

# Python Programming with IDLE

Raja Oktovin P. Damanik

July 18, 2021

- 1 Print and Input**
- 2 Data and Type**
- 3 Variables and Assignment**
- 4 Expression, Statement, and Using modules**
- 5 Branching**
- 6 Iteration**
- 7 String**
- 8 Function**
- 9 List, Set, Tuples, and Dictionary**
- 10 Object and Class**
- 11 Exercise**

## **11.1 Latihan untuk Bagian 1-5**

1. Dalam satu hari, ada 86.400 detik ( $24 \times 60 \times 60$ ). Diberikan sebuah bilangan bulat dalam rentang 1-86.400, program mencetak hasil konversi bilangan tersebut yang awalnya bersatuan detik ke dalam jam, menit, dan detik. Sebagai contoh, 70.000 detik adalah 19 jam, 26 menit, dan 40 detik.

2. Jika panjang dari dua sisi sejajar dari sebuah trapesium adalah  $X$  meter dan  $Y$  meter, berturut-turut, dan tingginya adalah  $H$  meter, berapakah luas dari trapesium? Buat program Python untuk mencetak luasnya.
3. Bunga sederhana dihitung dengan mengalikan nilai utama, banyak tahun, dan nilai bunganya, lalu dibagi 100. Buat program untuk menghitung bunga sederhana pada nilai utama sebesar Rp 100.000.000,00 untuk durasi 5 tahun dengan bunga 12.5%.
4. Pandang sebuah segitiga dengan panjang sisi 3, 7, dan 9. Hukum kosinus mengatakan bahwa diberikan tiga sisi segitiga  $a$ ,  $b$ , dan  $c$ , sudut  $C$  di antara sisi  $a$  dan  $b$ , berlaku  $c^2 = a^2 + b^2 - 2ab \cos(C)$ . Buat program Python untuk menghitung besar sudut di ketiga segitiga.
5. (Menggunakan modul) Python memiliki ratusan modul yang dapat digunakan oleh pemrogramnya. Berikut adalah tantangan untuk Anda: cari sebuah modul yang dapat Anda impor yang menghasilkan tanggal hari ini sehingga Anda bisa cetak tanggal tersebut. Gunakan mesin pencari favorit Anda untuk mencari modul apa yang Anda butuhkan dan bagaimana cara menggunakannya. Pada akhirnya, tugas Anda adalah untuk melakukan hal berikut:
 

```
>>> print("Today's date is:", X)
Today's date is: 2009-05-22
```
6. Body Mass Index (BMI) adalah sebuah bilangan yang dihitung dari berat dan tinggi seseorang. Berdasarkan Centers for Disease Control, nilai BMI merupakan indikator yang cukup bisa dipercaya untuk kondisi lemak di tubuh seseorang. BMI tidak mengukur lemak tubuh secara langsung, namun riset sudah menunjukkan bahwa hasilnya berkorelasi dengan pengukuran lemak tubuh secara langsung, seperti melalui pembobotan di bawah air dan *dual energy X-ray absorptiometry*. Rumus BMI adalah  $\text{berat}/\text{tinggi}^2$  di mana *berat* bersatuan kilogram dan *tinggi* bersatuan meter.
  - (a) Buat sebuah program yang meminta berat dan tinggi pengguna dalam satuan metrik (kilogram dan meter), lalu cetak BMI-nya.
  - (b) Buat program yang meminta berat dalam pon dan tinggi dalam inci, lalu mengonversi nilainya ke dalam metrik, dan mencetak nilai BMI-nya.
7. Buat sebuah program yang meminta pengguna untuk memasukkan sebuah bilangan bulat yang menyatakan suatu nilai Rupiah (kelipatan 1000). Program akan menghitung kombinasi dengan paling sedikit kertas yang

digunakan yang nilainya dalam Rupiah sama dengan nilai yang dimasukkan pengguna. Sebagai contoh, nilai Rp 178.000,00 dapat diperoleh dengan 1 lembar Rp 100.000,00, 1 lembar Rp 50.000,00, 1 lembar Rp 20.000,00, 1 lembar Rp 2.000,00, dan 1 lembar Rp 1.000,00. Lembar kertas yang tersedia adalah:

- Rp 100.000,00
- Rp 50.000,00
- Rp 20.000,00
- Rp 10.000,00
- Rp 5.000,00
- Rp 2.000,00
- Rp 1.000,00.

## 11.2 Exercise for Section 5 and 6

1. Pada latihan sebelumnya, Anda diminta untuk menghitung BMI seseorang. Lengkapi program Anda dengan mencetak di kategori apa BMI tersebut berada berdasarkan standar berat badan CDC berikut:

BMI	Standar Berat
Di bawah 18.5	Underweight
18.5-24.9	Normal
25.0-29.9	Overweight
30.0 ke atas	Obese

2. Buat sebuah program yang menghitung banyaknya bilangan ganjil, bilangan genap, dan bilangan kuadrat sempurna dari 2 hingga 25 (inklusif).
3. Sederhanakan program berikut sehingga ia tidak menggunakan perintah `break` ataupun `continue`, namun logikanya masih ekivalen.

```
counter = 0
while True:
    if counter >= 100:
        break
    elif counter % 2 == 0:
        counter += 1
        continue
    else:
```

```

    counter *= 2
    continue

```

4. (a) Buat sebuah program yang meminta sebuah bilangan bulat positif – sebut saja  $X$  – dan mencetak hasil jumlah  $X$  bilangan bulat positif pertama dimulai dari 1. Yaitu, jika  $X = 5$ , jumlahnya adalah 15 karena  $1 + 2 + 3 + 4 + 5 = 15$ .  
 (b) Lengkapi program Anda sehingga ia mencetak operasi aritmetika yang relevan. Sebagai contoh, untuk  $X = 5$ , program Anda mencetak  $1 + 2 + 3 + 4 + 5 = 15$ .
5. Buat sebuah program yang meminta sebuah bilangan bulat dan mencetak bilangan bulat tersebut; namun jika yang dimasukkan bukan bilangan bulat, program meminta ulang hingga pengguna memasukkan bilangan bulat. Berikut adalah contoh sesinya:
 

```

Input an integer: abc
Error: try again
Input an integer: 4a
Error: try again
Input an integer: 2.5
Error: try again
Input an integer: 123
The integer is: 123
      
```
6. Sebuah bilangan bulat positif disebut *sempurna* jika hasil jumlah semua faktornya yang kurang dari dirinya sama dengan dirinya sendiri. Sebagai contoh, 6 merupakan bilangan sempurna, karena  $1 + 2 + 3 = 6$ . Buatlah sebuah program yang meminta pengguna memasukkan sebuah bilangan bulat positif dan memeriksa apakah bilangan tersebut merupakan bilangan sempurna atau bukan.
7. Buat sebuah program yang memeriksa apakah sebuah bilangan asli  $N$  merupakan bilangan prima atau bukan. Salah satu pendekatan sederhana adalah mengecek semua bilangan dari 2 hingga  $N$ , namun sebenarnya pengecekan hanya perlu sampai  $\sqrt{N}$ . Buat sebuah program yang memeriksa keprimaan tanpa melakukan pengecekan yang tidak perlu.
8. Berikut adalah sebuah teka-teki.

	S	E	N	D
+	M	O	R	E
<hr/>				
M	O	N	E	Y

Digit apa yang dapat disubstitusi ke huruf-huruf tersebut sehingga ekspresi penjumlahan di atas benar? Buat sebuah program untuk menyelesaikan teka-teki ini. Petunjuk: Pendekatan *brute force* dapat digunakan – cari semua kemungkinan.

9. Buat sebuah program yang meminta pengguna memasukkan bilangan bulat positif  $n$  dan mencetak  $n$  suku pertama dari barisan Fibonacci. Program harus memeriksa bahwa  $n$  merupakan bilangan bulat positif. [Petunjuk: Dua bilangan pertama dari barisan Fibonacci adalah 1 dan 1 dan suku ke- $n$  adalah hasil jumlah suku ke  $n-1$  dan suku ke- $n-2$ . Sebagai contoh, enam suku pertama barisan Fibonacci adalah 1, 1, 2, 3, 5, 8.]
10. Buat sebuah program yang menjumlah semua digit-digit dari sebuah bilangan bulat positif. Jika hasil jumlahnya lebih dari satu digit, program mengulangi proses ini terus menerus hingga hasil jumlah yang didapatkan satu digit. Sebagai contoh 827 memiliki hasil jumlah  $8 + 2 + 7 = 17$  yang memiliki hasil jumlah  $1 + 7 = 8$ , yang merupakan jawaban akhir Anda.
11. Saya berpikir tentang sebuah bilangan dari 1 hingga 100. Anda dan saudara Anda bertengkar demi mendapatkan sepotong kue. Buatlah sebuah program yang memilih sebuah bilangan acak antara 1 hingga 100 dan meminta ke dua orang pengguna masing-masing sebuah bilangan. Pengguna yang masukannya lebih dekat dengan bilangan rahasia menang. Pastikan program Anda mencetak bilangan rahasia tersebut juga selain mencetak juga siapa yang menang.
12. Buat program yang memungkinkan Anda bermain suit *rock, paper, scissor* melawan komputer. Impor modul `random` untuk memilih pilihan komputer.

### 11.3 Latihan untuk String

1. Diberikan string `'abcdefghij'`. Tuliskan sebuah kode yang akan mencetak string berikut menggunakan *slicing* pada string:
  - (a) `'jihgfedcba'`
  - (b) `'adgj'`
  - (c) `'igeca'`.
2. Buatlah program yang menerima sebuah kalimat dalam bahasa Inggris lalu menghitung banyaknya huruf kapital, huruf kecil, digit, dan tanda baca pada kalimat tersebut.

3. Buat sebuah program yang menerima nama pertama dan nama akhir dari seseorang, yang mana nama terakhir hanya terdiri dari satu kata kemudian mencetak dalam satu baris gabungan namanya dengan setiap kata pada nama pertama disingkat menggunakan titik. Sebagai contoh, jika nama pertamanya adalah **Raja Oktovin Parhasian** dan nama akhirnya **Damanik**, program mencetak **R. O. P. Damanik**.
4. Anda membuat sebuah akun dan harus membuat kata sandi. Kata sandi harus memenuhi syarat berikut:
  - (a) Kata sandi harus terdiri dari minimal 6 karakter dan maksimal 20 karakter.
  - (b) Kata sandi harus mengandung minimal satu huruf kapital, satu huruf kecil, dan satu bilangan.

Buat sebuah program yang meminta pengguna memasukkan sebuah kata sandi dan memeriksa apakah kata sandi yang dimasukkan valid atau tidak. Jika kata sandinya valid, cetak pesan konfirmasi. Jika tidak, berikan juga pesan konfirmasi, namun minta ulang kata sandi yang baru dimasukkan hingga valid.

5. Buat sebuah program yang mencetak tabel perkalian untuk bilangan 0 hingga 12. Semua perhitungan aritmetika yang dibutuhkan di tabel dilakukan oleh komputer. Lakukan pengaturan format (dengan `format-string`) tabel agar tanda kali, tanda sama dengan, dan bilangan yang muncul secara rata pada tabelnya.
6. Berdasarkan sebuah riset tentang kemampuan otak manusia yang dilakukan oleh peneliti di Cambridge University, urutan kemunculan huruf di dalam sebuah kata tidak penting; yang penting hanyalah bahwa huruf pertama dan terakhir kata tersebut berada di posisi yang benar. Sisanya dapat diacak-acak dan Anda tetap dapat membacanya tanpa masalah. Ini karena pikiran manusia tidak membaca sebuah kata secara huruf per huruf, namun sebuah kata secara utuh. Keren, bukan? Buatlah sebuah program yang meminta sebuah kata dalam bahasa Indonesia dan mengacak huruf-hurufnya kecuali huruf pertama dan terakhir. *Petunjuk: Gunakan modul `random` yang ada di Python.*
7. Buatlah sebuah program yang memungkinkan pengguna bermain HANGMAN. Gunakan karakter untuk mencetak status HANGMAN. String dengan tiga tanda petik ''' akan berguna untuk menggambar HANGMAN-nya. *Petunjuk: Gambarkan seluruh status HANGMAN sebagai sebuah gambar menggunakan string, satu string gambar untuk masing-masing status.*

## 11.4 Latihan untuk Fungsi

1. Buat sebuah program yang menerima sebuah kalimat dalam bahasa Indonesia (sebuah string) dan mencetak banyaknya huruf vokal dan huruf konsonan di dalam kalimat tersebut. Fungsi tersebut tidak me-**return** apapun. Perhatikan bahwa kalimat tersebut dapat mengandung tanda baca seperti spasi, titik, koma, tanda seru, dan lain-lain.
2. Buat sebuah program yang menerima sebuah string dalam format

`"MM/DD/YYYY HR:MIN:SEC"`

dan mencetak:

- DD/MM/YYYY
- HR:MIN:SEC
- Apakah waktunya AM atau PM.

Validasi masukan wajib dilakukan. Misalnya, jika pengguna memasukkan "122/04/1990 13:12:12", string ini tidak valid sebab hanya mungkin terdapat 12 bulan dalam satu tahun. Pikirkan semua kemungkinan eror yang dapat terjadi pada masukan, dan buat program untuk menanganinya. Fungsi ini tidak me-**return** apa-apa.

3. Buat sebuah program yang menerima sebuah nominal Rupiah (kelipatan 1000), lalu mencetak semua kemungkinan kombinasi uang kertas yang jumlah nilainya sebesar nominal tersebut jika seandainya uang kertas yang tersedia adalah:
  - Rp 1.000,00
  - Rp 2.000,00
  - Rp 5.000,00
  - Rp 10.000,00
  - Rp 20.000,00
  - Rp 50.000,00
  - Rp 100.000,00.
4. (a) Buat program yang melakukan hal yang sama dengan sebelumnya, namun hanya menggunakan kombinasi banyak uang kertas yang paling sedikit. Buat agar programnya seefisien mungkin.  
(b) Bagaimana jika ditambahkan uang senilai Rp 75.000,00?