

Dasar-Dasar Pemrograman 1

Lab 03

More on single loops,
nested loops, string type,
string methods &
functions



FAKULTAS
ILMU
KOMPUTER

Dek Depe Menjadi Hacker Bunker



Ilustrasi Bunker. Sumber : <https://i.imgur.com/z8AWriz.png>

Ketika melakukan renovasi rumah, Pak Esde menemukan *bunker* rahasia di dalam rumahnya. Di dalam *bunker* tersebut terdapat sebuah brankas yang membutuhkan pin untuk dapat membukanya. Pak Esde tidak mengetahui susunan pin yang benar untuk membuka brankas tersebut. Namun, setelah meneliti lebih lanjut, Pak Esde menyadari bahwa pin brankas merupakan susunan angka hasil konversi dari desimal (basis-10) menjadi oktal (basis-8). Pak Esde kemudian meminta bantuan

seorang programmer, Dek Depe, untuk memecahkan kode pin brankas. Dek Depe diminta untuk membuat program yang dapat **mengubah sebuah bilangan desimal (basis 10) menjadi bilangan dengan basis 8**.

Ketentuan Program

- Program akan meminta input dari user yakni:
 - **berapa banyak angka yang ingin dikonversi**
 - **angka yang ingin dikonversi pada setiap iterasi**



Hint : Buatlah loop sebanyak yang user minta untuk meminta input bilangan desimal ke n dari user dan konversi bilangan tersebut di dalam setiap iterasi ke n.

- Input user dijamin valid dengan bilangan yang diinput adalah **int**.
- Angka yang ingin dikonversikan **dijamin lebih besar dari 0**.
- Program akan mencetak **hasil konversi** dari input setiap iterasi.
- Hasil konversi **harus** bertipe **String**.
- **Diwajibkan menggunakan String** untuk menyimpan sisa hasil pembagian, sehingga **tidak diperbolehkan** menggunakan **List, Set, dan Dictionary**.
- **Tidak diperbolehkan** menggunakan built-in function Python seperti `bin()`, `oct()`, dan `hex()` untuk melakukan konversi.



Hint : Gunakan [string slicing](#) untuk *reverse* string dan operasi string untuk [concatenate](#) sisa hasil pembagian ke string.

- Cara konversi dengan basis **N** adalah sebagai berikut :

Misalkan, anda ingin mengkonversikan angka **X** ke basis **N**. Caranya adalah dengan **mengumpulkan X % N (modulo)** ke sebuah **string** dari setiap iterasi, lalu **melakukan X // N (floor division)** hingga angka **X** sama dengan **0**. Saat sudah selesai loopnya, **reverse string** sisa-sisa pembagian yang telah anda peroleh. Selesai!

Contoh :

Ubah angka desimal **20** menjadi basis **8** : (`'%'` adalah operator **modulo** dan `'//'` adalah operator **floor division**)

- $20 \div 8 = 2$, sisa **4** $\rightarrow 20 \% 8 = 4$, $20 // 8 = 2$
- $2 \div 8 = 0$, sisa **2** $\rightarrow 2 \% 8 = 2$, $2 // 8 = 0$
- Semua sisanya dikumpulkan dan diurutkan dari yang terakhir menjadi **24**.

Contoh Input/Output



Teks berwarna **merah** adalah masukan dari pengguna.

Contoh input/output 1:

```
Selamat Datang di Bunker Hacker!  
  
Masukkan berapa kali konversi yang ingin dilakukan: 2  
  
Masukkan angka ke-1 yang ingin dikonversikan (dalam desimal): 40  
Hasil konversi desimal ke basis 8 : 50  
  
Masukkan angka ke-2 yang ingin dikonversikan (dalam desimal): 30  
Hasil konversi desimal ke basis 8 : 36  
  
Terima kasih telah menggunakan Bunker Hacker!
```

Penjelasan output 1:

Pengkonversian 40 ke basis 8:

- $40 \div 8 = 5$, sisa **0**
- $5 \div 8 = 0$, sisa **5**
- Semua sisanya dikumpulkan dan diurutkan dari yang terakhir menjadi **50**.

Pengkonversian 30 ke basis 8:

- $30 \div 8 = 3$, sisa **6**
- $3 \div 8 = 0$, sisa **3**
- Semua sisanya dikumpulkan dan diurutkan dari yang terakhir menjadi **36**.

Contoh input/output 2:

```
Selamat Datang di Bunker Hacker!  
  
Masukkan berapa kali konversi yang ingin dilakukan: 1  
  
Masukkan angka ke-1 yang ingin dikonversikan(dalam desimal): 101  
Hasil konversi desimal ke basis 8 : 145  
  
Terima kasih telah menggunakan Bunker Hacker!
```

Penjelasan output 2:

Pengkonversian 101 ke basis 8:

- $101 \div 8 = 12$, sisa **5**
- $12 \div 8 = 1$, sisa **4**
- $1 \div 8 = 0$, sisa **1**
- Semua sisanya dikumpulkan dan diurutkan dari yang terakhir menjadi **145**.

Contoh input/output 3:

Selamat Datang di Bunker Hacker!

Masukkan berapa kali konversi yang ingin dilakukan: **2**

Masukkan angka ke-1 yang ingin dikonversikan (dalam desimal): **33288**

Hasil konversi desimal ke basis 8 : 101010

Masukkan angka ke-2 yang ingin dikonversikan (dalam desimal): **3143**

Hasil konversi desimal ke basis 8 : 6107

Terima kasih telah menggunakan Bunker Hacker!

Penjelasan output 3:

Pengkonversian 33288 ke basis 8 :

- $33288 \div 8 = 4161$, sisa **0** ($33288 // 8 = 4161$, $33288 \% 8 = 0$)
- $4161 \div 8 = 520$, sisa **1** ($4161 // 8 = 520$, $4161 \% 8 = 1$)
- $520 \div 8 = 65$, sisa **0** ($520 // 8 = 65$, $520 \% 8 = 0$)
- $65 \div 8 = 8$, sisa **1** ($65 // 8 = 8$, $65 \% 8 = 1$)
- $8 \div 8 = 1$, sisa **0** ($8 // 8 = 1$, $8 \% 8 = 0$)
- $1 \div 8 = 0$, sisa **1** ($1 // 8 = 0$, $1 \% 8 = 1$)
- Semua sisanya dikumpulkan dan diurutkan dari yang terakhir menjadi **101010**.

Pengkonversian 3143 ke basis 8:

- $3143 \div 8 = 392$, sisa **7**
- $392 \div 8 = 49$, sisa **0**
- $49 \div 8 = 6$, sisa **1**
- $6 \div 8 = 0$, sisa **6**
- Semua sisanya dikumpulkan dan diurutkan dari yang terakhir menjadi **6107**.

Komponen Penilaian

- **50%** Memenuhi *test case*
- **25%** Menggunakan **nested for/while loop** dan **manipulasi string**.
- **5%** Dokumentasi kode.
- **10%** Memenuhi kriteria standar penulisan kode Python*
- **10%** Mengumpulkan dengan format dan penamaan file yang benar

*Standar penulisan kode yang harus dipenuhi yaitu:

1. Indentasi yang konsisten
2. Aturan penamaan variabel mengikuti Python Naming Convention ([sumber](#))
3. Penamaan Module, Class, Method, dan Variabel yang tidak ambigu

Deliverables

Kumpulkan berkas [Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap]_Lab03.py yang telah di-zip dengan format penamaan seperti berikut.

[Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap]_Lab03.zip

Contoh:

A_ABC_2206123456_ThamiEndamora_Lab03.zip

Acknowledgements

- [PPT Strings \(Part 1\) - DDP 1 Gasal 2021/2022](#)
- [PPT Repetition - DDP 1 Gasal 2022/2023](#)
- https://www.w3schools.com/python/python_strings_concatenate.asp
- <https://python-reference.readthedocs.io/en/latest/docs/brackets/slicing.html>