

Lab 07

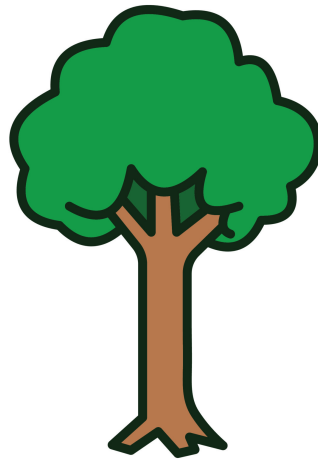
Recursion



FAKULTAS
ILMU
KOMPUTER

Relation Finder

Sub-CPMK 7 : Mampu menjabarkan dan merancang program rekursi



source:

<https://www.vecteezy.com/vector-art/24444481-colored-tree-vector-icon-isolated-on-square-white-background-simple-flat-outline-d-cartoon-icon-drawing-with-nature-botanical-theme>

Setelah sukses dengan program *plagiarism checker*-nya, Pak Esde, selaku ketua RT di tempat tinggalnya, meminta tolong kepada Dek Depe untuk membuat program sederhana yang dapat membantu Pak Esde mencatat silsilah keluarga seseorang. Melihat Dek Depe yang mulai kewalahan dengan permintaan Pak Esde, kamu sebagai teman yang baik pun berniat menolongnya.

Ketentuan Program

Di awal, program akan meminta data relasi. Format masukannya adalah:

`<nama_parent> <nama_child>`

Program akan terus meminta masukan sampai Anda memberi masukan "SELESAI".
Dipastikan format masukan sesuai.

- **Keterangan dan Batasan**

- `<nama_parent>` dan `<nama_child>` bersifat *case sensitive*.
- `<nama_parent>` dan `<nama_child>` **dijamin hanya terdiri dari satu kata.**
- Setiap child hanya memiliki **maksimal satu parent** tetapi satu parent **bisa memiliki banyak child.**
- Untuk setiap baris input, `<nama_parent>` dan `<nama_child>` berjarak 1 dari satu sama lain.
- Jarak generasi antara suatu parent dan suatu child **dijamin tidak lebih dari 100.**
- Dalam data relasi, **dijamin suatu child tidak akan memiliki relasi kepada parent maupun pendahulunya.**

- **Daftar Perintah**

- `CEK_KETURUNAN <nama_parent> <nama_child>`

Perintah ini diwakili dengan angka 1. Merupakan perintah yang akan mengecek apakah `<nama_child>` merupakan keturunan dari `<nama_parent>`, baik secara langsung maupun tidak.

- Jika `<nama_child>` merupakan keturunan dari `<nama_parent>` maka cetak:
 - `<nama_child>` benar merupakan keturunan dari `<nama_parent>`
- Jika `<nama_child>` **bukan** merupakan keturunan dari `<nama_parent>` maka cetak:
 - `<nama_child>` bukan merupakan keturunan dari `<nama_parent>`

- **CETAK_KETURUNAN** <nama_parent>

Perintah ini diwakili dengan angka 2. Merupakan perintah yang akan mencetak seluruh child dari <nama_parent>, baik yang merupakan child secara langsung maupun tidak. **Hasil pencetakan tidak harus terurut.**

- **JARAK_GENERASI** <nama_parent> <nama_child>

Perintah ini diwakili dengan angka 3. Merupakan perintah yang akan menghitung jarak antara <nama_parent> dan <nama_child>. Setiap relasi <nama_parent> dan <nama_child> berjarak 1. Untuk lebih jelasnya, silakan lihat pada Contoh Test Cases.

- **EXIT**

Perintah ini diwakili dengan angka 4. Program akan berhenti dan mencetak pesan “**Terima kasih telah menggunakan Relation Finder!**”

- **Mahasiswa WAJIB menggunakan solusi rekursif untuk lab 7 ini.**

Contoh Test Cases:

Input hanya untuk memudahkan pengujian program, hasil input dan output disesuaikan dengan Output.

Input 1

```
Asep Dwi
Agus Budi
Budi Bambang
Agus Asep
Budi Fatalis
SELESAI
2
Agus
1
Bambang
Agus
1
Agus
Bambang
3
Budi
Asep
3
Agus
Bambang
4
```

Output 1

Masukkan data relasi:

```
Asep Dwi
Agus Budi
Budi Bambang
Agus Asep
Budi Fatalis
SELESAI
```

=====

Selamat Datang di Relation Finder! Pilihan yang tersedia:

1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI

4. EXIT

Masukkan pilihan: 2

Masukkan nama parent: Agus

- Budi Asep
- Bambang Fatalis
- Dwi

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:

1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT

Masukkan pilihan: 1

Masukkan nama parent: Bambang

Masukkan nama child: Agus

Agus bukan merupakan keturunan dari Bambang

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:

1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT

Masukkan pilihan: 1

Masukkan nama parent: Agus

Masukkan nama child: Bambang

Bambang benar merupakan keturunan dari Agus

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:

1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT

Masukkan pilihan: 3

Masukkan nama parent: Budi

Masukkan nama child: Asep

Tidak ada hubungan antara Budi dengan Asep

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:

1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI

```
4. EXIT
Masukkan pilihan: 3
Masukkan nama parent: Agus
Masukkan nama child: Bambang
Agus memiliki hubungan dengan Bambang sejauh 2

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:
1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT
Masukkan pilihan: 4
=====
Terima kasih telah menggunakan Relation Finder!
```

Input 2

```
Susan Putri
SELESAI
1
Putri
Agus
2
Putri
3
Putri
Putri
4
```

Output 2

```
Masukkan data relasi:
Susan Putri
SELESAI

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:
1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT
Masukkan pilihan: 1
```

```

Masukkan nama parent: Putri
Masukkan nama child: Agus
Agus bukan merupakan keturunan dari Putri

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:
1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT
Masukkan pilihan: 2
Masukkan nama parent: Putri

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:
1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT
Masukkan pilihan: 3
Masukkan nama parent: Putri
Masukkan nama child: Putri
Putri memiliki hubungan dengan Putri sejauh 0

=====
Selamat Datang di Relation Finder! Pilihan yang tersedia:
1. CEK_KETURUNAN
2. CETAK_KETURUNAN
3. JARAK_GENERASI
4. EXIT
Masukkan pilihan: 4

=====
Terima kasih telah menggunakan Relation Finder!

```

Keterangan Test Case 2

- Putri tidak memiliki keturunan sehingga tidak memunculkan Output apapun
- Jarak Generasi Putri terhadap dirinya sendiri adalah nol.

Komponen Penilaian

- **40%** Kebenaran fungsionalitas program
- **35%** Penggunaan fungsi rekursif
- **10%** Memenuhi kriteria standar penulisan kode Python*

- **10%** Mengumpulkan dengan format dan penamaan file yang benar
- **5%** Dokumentasi kode

*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

Berkas yang Perlu Dikumpulkan

- lab07.py

Kumpulkan berkas lab07.py dalam sebuah file .zip dengan format penamaan seperti berikut:

`[KodeAsdos]_[Kelas]_[NPM]_[NamaLengkap]_Lab07.zip`

Contoh:

`IAN_A_2306969420_MikasaAckerman_Lab07.zip`