|  |  |
| --- | --- |
| A picture containing text, clipart  Description automatically generated | **Laporan Praktikum Algoritma dan Pemrograman**  Semester Genap 2023/2024 |

|  |  |
| --- | --- |
| **NIM** | **71230995** |
| **Nama Lengkap** | **Alven Tendrawan** |
| **Minggu ke / Materi** | **10/ Dictionary** |

**SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.**

**SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2024**

# BAGIAN 1: MATERI MINGGU INI (40%)

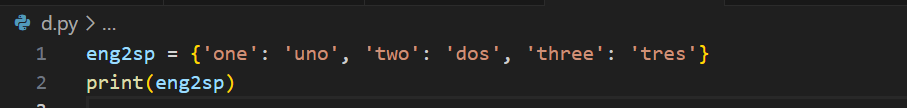
Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

## Tipe Data Dictionary

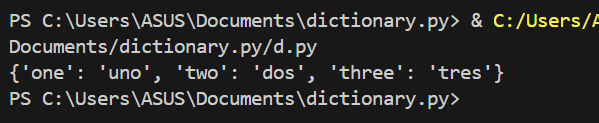
Dictionary adalah kumpulan kunci dan nilai yang saling berhubungan, di mana setiap kunci mengaitkan dengan nilai tertentu. Setiap pasangan kunci:nilai disebut sebagai item. Sebagai ilustrasi, kita bisa membuat dictionary yang memetakan kata-kata dari bahasa Inggris ke bahasa Spanyol, di mana kata-kata tersebut menjadi kunci dan nilai dalam bentuk string, dengan asumsi bahwa setiap kata dalam bahasa Inggris memiliki satu arti dalam bahasa Spanyol. Dictionary mirip dengan list, tapi lebih fleksibel. Setiap item dalam dictionary terdiri dari kunci dan nilainya, dan setiap kunci harus unik. Ini memungkinkan pemetaan antara kumpulan kunci dan nilai.

Fungsi dict digunakan untuk membuat dictionary baru yang kosong. Karena dict merupakan built-in function dari python, maka penggunaanya perlu dihindari sebagai nama variabel.

Sc:

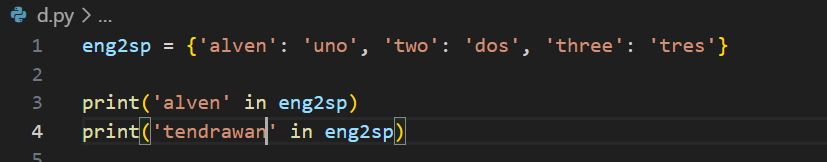


Output:

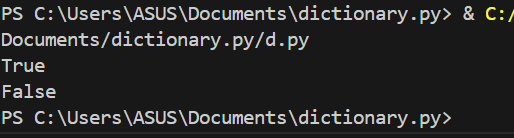


Operator in dalam dictionary mengembalikan nilai benar (true) atau salah (false) sesuai dengan kunci yang ada dalam dictionary. Pada list, operator in menggunakan algoritma pencarian linear, yang berarti waktu pencarian bertambah seiring bertambahnya panjang list. Namun, dalam dictionary, Python menggunakan algoritma Hash Table yang efisien. Algoritma ini memungkinkan operator in untuk memproses dengan cepat tanpa memperdulikan jumlah item di dalam dictionary.

Sc:



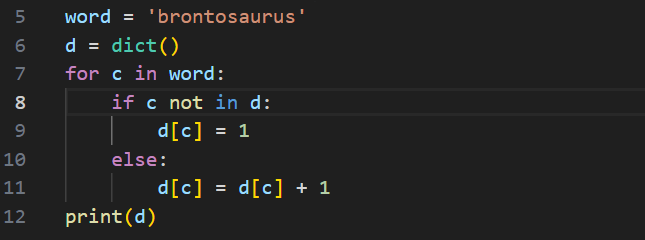
Output:



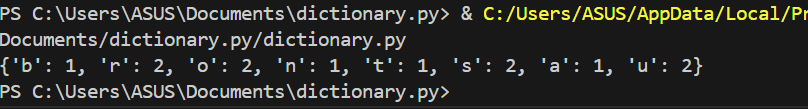
## Dictionary sebagai set penghitung (counters)

Penggunaan model dictionary dalam perhitungan dianggap lebih praktis karena kita tidak perlu mengetahui sebelumnya tentang karakter apa yang akan muncul. Berikut adalah beberapa contoh perhitungan yang dapat dilakukan dengan model dictionary.

Sc:



Output:



Dalam melakukan perhitungan jumlah kemunculan huruf dalam sebuah string, ada beberapa pendekatan yang dapat digunakan:

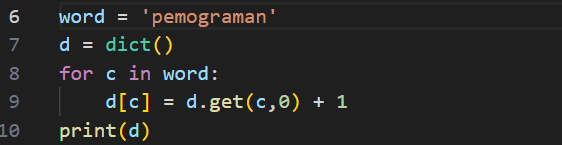
1. Pendekatan pertama melibatkan pembuatan 26 variabel untuk setiap huruf dalam alfabet, kemudian setiap karakter dalam string dimasukkan ke dalam variabel yang sesuai, dan dilakukan penambahan perhitungan yang sesuai dengan setiap karakter. Pendekatan ini menggunakan kondisional berantai untuk mengelola setiap karakter.

2. Pendekatan kedua melibatkan pembuatan sebuah list dengan 26 elemen, di mana setiap karakter dalam string dikonversi menjadi angka menggunakan fungsi bawaan, kemudian angka tersebut digunakan sebagai indeks dalam list untuk menambahkan perhitungan yang sesuai.

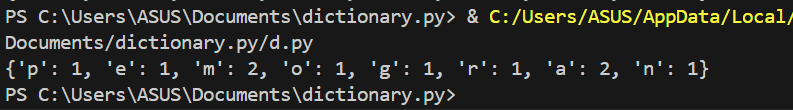
3. Pendekatan ketiga melibatkan pembuatan sebuah dictionary dengan karakter sebagai kunci dan jumlah kemunculannya sebagai nilai yang sesuai. Setiap karakter dalam string ditambahkan sebagai item ke dalam dictionary, dan nilai dari setiap item ditambahkan sesuai dengan kemunculan karakter tersebut.

Dictionary memiliki metode get yang berguna untuk mengambil nilai dari kunci yang spesifik, dan dapat mengembalikan nilai default jika kunci tidak ada dalam dictionary. Jika kunci ditemukan, metode get akan mengembalikan nilai yang terkait dengan kunci tersebut; namun, jika kunci tidak ditemukan, metode ini akan mengembalikan nilai default yang telah ditentukan sebelumnya. Berikut ini adalah contoh penerapannya.

Sc:



Output:



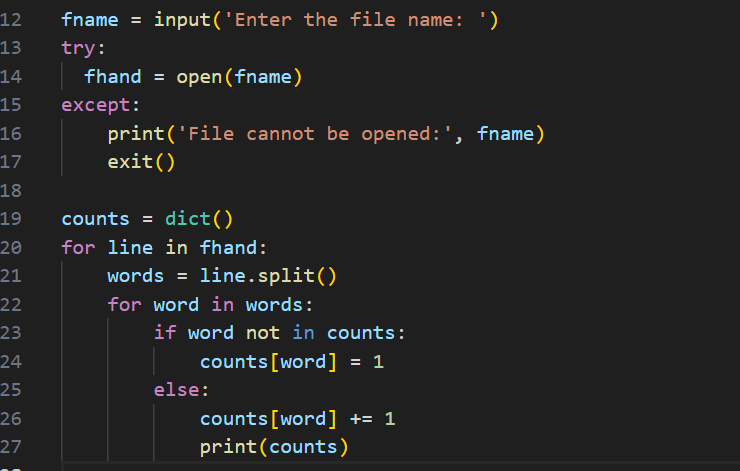
## Dictionary dan File

Salah satu penggunaan umum dari dictionary adalah untuk menghitung frekuensi kemunculan kata-kata dalam sebuah file teks. Misalnya, kita dapat memulai dengan menggunakan sebuah file yang berisi teks yang disederhanakan dari Romeo dan Juliet tanpa tanda baca. Untuk contoh awal, kita akan membaca setiap baris dari file tersebut, memecahnya menjadi daftar kata-kata, dan kemudian mengiterasi melalui setiap kata dalam setiap baris. Dalam proses ini, kita akan menggunakan dictionary untuk menghitung jumlah kemunculan setiap kata.

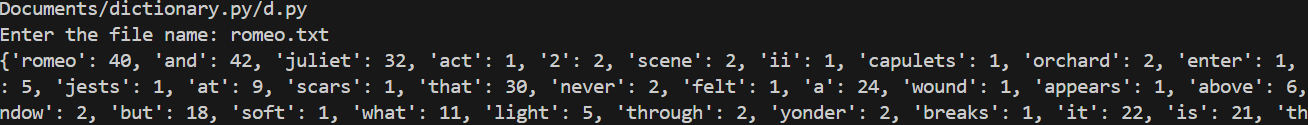
Kita asumsikan bahwa kita menggunakan dua perulangan "for". Perulangan luar memungkinkan kita untuk membaca setiap baris dalam file, dan perulangan dalam memungkinkan kita untuk melakukan iterasi melalui setiap kata yang ada di baris tertentu. Karena fakta bahwa satu perulangan berada di dalam yang lain, pola ini disebut sebagai nested loop.

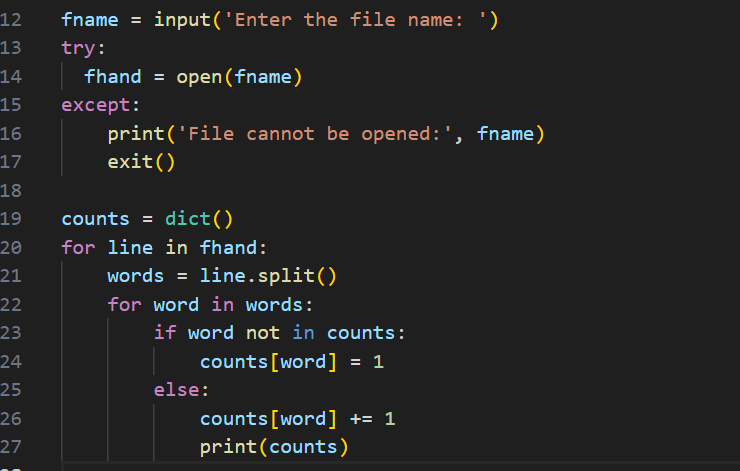
Untuk setiap iterasi dari perulangan luar, perulangan dalam akan dieksekusi. Dalam hal ini, perulangan dalam akan berjalan "lebih cepat" daripada perulangan luar. Setiap kata yang ada di setiap baris file akan dihitung dengan benar berkat kombinasi dari kedua peristiwa ini.

Sc: Bagian Atas Formulir



Output:

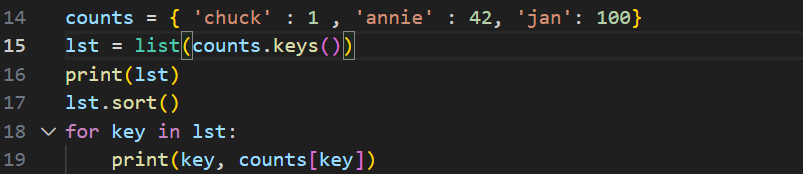


Pada statement else digunakan model penulisan yang lebih singkat untuk menambhakan variable. counts[word] += 1 sama dengan counts[word] = counts[word] + 1. Metode lain dapat digunakan untuk mengubah nilai suatu variabel dengan jumlah yang diinginkan, misalnya dengan menggunakan - =, \* =, dan / =. Ketika kita menjalankan program, akan didapatkan data dump jumlah semua dalam urutan hash.

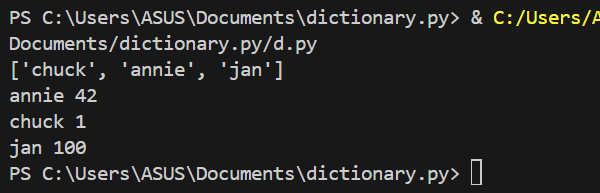
## Looping dan DictionaryBagian Atas Formulir

Untuk mencetak kunci dalam urutan alfabet, langkah pertama adalah membuat daftar kunci di kamus dengan menggunakan metode kunci yang tersedia pada object dictionary. Langkah selanjutnya adalah melakukan pengurutan (sort), list, dan loop melalui sorted list. Langkah terakhir adalah memeriksa setiap kunci dan mencetak pasangan nilai key yang telah diurutkan.

Sc:



Output:



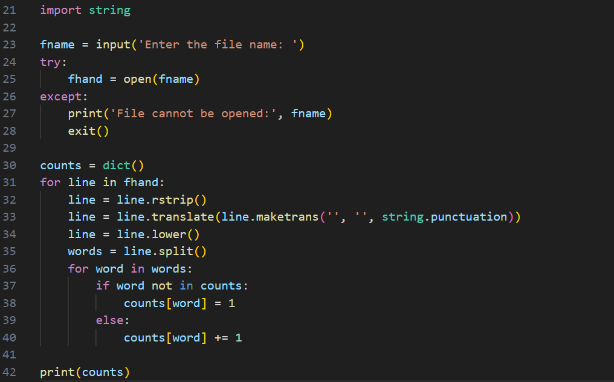
## Advenced Text Parsing

Fungsi "split()" Python secara default membagi string berdasarkan spasi, menjadikan setiap kata sebagai token yang terpisah oleh spasi. Misalnya, kata "soft!" dan "soft" akan dianggap sebagai dua kata yang berbeda, dan keduanya akan dihitung secara terpisah dalam dictionary.

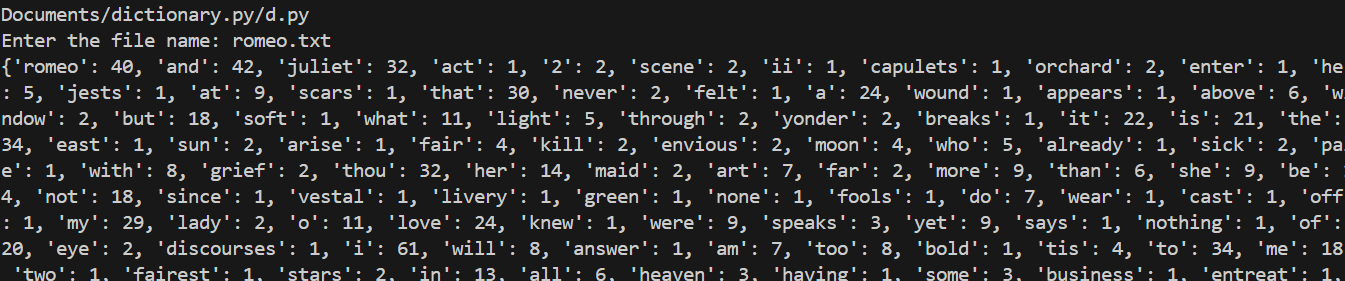
Selain fungsi "split()", Anda juga dapat menggunakan metode string lain seperti "lower()", "punctuation", dan "translate()". Metode string yang paling halus (paling tidak terlihat) adalah metode "translate()".

Dalam metode "translate()", karakter pada "fromstr" diganti dengan karakter pada posisi yang sama pada "tostr", dan semua karakter yang ada dalam "deletetr" dihapus. Jika "fromstr" atau "tostr" kosong, karakter yang sesuai akan diabaikan. Kita tidak akan menentukan "tostr" dalam hal ini, tetapi kita akan menggunakan parameter "deletetr" untuk menghapus semua tanda baca. Python menggunakan karakter mana yang dianggap sebagai "tanda baca" akan diidentifikasi secara otomatis.

Sc:



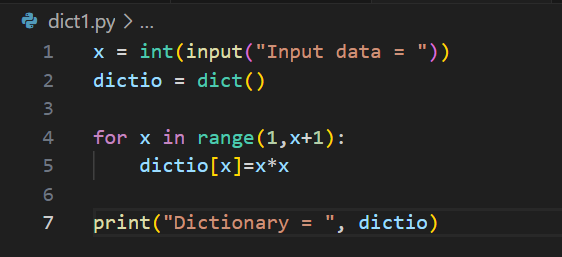
Output:



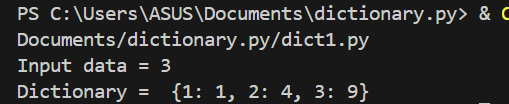
## Kegiatan praktikkum:

Soal 1:

Sc:



Output:

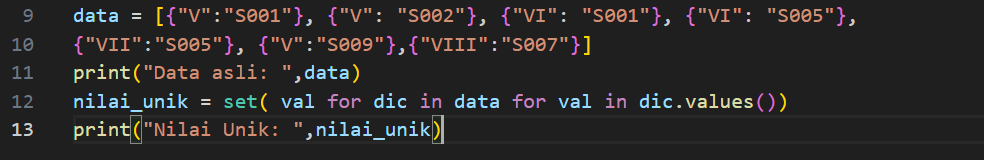


Penjelasan:

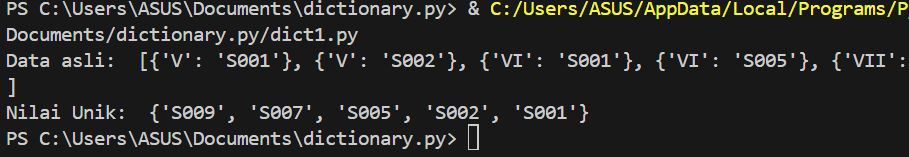
Dalam program ini, pengguna diminta untuk memasukkan angka, kemudian membuat dictionary yang berisi kuadrat dari setiap bilangan bulat dari 1 hingga angka yang dimasukkan, dan akhirnya mencetak dictionary tersebut menggunakan pemahaman dictionary, yang secara sederhana memungkinkan pembuatan dictionary dengan hanya satu baris kode.

Soal 2:

Sc:



Output:



Penjelasan:

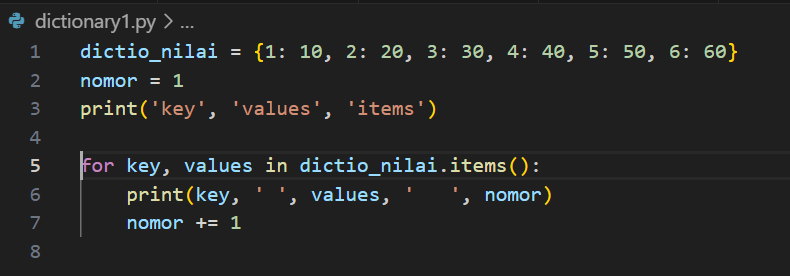
Pada program ini kita dapat menggunakan fungsi values pada dictionary untuk mencari nilai unik kita ambil masing-masing nilai dari anggota dictionary, kemudiah kita kumpulkan dalam satu

# BAGIAN 2: LATIHAN MANDIRI (60%)

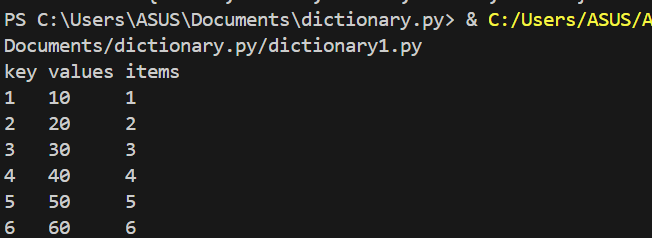
Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

## Soal 1:

Sc:



Output:

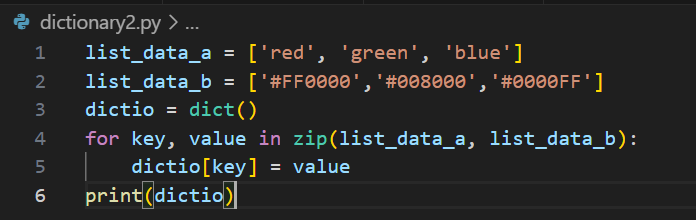


Penjelasan:

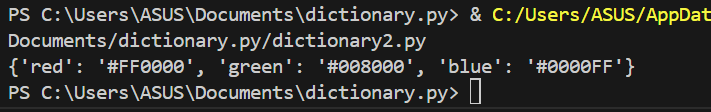
Pada program ini terdapat fungsi dictio\_nilai, yang berisi sejumlah pasangan kunci-nilai. Selanjutnya, variabel nomor diinisialisasi dengan 1. Program menggunakan pernyataan for untuk mengiterasi pasangan kunci-nilai dalam dictio\_nilai pada setiap iterasi. Pada setiap iterasi, kunci, nilai, dan nomor urut pasangan kunci-nilai dicetak. Untuk mengatur nomor urut pasangan selanjutnya, variabel nomor ditingkatkan satu setelah mencetak. Hal ini memungkinkan kita untuk melihat bagaimana pasangan kunci-nilai dalam dictionary diurutkan secara relatif.

## Soal 2:

Sc:



Output:

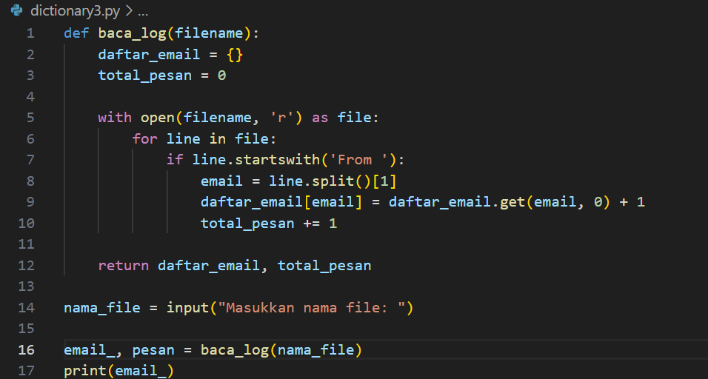


Penjelasan:

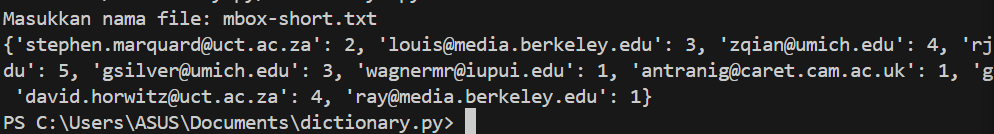
Program ini menggunakan fungsi zip() untuk menghubungkan dua daftar, satu sebagai kunci dan yang lainnya sebagai nilai. Dictionary "dictio" mencakup semua pasangan kunci-nilai, yang berarti bahwa kita dapat langsung mengakses nilai suatu kunci dalam dictionary jika kita ingin tahu nilainya. Proses ini memungkinkan data diakses dan dimanipulasi secara efektif dalam kode dan memungkinkan pengaturan data yang terstruktur. Hubungan kunci-nilai yang diwakili oleh pasangan list\_data\_a dan list\_data\_b dalam situasi ini adalah contoh hubungan konsep yang lebih kompleks atau tersebar luas.

## Soal 3:

Sc:



Output:

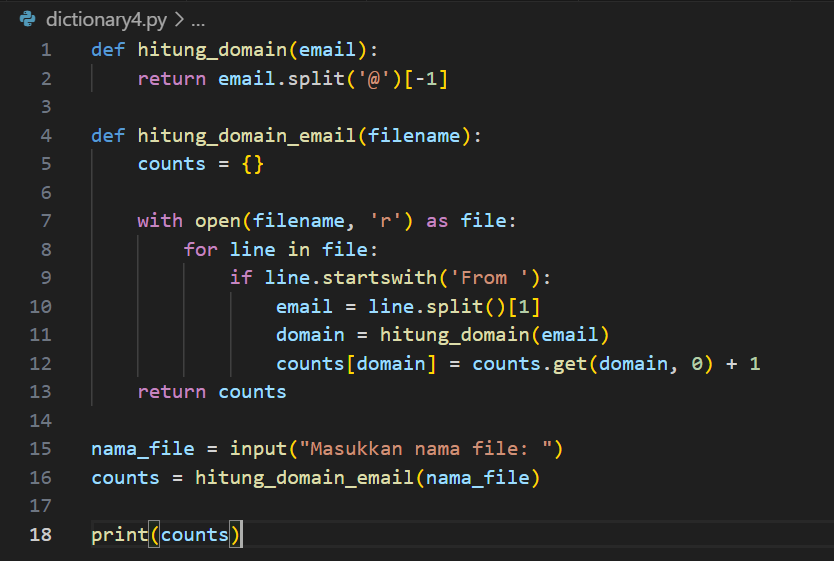


Penjelasan:

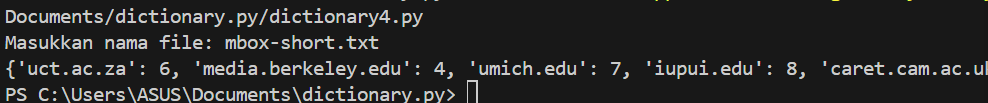
Pada program ini, kita memiliki fungsi **baca\_log** yang membaca file log dan menghitung jumlah pesan dari setiap alamat email yang ditemukan. Setiap alamat email dan jumlahnya disimpan dalam sebuah dictionary. Setelah membaca file, fungsi mengembalikan dictionary tersebut bersama dengan total pesan yang telah dihitung. Kemudian, pengguna diminta untuk memasukkan nama file, dan fungsi **baca\_log** dipanggil dengan nama file tersebut. Hasilnya, dictionary yang berisi alamat email dan jumlahnya dicetak.

## Soal 4:

Sc:



Output:



Penjelasan:

Fungsi hitung\_domain dalam program ini mengambil dan mengembalikan domain dari alamat email. Fungsi hitung\_domain\_email kemudian membaca file log dan menghitung jumlah kemunculan domain dari alamat email yang ditemukan. Hasilnya, kamus yang berisi jumlah kali domain muncul dicetak.

Link github: