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

|  |  |
| --- | --- |
| **NIM** | **71230997** |
| **Nama Lengkap** | **Christ Jevicto Ajimas Kirana** |
| **Minggu ke / Materi** | **11 / Tipe Data Tuples** |

**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.

## MATERI 1

* 1. **Tuple Immutable**

Tuple hamper sama dengan list. Elemen dalam tuple dapat merupakan apapun dan akan diberikan indeks berupa integer, namun tuple bersifat immutable yang artinya elemen dalam tuple tidak dapat dirubah. Tuple dapat di compare dan bersifat hasable sehinga dapat dimasukan dalam list sebagai **key** pada dictionary pada python.

Penulisan tuple :

A screen shot of a computer program

Description automatically generated

Untuk menandakan bahwa 1 elemen adalah tuple dengan cara memberi koma di belakang elemen tersebut jika tidak maka akan dianggap sebagai string, jika argument berbentuk string, list, atau tuple akan mengbalikan elemen tuple dengan elemen yang berurutan contohnya pada tuple’dutawacana’.

Tuple merupakan nama dari **constructor**, kita tidak bisa menggunakan nama tuple untuk sebuah variabel. Sebagian besar operator pada list juga bekerja pada tuple, contohnya tanda [] untuk menandakan indeks elemen pada tuple.

Contoh :

A screen shot of a computer program

Description automatically generated

Tuple bersifat immutable, jadi kita tidak bisa mengganti isi tuple, namun isi elemen tuple dapat diganti meskipun tidak bisa dirubah. Contoh:

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

* 1. **Membandingkan Tuple**

Operator compare dapat bekerja pada Tuple dan model skuensial lainnya (list, dictionar, set). Cara kerjanya adalah membandingkan dari elemen pertama jika sama maka akan lanjut ke elemen selanjutnya hingga ditemukan perbedaan. Contoh:

A screenshot of a computer program

Description automatically generated 🡪A screen shot of a computer code

Description automatically generated

Fungsi **sort** pada python bekerja dengan cara yang sama di tuple. Ada fitur yang disebut dengan DSU (**Decorate, Sort, Undercorat**), DSU adalah

* Decorate – urutan (sekuensial) membangun daftar tuple dengan satu atau lebih key pengurutan sebelum eleman dari urutan (sekuensial).
* Sort – list tuple menggunakan sort (fungsi bawaan di python).
* Undercorate – melakukan ekstraksi pada elemen yang telah diurutkan pada satu sekuensial.

Contoh :

A screen shot of a computer

Description automatically generated

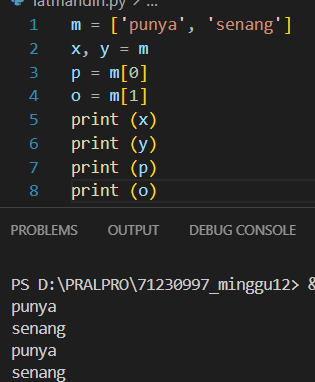
Looping yang pertama akan membuat daftar tuple, yang berisi daftar kata sesuai dengan panjangnya. Fungsi sort akan membandingkan elemen pertama dari list dari panjang kata yang ada dan kemudian akan menuju ke elemen kedua jika kondisi sesuai. fungsi reverse=True digunakan untuk melakukan urutan secara terbalik.

Looping kedua akan membangun daftar tuple dalam urutan alfabet, diurutkan berdasarkan panjangnya. Empat kata dalam kalimat akan diurutkan secara alfabet terbalik, sehingga kata "what" akan muncul sebelum "soft" dalam daftar.

* 1. **Penugasan Tuple**

Python memiliki fitur unik yang memungkinkan penugasan tuple di sisi kiri pernyataan penugasan, sehingga bisa menetapkan beberapa variabel sekaligus secara berurutan. Misalnya, dari dua daftar elemen, kita bisa menetapkan elemen pertama dan kedua ke variabel x dan y dalam satu pernyataan.

Contoh :

 dalam penugasan tuple dapat menjadi dua yaitu contohnya seperti di gambar di variable **x, y** dan variabel **p, o.** Tuple juga bisa untuk menukar nilai dalam satu statement, Contoh :

A screen shot of a computer program

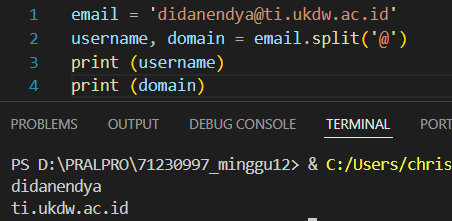
Description automatically generated

Contoh tersebut menunjukkan bahwa kedua statement adalah tuple. Bagian kiri adalah tuple dari variabel, dan bagian kanan adalah tuple dari ekspresi. Tiap nilai di bagian kanan ditugaskan ke variabel yang sesuai di bagian kiri. Semua ekspresi di bagian kanan dievaluasi sebelum penugasan dilakukan. Kita harus memperhatikan jumlah variabel dan jumlah elemen saat memberikan penugasan, karena jika sisi kanan dan kiri tidak sama maka akan terjadi error. Contoh :

A screen shot of a computer

Description automatically generated

Pada sisi kanan biasanya terdapat data sekuensial seperti string, list, atau tuple. Misalnya, untuk membagi alamat email menjadi username dan domain, kita bisa melakukan seperti berikut.



Nilai yang dikembalikan dari split terdiri dari dua elemen yang dipisahkan tanda "@": elemen pertama berisi username dan elemen kedua berisi domain.

* 1. **Dictionaries and Tuple**

Dictionaries mempunyai metode yang disebut *items* untuk mengembalikan nilai list dari tuple

dimana tiap tuple-nya merupakan key-value pair (pasangan kunci dan nilai).A close-up of a code

Description automatically generated

Seperti dictionary umumnya, item yang dikembalikan tidak berurutan. Namun, karena list dari tuple adalah list, kita bisa mengurutkan tuple-tuple tersebut. Mengonversi dictionary menjadi list of tuples memungkinkan kita menampilkan isi dictionary yang diurutkan berdasarkan kunci.

A white background with black and red text

Description automatically generated

List yang muncul diurukan secara ascending berdasarkan alfabet dan key value.

* 1. **Multipenugasan dengan dictionaries**

Mengombinasikan `items`, penugasan tuple, dan `for` memungkinkan iterasi melalui keys dan values dari dictionary dalam satu loop.

A black background with multicolored text

Description automatically generated

Dalam loop ini, terdapat dua variabel iterasi karena `items` mengembalikan list dari tuple (key, value). Penugasan tuple iterasi berulang melalui pasangan key-value pada dictionary. Pada setiap iterasi loop, baik key maupun value akan bergerak ke pasangan key-value berikutnya dalam urutan hash.

Dengan menggabungkan kedua teknik di atas, kita bisa mencetak isi dictionary yang diurutkan berdasarkan nilai setiap pasangan key-value. Langkah pertama adalah membuat list tuple (value, key) menggunakan `items`. Setelah itu, urutkan list berdasarkan value. Kemudian, cetak list baru yang sudah diurutkan dalam urutan terbalik.  
A screen shot of a computer program

Description automatically generatedDengan menyusun list dari tuple yang memiliki nilai sebagai elemen pertama, kita dapat dengan mudah mengurutkan list tersebut dan mendapatkan isi dictionary yang diurutkan berdasarkan nilai-nilainya.

* 1. **Kata yang sering muncul**

Kita akan mencoba menampilkan kata yang sering muncul dalam teks dari Romeo and Juliet Act 2, Scene 2, yang tersedia dalam file "romeo-full.txt". Program akan menggunakan list dari tuple untuk menampilkan 10 kata yang paling sering muncul.

A screen shot of a computer program

Description automatically generated

Program pertama membaca file dan menghitung jumlah kemunculan setiap kata dalam teks. Dengan mencetak counts, list tuple (value, key) dibuat dan diurutkan dalam list dengan urutan terbalik. Nilai pertama digunakan dalam perbandingan. Jika ada nilai yang sama, urutan abjad digunakan sebagai kriteria kedua. Selanjutnya, dilakukan iterasi pada list dan dicetak 10 kata yang paling sering muncul. Ini memberikan output yang sesuai untuk analisis frekuensi.

Analisis data yang kompleks dapat dilakukan dengan hanya menggunakan 19 baris program Python yang mudah dipahami. Hal ini menjadi salah satu alasan mengapa Python dipilih sebagai bahasa yang ideal untuk mengeksplorasi informasi.

* 1. **Tuple sebagai kunci dictionaries**

Tuple adalah hashable sedangkan list tidak. Untuk membuat kunci komposit dalam dictionary, kita dapat menggunakan tuple sebagai kunci. Sebagai contoh, jika ingin membuat direktori telepon yang memetakan pasangan last-name, first-name ke nomor telepon, kita bisa menggunakan tuple sebagai kunci. Dengan asumsi telah didefinisikan variabel last, first, dan nomor. Penulisan pernyataan penugasan dalam dictionary sebagai

berikut:

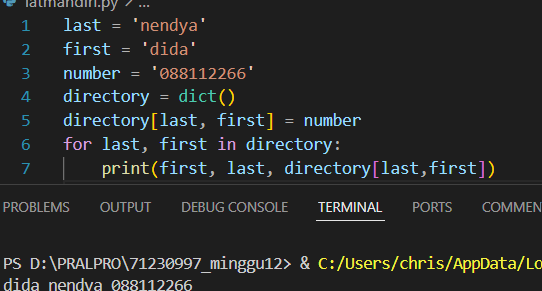
***directory[last,first] = number***

Ekspresi di dalam kurung kotak adalah tuple. Langkah selanjutnya adalah memberikan penugasan tuple pada loop for yang terkait dengan dictionary.

***for last, first in directory:***

***print(first, last, directory[last,first])***

Looping yang terkait dengan kunci pada direktori tersebut adalah tuple. Setiap elemen dari tuple ditugaskan terlebih dahulu, kemudian dilakukan pencetakan nama dan nomor telepon yang sesuai.



## MATERI 2

**2.1 Kegiatan Praktikum**

**Kasus 2.1** Buatlah program yang dapat melakukan proses berikut:

1. Membuat dan mencetak tuple.

2. Membagi tuple kedalam string.

3. Membuat tuple yang berisi string dari sebuah kata.

4. Membuat tuple yang berisi semua, kecuali huruf pertama dari sebuah string

5. Mencetak tuple secara terbalik

Contoh:

kota: ('Jakarta', 'Jogja', 'Surabaya')

kota[0]: Jakarta

kota[1]: Jogja

kota[2]: Surabaya

str1: Jakarta

str2: Jogja

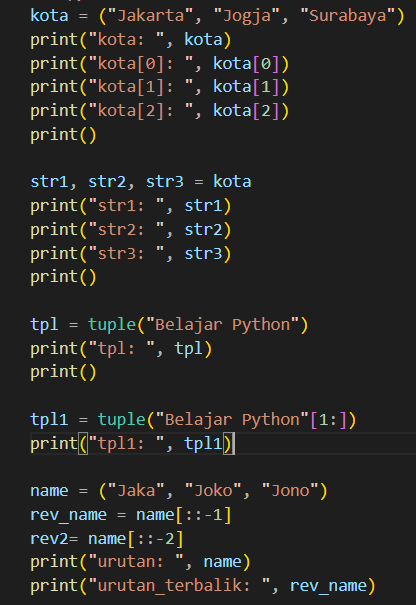
str3: Surabaya

tpl: ('B', 'e', 'l', 'a', 'j', 'a', 'r', ' ', 'P', 'y', 't', 'h', 'o', 'n')

tpl1: ('e', 'l', 'a', 'j', 'a', 'r', ' ', 'P', 'y', 't', 'h', 'o', 'n')

urutan: ('Jaka', 'Joko', 'Jono')

urutan\_terbalik: ('Jono', 'Joko', 'Jaka')

Pembahasan kasus 2.1

**Kasus 2.2** Buatlah program yang dapat menghitung jumlah commit yang dilakukan oleh akun

email tertentu. Gunakan file mbox-short.txt atau mbox.txt

Enter a file name: mbox-short.txt

cwen@iupui.edu 5

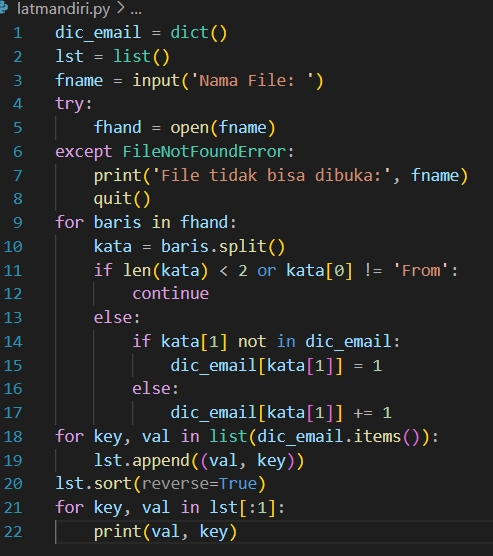
Enter a file name: mbox.txt

zqian@umich.edu 195

Untuk menyelesaikan kasus 11.2, langkah-langkah yang perlu diambil adalah sebagai berikut:

1. Membaca file teks.
2. Parsing baris yang dimulai dengan "From" untuk menghitung jumlah pesan berdasarkan email pengirim.
3. Membuat daftar tuple (email, jumlah pesan).
4. Mengurutkan daftar tersebut untuk mencetak email yang paling banyak melakukan commit.

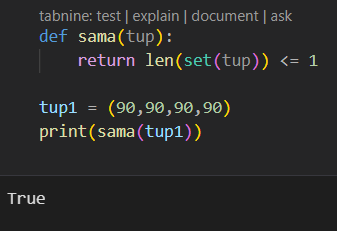
Berikut adalah kode program yang diringkas untuk mencapai tujuan tersebut:



# BAGIAN 2: LATIHAN MANDIRI (60%)

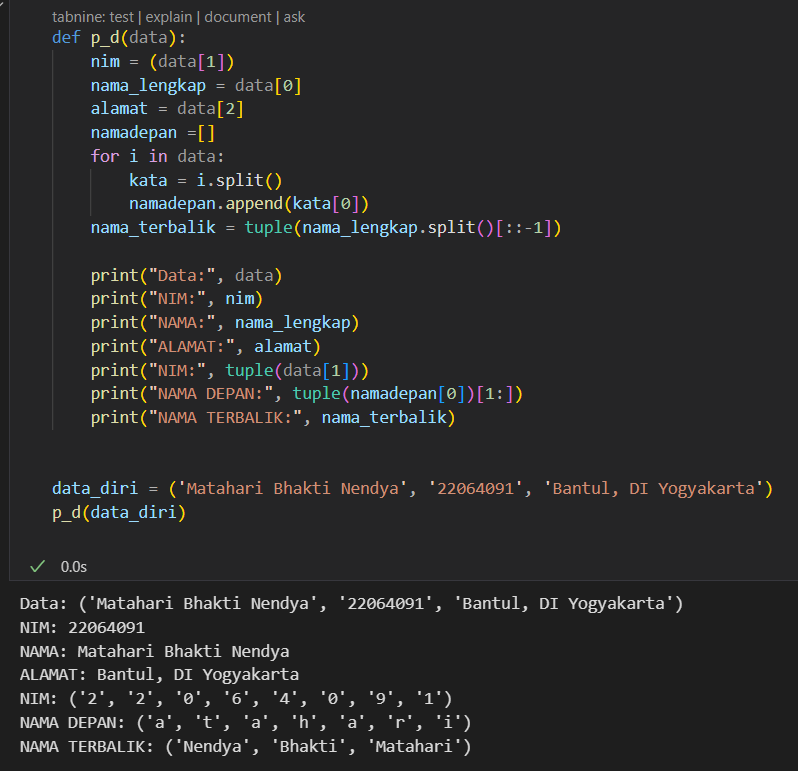
Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

## SOAL 1



Fungsi sama(tup) memeriksa apakah semua elemen dalam tuple tup adalah sama. set(tup) mengubah tup menjadi set yang hanya berisi elemen unik. len(set(tup)) menghitung jumlah elemen unik dalam set tersebut. return len(set(tup)) <= 1 mengembalikan True jika jumlah elemen unik dalam tup kurang dari atau sama dengan 1, artinya semua elemen sama atau tup kosong; jika tidak, mengembalikan False.

## SOAL 2



Fungsi `p\_d(data)` pada kode tersebut melakukan beberapa hal sebagai berikut:

1. Mengambil nilai dari elemen kedua dalam tuple `data` sebagai `nim`.

2. Mengambil nilai dari elemen pertama dalam tuple `data` sebagai `nama\_lengkap`.

3. Mengambil nilai dari elemen ketiga dalam tuple `data` sebagai `alamat`.

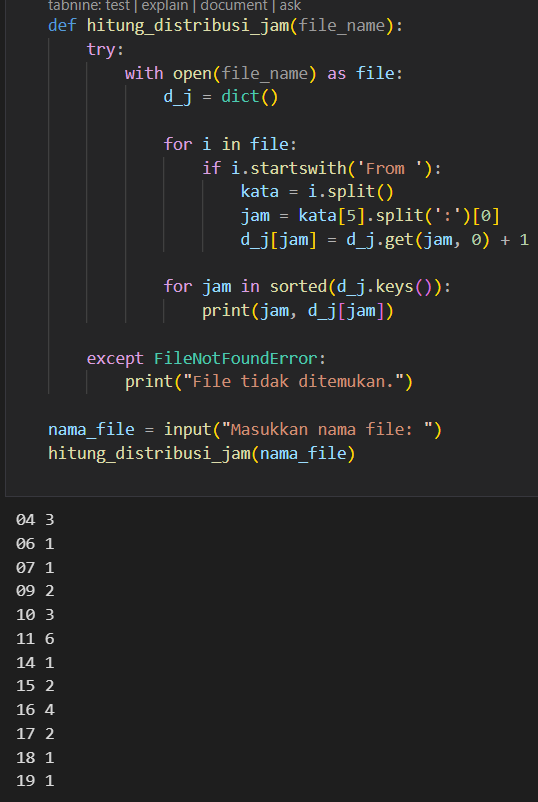
4. Membuat daftar kosong `namadepan`.

5. Melakukan iterasi untuk setiap elemen dalam `data`, memisahkannya berdasarkan spasi, dan menambahkan huruf pertama dari setiap kata ke dalam daftar `namadepan`.

6. Membalik urutan kata dalam `nama\_lengkap` dan menyimpannya dalam `nama\_terbalik`.

7. Mencetak informasi yang telah diproses, termasuk `nim`, `nama\_lengkap`, `alamat`, `nim` sebagai tuple, huruf pertama dari setiap kata dalam `nama\_lengkap` kecuali yang pertama, dan `nama\_lengkap` dengan urutan kata dibalik.

## SOAL 3



Fungsi hitung\_distribusi\_jam(file\_name) pada kode tersebut melakukan analisis distribusi jam dari baris yang dimulai dengan 'From ' dalam sebuah file. Berikut adalah penjelasan ringkas mengenai langkah-langkah yang dilakukan dalam fungsi tersebut:

1. Membuka File: Mencoba membuka file dengan nama yang diberikan (file\_name). Jika file tidak ditemukan, akan menampilkan pesan "File tidak ditemukan."
2. Inisialisasi Kamus: Membuat kamus d\_j untuk menyimpan jumlah kemunculan setiap jam.
3. Membaca File Baris per Baris:
   * Memeriksa Baris: Setiap baris dalam file diperiksa apakah diawali dengan 'From '.
   * Ekstraksi Jam: Jika baris dimulai dengan 'From ', baris tersebut dipisah menjadi kata-kata, kemudian mengambil bagian jam dari waktu (kata kelima, lalu bagian pertama sebelum ':' dari waktu tersebut).
   * Menghitung Frekuensi: Menambah jumlah kemunculan jam tersebut dalam kamus d\_j.
4. Menampilkan Distribusi Jam:
   * Mengurutkan Jam: Mengurutkan kunci-kunci dalam kamus d\_j.
   * Mencetak Hasil: Mencetak setiap jam dan jumlah kemunculannya.
5. Input Pengguna: Meminta pengguna untuk memasukkan nama file dan memanggil fungsi hitung\_distribusi\_jam dengan nama file tersebut.

Kode ini berguna untuk menganalisis frekuensi kemunculan jam dalam file log email, di mana setiap baris yang diawali dengan 'From ' menyimpan informasi waktu pengiriman email.