

## Lab 05 Kelas F: Function, Exceptions, and File Writing



FAKULTAS  
ILMU  
KOMPUTER

**Deadline: Selasa, 8 Oktober 2024, pukul 16:20 WIB (80 menit)**

### Komponen yang Diuji:

#### CPMK:

1. Menguasai konsep-konsep pemrograman umum serta bahasa pemrograman berorientasi objek modern.
2. Apabila diberikan suatu soal pemrograman dasar, mampu memprogram yang memecahkan soal tersebut.

#### Sub-CPMK:

1. Mampu menganalisis dan merancang mekanisme kontrol seperti *selection* dan *repetition* (CPMK 1)
2. Mampu memanipulasi variabel, tipe data (CPMK 1)
3. Mampu memanipulasi rangkaian karakter (*strings*) (CPMK 1, 2)
4. Mampu memanipulasi koleksi data berbentuk teks sederhana (*text files*) serta menangani *exceptions* (CPMK 1, 2)
5. Mampu memanipulasi List

### Versi

Versi	Timestamp	Keterangan	Warna
1	8-10-2024; 15:00 WIB	Rilis Pertama	-

# Bermain dengan Kata bersama DekDepe



Sumber: DALL-E

Hari ini, Dek Depe mengajak kamu untuk membuat 3 fungsi yang bermain dengan kata dan files. Empat fungsi tersebut adalah: **unique\_words**, **count\_words**, dan **word\_freqs**. Setelah itu, kamu diminta untuk membuat **program utama** yang membaca sebuah file teks dan menampilkan frekuensi kemunculan dari setiap kata unik pada teks file tersebut. Yuk, kita bekerja bersama dengan DekDepe!

## Ketentuan Program

### Part I: **Unique Words** (30 Point)

Anda diminta untuk mendefinisikan sebuah fungsi **unique\_words()** yang menerima **list of words** dan mengembalikan **list baru** yang berisi daftar kata unik-nya. Daftar kata unik yang dikembalikan semuanya dalam bentuk **lower-cased**.

```
>>> unique_words(["halo", "HALO", "HaLo"])
["halo"]

>>> unique_words([])
[]

>>> unique_words(["halo", "semua"])
["halo", "semua"]

>>> unique_words(["halo", "semua", "halo"])
["halo", "semua"]
```

```
def unique_words(words):

    # lanjutkan ...
```

## Part II: Count Words (20 Point)

Anda diminta untuk mendefinisikan sebuah fungsi `count_words(word, words)` yang menerima dua buah argument: sebuah kata yang bertipe string, dan sebuah **list of words**. Fungsi kemudian mengembalikan bilangan bulat yang menyatakan berapa kali word muncul pada words. Anda **dilarang menggunakan method `.count()`**

```
>>> count_words("halo", ["halo", "semua", "HALO"])
2

>>> count_words("halow", ["halo", "semua", "halo"])
0

>>> count_words("HALO", ["halo", "semua", "HALo", "ya",
"haLO"])
3
```

```
def count_words(word, words):

    # lanjutkan ...
```

## Part III: Word Frequencies (30 Point)

Di bagian ketiga ini, Anda diminta untuk menggunakan fungsi `count_words(word, words)` di atas untuk menampilkan **daftar frekuensi dari setiap kata-kata**. Definisikan sebuah fungsi **`word_freqs(ws, words)`** yang menerima 2 buah **list of words**, dan mengembalikan list yang berisi frekuensi kemunculan setiap kata pada list **ws**.

```
>>> word_freqs(["halo", "kamu"], ["kamu", "halo", "iya",
"kamu"])
[1, 2]
```

Kata "halo" muncul satu kali, dan kata "kamu" muncul dua kali.

```
>>> word_freqs(["yes", "iya"], ["kamu", "halo", "iya",
"kamu"])
[0, 1]
```

Kata "yes" tidak muncul pada words; dan kata "iya" muncul satu kali.

## Part IV Program Utama (10 Point)

Di bagian akhir ini, kita akan membuat program utama yang membaca sebuah file teks dan menampilkan frekuensi kemunculan dari setiap kata unik pada file teks tersebut. Anda sudah diberikan kode-nya, yaitu sebagai berikut:

```
nama_file = input("nama file = ")

with open(nama_file) as file:
    content = file.read()

    words = content.split()
    unique_ws = unique_words(words)
    freqs_ws = word_freqs(unique_ws, words)

    print("Daftar kata unik:")
    for (w, freq) in zip(unique_ws, freqs_ws):
        print(w, freq)
```

Coba Anda buat sebuah file teks (bebas beri nama apapun) dan diisi dengan beberapa kalimat bebas, lalu jalankan program di atas. Apa yang terjadi? Jika ada kesalahan apa yang harus diperbaiki? **Tuliskan dalam bentuk komentar pada kode Anda**

### Task

Modifikasi program di atas agar ada proses **validasi input** “nama file” dengan **try-except exception**. Jadi, program akan mengulang terus selama nama file yang dimasukkan user tidak ditemukan. Misal, nama file-nya adalah “input.txt”:

```
nama file = halo.txt [enter]
File tidak ditemukan. Ulangi lagi!

nama file = yes.txt [enter]
File tidak ditemukan. Ulangi lagi!

nama file = input.txt [enter]
Daftar kata unik:
...
```

## Ketentuan Umum Program

1. **Dilarang melakukan kerja sama, menyontek, dan tindakan kecurangan lainnya** dengan sesama mahasiswa/i maupun dengan menggunakan **artificial intelligence** dalam mengerjakan lab.
2. Penamaan variabel harus mengikuti Python Naming Convention ([referensi](#)).
3. Penamaan modul, class, method, dan variabel harus jelas dan tidak ambigu ([referensi](#)).
4. Dokumentasikan kode menggunakan *comment*.
5. Deadline pengumpulan adalah pukul **16:20**. Tempat pengumpulan akan ditutup pukul 16:40. Setelah periode tersebut, submisi tidak akan diterima.

## Komponen Penilaian

Komponen penilaian dapat dilihat pada tautan [ini](#)

## Berkas yang Perlu Dikumpulkan

- **lab05.py**

Kumpulkan berkas lab05.py yang telah di-zip dengan format penamaan seperti berikut.

**[KodeAsdos]\_[Kelas]\_[NPM]\_[NamaLengkap]\_Lab05.zip**

Contoh:

FER\_F\_2106701936\_HieronimusFerry\_Lab05.zip