedua segitiga: 14
Masukkan sisi ketiga segitiga: 16
```

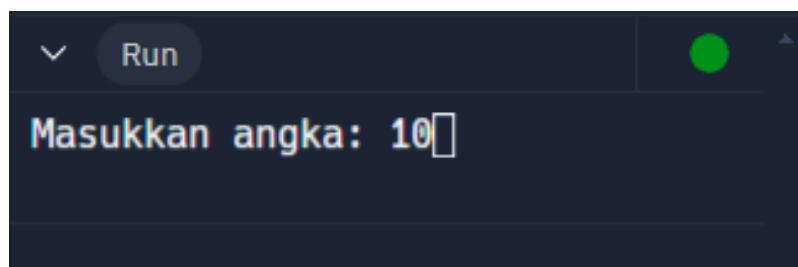
Contoh Setelah memilih bangun datar Segitiga (opsi 3) , Pengguna diminta untuk memasukkan alas,tinggi,sisi pertama,kedua,dan ketiga segitiga.

```
--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4): 3
Masukkan alas segitiga: 10
Masukkan tinggi segitiga: 18
Masukkan sisi pertama segitiga: 12
Masukkan sisi kedua segitiga: 14
Masukkan sisi ketiga segitiga: 16
Luas Segitiga: 90.0
Keliling Segitiga: 42.0

--- MENU ---
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Keluar
Pilih menu (1-4):
```

Setelah memasukkan alas,tinggi,sisi pertama,kedua,dan ketiga segitiga, Luas dan Keliling Segitiga akan ditampilkan, setelah itu pengguna akan kembali ke menu awal.

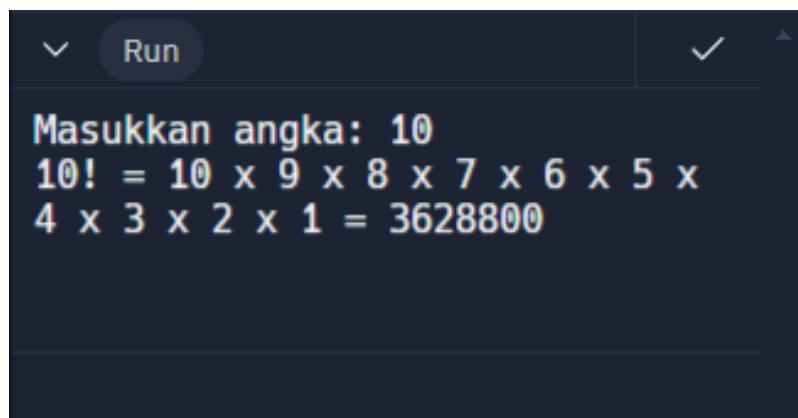
Projek_mencari_nilai_faktorial.py



```
Masukkan angka: 10
```

A screenshot of a terminal window. At the top, there is a toolbar with a dropdown arrow, the word "Run" in a button, and a green circular progress indicator. Below the toolbar, the text "Masukkan angka: 10" is displayed in white on a black background. There is a small rectangular input field next to the text.

Tampilan awal program, pengguna diminta untuk memasukkan angka.



```
Masukkan angka: 10
10! = 10 x 9 x 8 x 7 x 6 x 5 x
4 x 3 x 2 x 1 = 3628800
```

A screenshot of a terminal window. At the top, there is a toolbar with a dropdown arrow, the word "Run" in a button, and a green checkmark icon. Below the toolbar, the text "Masukkan angka: 10" is displayed in white. Underneath it, the factorial calculation for 10 is shown: "10! = 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 3628800".

Setelah memasukkan angka, nilai faktorial dari angka yang di input akan ditampilkan.

Pembahasan

Projek_bilangan_ganjil.py

No.	Penjelasan.
1	Program meminta input dari pengguna,
3	Memulai loop yang akan terus berjalan selama nilai bilangan lebih besar dari 0.
4	Mengecek apakah nilai bilangan adalah bilangan ganjil. Hal ini dilakukan dengan memeriksa sisa bagi bilangan dengan 2. Jika sisa bagi tidak sama dengan 0, bilangan adalah ganjil.
5	Jika kondisi bilangan ganjil benar, maka nilai bilangan dicetak ke konsol.
6	Mengurangi nilai bilangan sebanyak 1. Ini memastikan bahwa loop dapat memeriksa bilangan berikutnya pada iterasi berikutnya.

Projek_luas_bangun_datar.py

Projek_mencari_nilai_faktorial.py

No.	Penjelasan.
1	Mengambil input dari pengguna dan mengonversinya menjadi tipe data integer, yang disimpan dalam variabel hasil.
2	Menginisialisasi variabel faktorial dengan nilai 1, yang akan digunakan untuk menyimpan hasil perhitungan faktorial.
3	Menginisialisasi variabel penjabaran sebagai string kosong, yang akan digunakan untuk menyimpan penjabaran dari proses perhitungan faktorial.
5	Menginisialisasi variabel i dengan nilai dari hasil, yang digunakan sebagai penghitung dalam loop
6	Memulai loop while yang akan berjalan selama nilai i lebih besar dari 0.
9	Mengalikan nilai faktorial dengan nilai i untuk menghitung faktorial.
10	Memeriksa apakah nilai i sama dengan 1.

11	Jika i sama dengan 1, tambahkan nilai i ke variabel penjabaran.
12	Jika i tidak sama dengan 1, lakukan langkah berikut.
13	Tambahkan nilai i dan string " x " ke variabel penjabaran.
14	Mengurangi nilai i sebesar 1 untuk melanjutkan perulangan berikutnya.
16	Mencetak hasil faktorial dan penjabaran dari proses perhitungan ke konsol.

implementasi.py

No.	Penjelasan.
1-3	Mendeklarasikan code warna ANSI Hijau Merah dan Reset.
4	Mengambil nama petualang dengan variabel nama_pemain
5	Menginisialisasi variabel nyawa dengan nilai awal 100, yang menunjukkan kesehatan pemain..
7	Menginisialisasi variabel skor dengan nilai awal 0, yang digunakan untuk menghitung poin yang diperoleh pemain.
8	Menampilkan pesan selamat datang kepada pemain, termasuk nama dan kesehatan awal.
9	Memulai loop yang akan terus berjalan selama nilai nyawa pemain lebih besar dari 0.
10	Menampilkan informasi tentang lokasi pemain saat ini di dalam game.
11	Menampilkan pilihan pertama kepada pemain.
12	Menampilkan pilihan kedua kepada pemain.
13	Mengambil input dari pemain untuk memilih jalan yang ingin diambil.
14	Memeriksa apakah pilihan pemain adalah "1".
15	Jika pemain memilih jalan kiri, menampilkan pesan bahwa mereka menemukan harta karun.
16	Menambahkan 10 poin ke skor pemain setelah menemukan harta karun.
17	Menampilkan skor terkini pemain setelah menemukan harta karun.
18	Memeriksa apakah pilihan pemain adalah "2".
19	Jika pemain memilih jalan kanan, menampilkan pesan bahwa mereka bertemu monster.
20	Mengambil input dari pemain tentang apa yang ingin dilakukan (melawan atau lari).
21	Memeriksa apakah aksi yang dipilih pemain adalah "melawan".
22	Mengurangi nyawa pemain sebesar 20 jika mereka memilih untuk melawan.

23	Menampilkan pesan tentang sisa kesehatan pemain setelah bertarung.
25	Memeriksa apakah aksi yang dipilih pemain adalah "lari".
27	Menampilkan pesan bahwa pemain berhasil melarikan diri tetapi kehilangan 5 poin.
27	Mengurangi skor pemain sebesar 5 karena melarikan diri dari monster.
28	Menampilkan skor terkini pemain setelah melarikan diri.
30	Menangani aksi yang tidak dikenali dan Menampilkan pesan jika aksi yang dimasukkan tidak valid.
31	Menampilkan pesan jika aksi yang dimasukkan tidak valid.
33	Menanyakan pemain apakah mereka ingin melanjutkan petualangan atau tidak.
34	Memeriksa apakah jawaban pemain bukan "ya"
36	Menghentikan loop jika pemain memilih untuk tidak melanjutkan petualangan.
37	Menampilkan pesan bahwa petualangan pemain telah selesai.
38	Menampilkan skor akhir dan sisa kesehatan pemain setelah permainan selesai.
39	Memeriksa apakah pemain kalah.
40	Menampilkan pesan bahwa pemain kalah jika nyawa 0 atau kurang.
41	Menangani kasus di mana pemain masih memiliki nyawa dan menampilkan pesan terima kasih kepada pemain setelah permainan selesai.

Kesimpulan

Hasil Program data yang dibuat diatas sangatlah berguna, bermanfaat dan mempersingkat waktu karena perulangan While pada pemrograman bisa dipergunakan untuk mengeksekusi suatu aktivitas yang sama berulang kali dengan jumlah yang ditentukan, menjalankan baris kode di dalam blok kodennya secara terus menerus selama masih memenuhi ekspresi yang sudah ditentukan sebelumnya yang berarti ia akan terus mengulang selama kondisi bernilai True.

Saran

Menurut kami selama pelaksanaan praktikum, kendala yang masih sering dirasakan adalah pengajaran yang berlangsung terlalu cepat, serta mahasiswa yang duduk di bagian belakang sering kesulitan mendengar materi yang disampaikan. Selain itu, banyak mahasiswa di bagian belakang yang terhalang pandangannya, sehingga tidak dapat melihat papan tulis dengan jelas. Masih banyak mahasiswa yang merasa takut untuk bertanya atau mengakui bahwa mereka tidak memahami materi, yang pada akhirnya dapat menghambat pengeroaan tugas yang diberikan. Oleh karena itu, untuk mengatasi masalah ini, kami menyarankan agar mahasiswa lebih berani dan memiliki kesadaran untuk menyampaikan keluhan kepada asisten dosen apabila merasa materi tidak terdengar di bagian belakang, penjelasan terlalu cepat, atau merasa kesulitan melihat papan tulis dengan jelas. Hal ini penting agar semua mahasiswa dapat memahami materi dengan baik dan mengerjakan tugas tanpa hambatan.

Daftar Pustaka

Alfarisy, G. A. F. (n.d.). *Algoritma Pemrograman*. Institut Teknologi Kalimantan

Anonymous. (n.d.). *Definisi dan Pentingnya dalam Pemrograman*. Diakses dari.

<https://ikatandinas.com/arti-kata-variabel-definisi-dan-pentingnya-dalam-pemrograman/>

Anonymous. (n.d.). *Variabel Python - Belajarpython*. Situs Open Source Tutorial

Pemrograman Python Bahasa Indonesia. Diakses dari.

<https://belajarpython.com/tutorial/variabel-python/>

Anggun, D. (n.d.). *Pengertian input, proses, output, dan contohnya*. Diakses dari.

<https://tambahpinter.com/pengertian-input-proses-output-dan-contohnya/>

Windra. (n.d.). *Pemrograman dasar: Perintah kondisional if pada Python*. Pemrograman dasar: Perintah kondisional if pada Python. Diakses dari.

<https://kodepy.com/pemrograman-dasar-perintah-kondisional-if-pada-python>

Sharianja. (2015, Maret 14). *Struktur If-Then dan If-Then-Else*. Sharianja.

<https://www.uniksharianja.com/2015/03/struktur-if-then-dan-if-then-else.html>

Anonymous. (n.d.). *Percabangan if-else*. SKO Dev.

<https://sko.dev/wiki/percabangan-if-else>

Anonymous. (n.d.). Panduan Teknis Python While Loop. RevoU.

<https://revou.co/panduan-teknis/python-while-loop\>

Anonymous. (n.d.). Perulangan While pada Python. Jago Ngoding.

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