

# Dasar-Dasar Pemrograman 1

## Lab 03

### Strings, Text Files, Exceptions



FAKULTAS  
ILMU  
KOMPUTER

## Makanan Favorit Mahasiswa Baru

Sub-CPMK 4: Mampu memanipulasi rangkaian karakter (strings)

Sub-CPMK 5: Mampu memanipulasi koleksi data berbentuk teks sederhana (text files) serta menangani exceptions



source: [https://www.freepik.com/free-photo/top-view-table-full-food\\_21088367.htm](https://www.freepik.com/free-photo/top-view-table-full-food_21088367.htm)

Dek Depe, seorang mahasiswa baru **Fasilkom**, melakukan survei untuk mengetahui makanan yang disukai teman-teman barunya. Namun, saat melakukan survei, ia memisahkan jawaban dari teman jurusan ilmu komputer dengan jawaban dari teman jurusan sistem informasi. Sekarang, ia memiliki dua daftar makanan favorit dan ia ingin mengetahui makanan apa saja yang ada dalam kedua daftar tersebut.

Sebagai teman yang baik, kamu berencana membantu Dek Depe dengan membuat program yang dapat menampilkan makanan yang ada dalam kedua daftarnya. Selain itu, kamu juga berinisiatif menambahkan spesifikasi program agar dapat menampilkan semua makanan dari dua daftar tersebut. Namun, karena kamu baru belajar sampai materi Strings, kamu hanya mampu menggunakan string dan fungsi atau method dari string.

## Ketentuan Program

- Program akan meminta input dua daftar makanan untuk dilakukan operasi terhadapnya. Input program adalah dua buah file yang masing-masing berisi daftar makanan dengan format .txt output program juga berupa file .txt.
- Program harus memeriksa keberadaan **isi file input**. Jika file tidak ada, cetak di terminal '**Maaf, file input tidak ada**'. Keluaran tersebut tidak perlu dimasukkan ke file output.
- Isi file input berisi kedua daftar makanan. Daftar makanan pertama dan daftar makanan kedua dipisahkan oleh baris baru. Setiap nama makanan dipastikan terdiri dari satu kata dan hanya dipisahkan oleh tanda koma (,). Contoh isi file input.txt:

```
Daftar Makanan 1: makanan1,makanan2,makanan2
Daftar Makanan 2: makanan1,makanan3
```

- Program akan meminta input berupa aksi nomor berapa yang ingin dilakukan. Input pengguna dipastikan sesuai (hanya nomor yang disediakan).
- Output program adalah nama makanan yang dipisahkan oleh tanda koma (,). Contoh: "makanan1,makanan2,makanan3". Jika ada nama makanan yang sama dalam satu daftar, program hanya akan menampilkannya sekali.
- String keluaran semua langkah sebelumnya disimpan ke dalam **file output**.
- Program mencetak pesan penutup dan kembali ke menu utama. Program akan berulang terus kecuali aksi untuk keluar dipilih.
- **Selain untuk membaca isi text file**, operasi hanya boleh memanfaatkan **manipulasi string**. **Penggunaan metode split, join, intersection, dan union dilarang** karena melibatkan tipe data selain string.
- Gunakan fungsi dan *methods* dari string saja. Solusi yang tidak sesuai ketentuan **tidak akan dinilai**.
- Program bersifat **case insensitive**.



Perhatikan bahwa setiap makanan dipisahkan oleh koma sehingga Anda bisa mendapatkan nama tiap makanan dengan mencari koma. Anda dapat menggunakan method find untuk mencari indeks suatu karakter dalam string.

## Test Case



Teks berwarna merah adalah input dari pengguna.

Daftar file untuk test case: [files](#)

### Test case 1:

#### Isi file in1.txt

```
Daftar Makanan 1: cupcake,katsu,perkedel,kwetiau
Daftar Makanan 2: katsu,muffin,pempek,sushi
```

#### Console input/output:

```
Selamat datang! Masukkan dua nama file yang berisi daftar
makanan yang kamu miliki.
```

```
Masukkan nama file input daftar makanan: in1.txt
```

```
Masukkan nama file output: out1.txt
```

```
Apa yang ingin kamu lakukan?
```

```
=====
```

1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar

```
=====
```

```
Masukkan aksi yang ingin dilakukan: 1
```

```
Daftar makanan pertama:
```

```
cupcake,katsu,perkedel,kwetiau
```

```
Apa yang ingin kamu lakukan?
```

```
=====
```

1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar

```
=====
```

```
Masukkan aksi yang ingin dilakukan: 2
```

```
Daftar makanan kedua:
```

```
katsu,muffin,pempek,sushi
```

```
Apa yang ingin kamu lakukan?
```

```
=====
```

1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar

```
=====
```

```
Masukkan aksi yang ingin dilakukan: 3
```

```
Gabungan makanan favorit dari kedua daftar:
```

```
cupcake,katsu,perkedel,kwetiau,muffin,pempek,sushi
```

```
Apa yang ingin kamu lakukan?
```

```
=====
```

1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar

```
=====
```

```
Masukkan aksi yang ingin dilakukan: 4
```

```
Makanan yang sama dari dua daftar:
```

```
katsu
```

```
Apa yang ingin kamu lakukan?
```

```
=====
```

1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar

```
=====
```

```
Masukkan aksi yang ingin dilakukan: 5
```

```
Terima kasih telah menggunakan program ini!
```

```
Semua keluaran telah disimpan pada file out1.txt
```

### Isi file out1.txt

```
Daftar makanan pertama:
```

```
cupcake,katsu,perkedel,kwetiau
```

```
Daftar makanan kedua:
```

```
katsu,muffin,pempek,sushi
```

```
Gabungan makanan favorit dari kedua daftar:
```

```
cupcake,katsu,perkedel,kwetiau,muffin,pempek,sushi
```

```
Makanan yang sama dari dua daftar:
katsu
```

### Penjelasan test case 1:

Ketika pengguna memilih 3, program mencetak gabungan makanan dari kedua daftar. Makanan yang sama hanya ditampilkan satu kali. Ketika pengguna memilih 4, program mencetak makanan yang ada dalam kedua daftar tersebut.

### Test case 2:

#### Isi file in2.txt

```
Daftar Makanan 1: omelet
Daftar Makanan 2: soto
```

### Console input/output:

```
Selamat datang! Masukkan dua nama file yang berisi daftar
makanan yang kamu miliki.
Masukkan nama file input daftar makanan: in2.txt
Masukkan nama file output: out2.txt

Apa yang ingin kamu lakukan?
=====
1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar
=====
Masukkan aksi yang ingin dilakukan: 4

Tidak ada makanan yang sama dari kedua daftar.

Apa yang ingin kamu lakukan?
=====
1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar
=====
Masukkan aksi yang ingin dilakukan: 5
```

```
Terima kasih telah menggunakan program ini!  
Semua keluaran telah disimpan pada file out2.txt
```

### Isi file out2.txt

```
Tidak ada makanan yang sama dari kedua daftar.
```

### Penjelasan test case 2:

Apabila tidak ada makanan yang sama dalam kedua daftar tersebut, program mencetak pesan bahwa tidak ada makanan yang sama.

### Test case 3:

#### Isi file in3.txt

```
Daftar Makanan 1: SaTe,soto,bubur,satE,sOto  
Daftar Makanan 2: piZZA,lasagna,kebab,bubur,piZzA,risol,soTo
```

### Console input/output:

```
Selamat datang! Masukkan dua nama file yang berisi daftar  
makanan yang kamu miliki.  
Masukkan nama file input daftar makanan: file_asal.txt  
Masukkan nama file output: out3.txt  
  
Maaf, file input tidak ada  
  
Selamat datang! Masukkan dua nama file yang berisi daftar  
makanan yang kamu miliki.  
Masukkan nama file input daftar makanan: in3.txt  
Masukkan nama file output: out3.txt  
  
Apa yang ingin kamu lakukan?  
=====
```

1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar

```
=====
```

Masukkan aksi yang ingin dilakukan: 3

Gabungan makanan favorit dari kedua daftar:  
sate,soto,bubur,pizza,lasagna,kebab,risol

Apa yang ingin kamu lakukan?

```

=====
1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar
=====
Masukkan aksi yang ingin dilakukan: 4

Makanan yang sama dari dua daftar:
soto,bubur

Apa yang ingin kamu lakukan?
=====
1. Tampilkan daftar makanan pertama
2. Tampilkan daftar makanan kedua
3. Tampilkan gabungan makanan dari dua daftar
4. Tampilkan makanan yang sama dari dua daftar
5. Keluar
=====
Masukkan aksi yang ingin dilakukan: 5

Terima kasih telah menggunakan program ini!
Semua keluaran telah disimpan pada file out3.txt

```

### Isi file out3.txt

```

Gabungan makanan favorit dari kedua daftar:
sate,soto,bubur,pizza,lasagna,kebab,risol

Makanan yang sama dari dua daftar:
soto,bubur

```

### Penjelasan test case 3:

Ketika nama file input tidak ditemukan, maka program akan meminta input baru. Keluaran untuk file input salah tidak perlu dituliskan pada out3.txt. Meskipun ada nama makanan yang sama dalam daftar, pada output gabungan, program hanya menampilkannya sekali.

### Komponen Penilaian

- 50% Kebenaran fungsionalitas program
- 25% Penerapan manipulasi string dan repetition (for/while loop)

- 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

## Revisi

- Rilis pertama: 28 September 2023
- Revisi 1: menambahkan daftar file untuk test case

## Deliverables

- lab03.py

Kumpulkan lab03.py yang telah di-zip dengan format penamaan seperti berikut.

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

Contoh:

VIN\_A\_2306123456\_DavinaFardya\_Lab03.zip