|  |  |
| --- | --- |
| A picture containing text, clipart  Description automatically generated |  |

**Laporan Praktikum Algoritma & Pemrograman**

**Semester Genap 2024/2025**

**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.**

|  |  |
| --- | --- |
| **NIM** | **71241140** |
| **Nama Lengkap** | **Anugrah Dwantara** |
| **Minggu ke / Materi** | **12 / Tipe Data Dictionary** |

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2025**

**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.

**Dictionary sebagai Set Penghitung (Counters)**

Tujuan

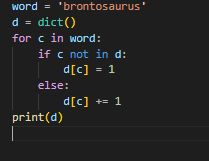
Menggunakan dictionary untuk menghitung frekuensi kemunculan karakter (atau elemen lainnya) dalam data, seperti string atau teks.

Metode Umum

Tiga cara menghitung jumlah huruf yang muncul:

1. Menggunakan banyak variabel untuk tiap huruf.
2. Menggunakan list, lalu konversi karakter ke indeks angka.
3. Menggunakan dictionary, dengan huruf sebagai kunci dan frekuensi sebagai nilai.

Contoh Program (Histrogram Manual)



Output:



## **Dictionary dan File**

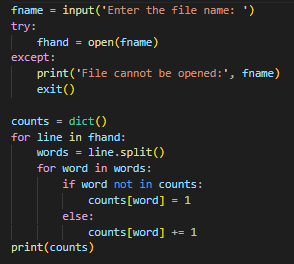
Tujuan

Menggunakan dictionary untuk menghitung jumlah kemunculan kata dalam file teks.

Langkah-langkah Umum

1. Buka file dengan open().
2. Baca setiap baris menggunakan loop.
3. Pisah kata-kata dalam setiap baris dengan .split().
4. Hitung frekuensi setiap kata menggunakan dictionary.

Contoh Program Sederhana



Penjelasan

* Menggunakan loop bersarang: loop luar membaca baris, loop dalam memproses kata.
* Hasil akhir adalah histogram kata (jumlah tiap kata dalam file).

## **Advanced Text Parsing (Pemrosesan Teks Lanjutan)**

Tujuan

Menghitung frekuensi kata dalam file teks yang mengandung tanda baca dan huruf kapital, dengan membersihkannya terlebih dahulu.

Masalah yang Dihadapi:

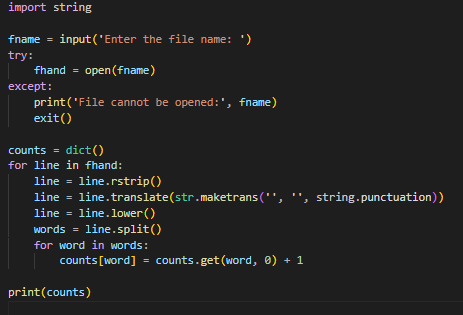
1. Kata dengan tanda baca dianggap berbeda (contoh: soft ≠ soft!).
2. Huruf kapital mempengaruhi pencocokan (Who ≠ who).

Solusi:

Gunakan method string untuk:

* Menghapus tanda baca: str.translate()
* Menjadikan huruf kecil semua: .lower()

Contoh Program:



**Penjelasan Kode:**

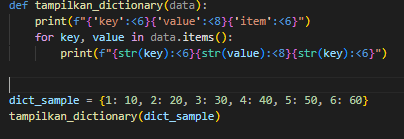
* translate() menghapus tanda baca.
* lower() mengubah semua huruf menjadi kecil.
* split() memecah baris jadi kata.
* counts[word] = counts.get(word, 0) + 1: menghitung frekuensi tiap kata.

# BAGIAN 2: LATIHAN MANDIRI (60%)

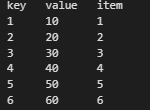
Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

## **SOAL 1**

SC:



Output:

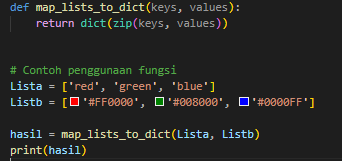


Penjelasan:

* Menampilkan judul kolom: key, value, item.
* :<6 dan :<8 artinya teks tersebut diratakan ke kiri dan memiliki lebar 6 atau 8 karakter.
* Tujuannya agar kolom-kolom tersusun sejajar walaupun data panjangnya berbeda.
* data.items() akan mengembalikan pasangan (key, value) dari dictionary.
* str(...) mengonversi angka ke string agar bisa diformat.
* <6, <8, dll. membuat tiap kolom rata kiri dan seragam lebarnya.

## **SOAL 2**

SC:



Output:

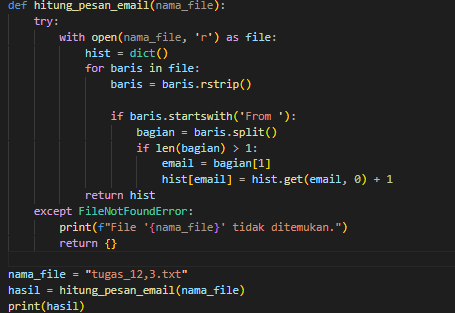


Penjelasan:

* **Parameter keys dan values**  
  Fungsi menerima dua parameter berupa list: keys untuk daftar nama (key dictionary), dan values untuk daftar isi (value dictionary).
* **Menggunakan zip**  
  zip(keys, values) menggabungkan dua list menjadi list pasangan tuple (key, value).
* **Mengubah ke dictionary**  
  dict() mengubah hasil zip tadi menjadi dictionary dengan key dan value sesuai pasangan.
* **Mengembalikan hasil**  
  Fungsi mengembalikan dictionary hasil pemetaan.

## **SOAL 3**

SC:



Output :



Penjelasan :

**Membuka file**  
Program membuka file dengan nama sesuai input pengguna.

**Membaca setiap baris**  
Untuk setiap baris dalam file, program akan:

* Menghapus spasi atau newline di akhir (rstrip()).
* Mengecek apakah baris diawali dengan 'From ' (ingat ada spasi setelah From, berbeda dengan 'From:').

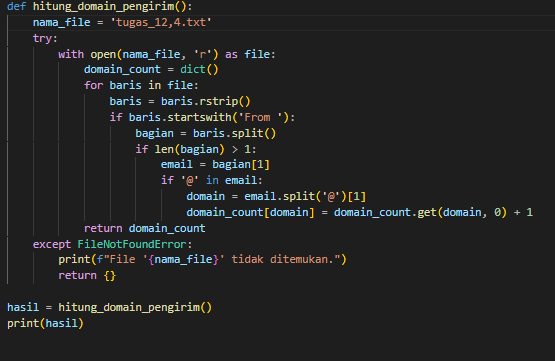
**Mengambil alamat email**  
Baris 'From ' biasanya diikuti oleh alamat email sebagai elemen kedua, jadi program mengambil bagian[1] dari hasil split().

**Membuat histogram**  
Program menggunakan dictionary hist untuk menghitung frekuensi kemunculan tiap email, dengan hist.get(email, 0) + 1.

**Menangani error**  
Jika file tidak ditemukan, program akan mencetak pesan error dan mengembalikan dictionary kosong.

## **SOAL 4**

SC:



Output:



Penjelasan:

* Program baca setiap baris file.
* Cari baris yang dimulai dengan "From " untuk ambil email pengirim.
* Ambil domain dari email dengan memisahkan bagian setelah karakter @.
* Hitung frekuensi kemunculan domain dalam dictionary domain\_count.
* Kembalikan dictionary tersebut.

https://github.com/AnugrahDwantara/LaPrak-AlPro