

Dasar-Dasar Pemrograman 1

Lab 08 Recursion



FAKULTAS
ILMU
KOMPUTER

The Disappearance of Dek Depe



Detektif Depram merupakan seorang detektif yang magang di kota Ku-Tech. Pada suatu hari, Depram mendapati kabar dari kepala detektif di kantor bahwa teman lamanya, Dek Depe telah menghilang! Detektif Depram berhasil mengumpulkan banyak nama orang yang mungkin memiliki informasi mengenai Dek Depe. Karena terlalu banyak, Detektif Depram kesulitan untuk memeriksanya sendiri. Oleh karena itu, bantulah Depram untuk membuat program yang dapat mengetahui hubungan antar orang!

Ketentuan Program

- Program akan meminta data berupa nama, nama kenalannya, dan jarak antara dua orang tersebut yang dipisahkan oleh spasi. Program akan

meminta input berulang kali hingga Anda memberikan input “SELESAI”.
Format input dijamin valid.

`<nama_x> <kenalan_x> <jarak>`

- Setelah itu, program meminta nama pertama `<nama_x>` dan nama tujuan `<nama_y>`. Program akan menghitung berapa jarak antara kedua orang tersebut, lalu **dikalikan dengan 10**.
 - **Input dijamin valid.**
- Output program berupa **jarak total** dan sebuah **pesan** berdasarkan jarak total yang didapatkan.
 - Jika jarak total > 1000 , maka cetak:
 - “`<nama_x>` dan `<nama_y>` tidak saling kenal.”
 - Jika $100 < \text{jarak total} \leq 1000$ program akan mencetak
 - “`<nama_x>` dan `<nama_y>` mungkin saling kenal.”
 - Jika $0 < \text{jarak total} \leq 100$ program akan mencetak
 - “`<nama_x>` dan `<nama_y>` kenal dekat.”
- Apabila **hubungan tidak ditemukan**, maka cetak
 - “Tidak ada hubungan antara `<nama_x>` dan `<nama_y>`.”

Keterangan:

- `<nama_x>` dan `<kenalan_x>` bersifat *case sensitive*.
- `<nama_x>` dan `<kenalan_x>` **dijamin hanya terdiri dari satu kata**.
- Pasangan `<nama_x>` dan `<kenalan_x>` pasti **unik**.
- **Setiap orang hanya kenal satu orang lainnya dan memiliki satu arah.**
Yaitu dari orang pertama ke selanjutnya / kenalannya.
Contoh: Input “Ren Dee 10” artinya Ren mengenal Dee, namun Dee tidak mengenal Ren. Ren juga hanya memiliki satu kenalan, yaitu Dee, sehingga Ren tidak akan muncul sebagai nama pertama lagi.
- `<jarak>` berupa **bilangan desimal positif**.
- **Input dijamin valid.**



Pengerjaan lab **WAJIB menggunakan rekursi**. Submisi yang tidak menggunakan rekursi tidak akan dinilai.

Test Case 1

Masukkan data hubungan:

A B 10.0

B C 6.4

C D 4.4

D E 5.6

SELESAI

Masukkan nama awal: A

Masukkan nama tujuan: E

Jarak total: 264

A dan E mungkin saling kenal.

Penjelasan

Visualisasi dari data hubungan adalah sebagai berikut

V-----V

A -> B -> C -> D -> E

Untuk mencapai dari A ke E, maka melewati 4 hubungan (A-B-C-D-E) yang berjarak $10.0 + 6.4 + 4.4 + 5.6 = 26.4$, sehingga didapatkan jarak total $26.4 * 10 = 264$, dan pesan "A dan E mungkin saling kenal."

Test Case 2

Masukkan data hubungan:

A B 7.6

B C 9.7

C D 5.2

D E 1.2

E F 1.2

F G 1.8

G H 6.6

SELESAI

Masukkan nama awal: D

Masukkan nama tujuan: G

Jarak total: 42

D dan G kenal dekat.



Penjelasan

Visualisasi dari data hubungan adalah sebagai berikut

```
          V-----V
A -> B -> C -> D -> E -> F -> G -> H
```

Untuk mencapai dari D ke G, maka melewati 3 hubungan (D-E-F-G), yang berjarak adalah $1.2 + 1.2 + 1.8 = 4.2$, sehingga didapatkan jarak total $4.2 * 10 = 42$, dan pesan "D dan G kenalan dekat."

Test Case 3

Masukkan data hubungan:

Roger Sumatera 1.2

Asif Mudasir 0.6

Sigit Rendang 1.1

Rendang Asif 2.4

Mudasir Roger 1.2

SELESAI

Masukkan nama awal: Mudasir

Masukkan nama tujuan: Asif

Tidak ada hubungan antara Mudasir dan Asif.

Penjelasan

Visualisasi dari data hubungan adalah sebagai berikut

```
Sigit -> Rendang -> Asif -> Mudasir -> Roger -> Sumatera
```

Karena Mudasir berada setelah Asif, maka hubungan Mudasir dengan Asif tidak ditemukan.

Komponen Penilaian

- **30%** Aspek komputasi
- **45%** Penggunaan recursion
- **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

Deliverables

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

[Kelas]_[KodeAsdos]_[NPM]_[NamaLengkap]_Lab08.zip

Contoh:

A_ABC_2206123456_ThamiEndamora_Lab08.zip