

LAPORAN

RENCANA TUGAS MANDIRI (RTM) Ke-4

MATA KULIAH ALGORITMA DAN PEMROGRAMAN LANJUT

**“Membuat Kode Script Python untuk Memvalidasi Inputan
User Menggunakan Setter dan Getter, serta Exception
Handling”**



DISUSUN OLEH:

Muhammad Aryasatya Nugroho (22083010085)

DOSEN PENGAMPU:

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

PROGRAM STUDI SAINS DATA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR

2023

Kode Script:

```
1 class InvalidAge(Exception):
2     pass
3 class InvalidGender(Exception):
4     pass
5 class InvalidEmail(Exception):
6     pass
7
8 class person:
9     def __init__(self, n, a, g, e):
10        self.name = n
11        self.age = a
12        self.gender = g
13        self.email = e
14
15    @property
16    def name(self):
17        return self._name
18
19    @name.setter
20    def name(self, val):
21        self._name = val
22
23    @property
24    def age(self):
25        return self._age
26
27    @age.setter
28    def age(self, val):
29        if not 0 <= val <= 120:
30            raise InvalidAge
31        self._age = val
32
33    @property
34    def gender(self):
35        return self._gender
36
37    @gender.setter
38    def gender(self, val):
39        if val not in ['P', 'L']:
40            raise InvalidGender
41        self._gender = val
42
43    @property
44    def email(self):
45        return self._email
46
47    @email.setter
48    def email(self, val):
49        if not val.endswith('@gmail.com'):
50            raise InvalidEmail
51        self._email = val
```

Kode Script diatas mendefinisikan sebuah kelas ‘person’ yang memiliki empat atribut yaitu ‘name’, ‘age’, ‘gender’, dan ‘email’. Setiap atribut memiliki method getter dan setter yang memungkinkan nilai atribut untuk diambil atau dimodifikasi. Pada setiap setter, dilakukan pengecekan validitas nilai yang dimasukkan menggunakan raise exception.

Ada tiga jenis exception yang didefinisikan dalam kelas ini, yaitu ‘InvalidAge’, ‘InvalidGender’, dan ‘InvalidEmail’. Ketika ada kesalahan dalam pengisian nilai atribut, maka salah satu dari tiga exception tersebut akan di-raise.

Pengecekan nilai atribut ‘age’ dilakukan dengan memastikan nilai umur berada dalam rentang 0 hingga 120 tahun, atribut ‘gender’ hanya menerima nilai (P/L), sedangkan pengecekan atribut email dilakukan dengan memastikan bahwa email yang dimasukkan memiliki format domain ‘@gmail.com’.

Seluruh atribut pada kelas ‘person’ dilengkapi dengan decorator ‘@property’ sehingga dapat diakses dan dimodifikasi melalui method getter dan setter.

Contoh Penggunaan Class:

```
1 try:
2     Ind1 = person('arya', 13, 'L', 'email@outlook.com')
3 except InvalidAge as e:
4     raise Exception("Invalid age")
5 except InvalidGender as e:
6     raise Exception("Invalid gender")
7 except InvalidEmail as e:
8     raise Exception("Invalid email")
9 else:
10    print("Informasi benar")
```

Kode Script diatas merupakan contoh penggunaan dari kelas 'person'. Tujuan dari kode script diatas membuat objek 'Ind1' dengan memasukkan parameter 'arya', 13, 'L', dan 'email@outlook.com' yang diinisialisasi pada kelas person.

Namun, karena kelas tersebut memiliki beberapa validasi untuk properti objek, maka kemungkinan terdapat exception ketika objek dibuat. Oleh karena itu, kode tersebut menggunakan blok 'try' dan 'except' untuk menangkap exception yang mungkin muncul.

Jika terdapat exception yang muncul, kode menampilkan pesan yang sesuai dengan jenis exception. Jika tidak ada exception yang muncul, maka kode akan menampilkan pesan "berhasil!".

Dalam contoh kode tersebut, parameter 13 untuk properti age akan lolos validasi, sedangkan email@outlook.com untuk properti email akan memicu exception InvalidEmail. Oleh karena itu, blok except untuk InvalidEmail akan dieksekusi, dan program akan menampilkan pesan "Invalid email".

```
InvalidEmail                                Traceback (most recent call 1
ast)
Input In [162], in <cell line: 1>()
1 try:
----> 2     Ind1 = person('arya', 13, 'L', 'email@outlook.com')
3 except InvalidAge as e:
```

```
Input In [126], in person.__init__(self, n, a, g, e)
12 self.gender = g
---> 13 self.email = e
```

```
Input In [126], in person.email(self, val)
49 if not val.endswith('@gmail.com'):
---> 50     raise InvalidEmail
51 self._email = val
```

InvalidEmail:

During handling of the above exception, another exception occurred:

```
Exception                                Traceback (most recent call 1
ast)
Input In [162], in <cell line: 1>()
6     raise Exception("Invalid gender")
7 except InvalidEmail as e:
----> 8     raise Exception("Invalid email")
9 else:
10    print("Informasi benar")
```

Exception: Invalid email

Jika Semua Objek Sesuai

```
1 try:
2     ind1 = person('arya', 13, 'L', 'helloworld@gmail.com')
3 except InvalidAge as exc:
4     message = "InvalidAgeException : umur hanya 0 - 120 tahun"
5 except InvalidGender:
6     message = "InvalidGenderException : gender hanya 'P' atau 'L'"
7 except InvalidEmail:
8     message = "InvalidEmailException : domain email bukan '@gmail.com'"
9 else:
10    message = "Informasi benar"
11
12 print(message)
```

Informasi benar

Kode script diatas menunjukkan penggunaan metode memberikan pesan pada saat terjadi exception dalam parameter objek dari class 'person'. Pada kode script diatas tidak terjadi exception dari semua parameter name, age, gender, dan email sehingga exception tidak dicetak dan else akan dieksekusi dengan mencetak pesan "Informasi benar"

Jika Objek Umur Tidak Sesuai

```
1 try:
2     ind1 = person('arya', 130, 'L', 'helloworld@gmail.com')
3 except InvalidAge as exc:
4     message = "InvalidAgeException : umur hanya 0 - 120 tahun"
5 except InvalidGender:
6     message = "InvalidGenderException : gender hanya 'P' atau 'L'"
7 except InvalidEmail:
8     message = "InvalidEmailException : domain email bukan '@gmail.com'"
9 else:
10    message = "Informasi benar!"
11
12 print(message)
```

InvalidAgeException : umur hanya 0 - 120 tahun

Kode di atas digunakan untuk membuat objek person dengan nama 'arya', usia 130 tahun, gender 'L', dan email 'helloworld@gmail.com'. Namun, karena umur yang dimasukkan di luar rentang yang valid yaitu 0 sampai 120, maka akan memunculkan InvalidAgeException. Exception tersebut kemudian ditangani dengan memberikan pesan kesalahan yang menjelaskan penyebab exception tersebut. Pesan kesalahan kemudian dicetak menggunakan fungsi print().

Jika Objek Gender Tidak Sesuai

```
1 try:
2     ind1 = person('arya', 13, 'A', 'helloworld@gmail.com')
3 except InvalidAge as exc:
4     message = "InvalidAgeException : umur hanya 0 - 120 tahun"
5 except InvalidGender:
6     message = "InvalidGenderException : gender hanya 'P' atau 'L'"
7 except InvalidEmail:
8     message = "InvalidEmailException : domain email bukan '@gmail.com'"
9 else:
10    message = "Informasi benar!"
11
12 print(message)
```

InvalidGenderException : gender hanya 'P' atau 'L'

Kode di atas digunakan untuk membuat objek person dengan nama 'arya', usia 13 tahun, gender 'A', dan email 'helloworld@gmail.com'. Namun, karena gender yang dimasukkan selain 'P' atau 'L', maka akan memunculkan InvalidGenderException. Exception tersebut kemudian ditangani dengan memberikan pesan kesalahan yang menjelaskan penyebab exception tersebut. Pesan kesalahan kemudian dicetak menggunakan fungsi print().

Jika Objek Email Tidak Sesuai

```
1 try:
2     ind1 = person('arya', 13, 'L', 'helloworld@outlook.com')
3 except InvalidAge as exc:
4     message = "InvalidAgeException : umur hanya 0 - 120 tahun"
5 except InvalidGender:
6     message = "InvalidGenderException : gender hanya 'P' / 'L'"
7 except InvalidEmail:
8     message = "InvalidEmailException : domain email bukan '@gmail.com'"
9 else:
10    message = "Informasi benar!"
11
12 print(message)
```

InvalidEmailException : domain email bukan '@gmail.com'

Kode di atas digunakan untuk membuat objek person dengan nama 'arya', usia 13 tahun, gender 'L', dan email 'helloworld@outlook.com'. Namun, karena domain email yang dimasukkan bukan domain '@gmail.com', maka akan memunculkan InvalidEmailException. Exception tersebut kemudian ditangani dengan memberikan pesan kesalahan yang menjelaskan penyebab exception tersebut. Pesan kesalahan kemudian dicetak menggunakan fungsi print().

Kesimpulan:

Kode tersebut merupakan contoh penggunaan class dan exception. Class tersebut bernama person dan memiliki beberapa atribut seperti nama, umur, gender, dan email. Setiap atribut memiliki validasi tertentu, jika ada input yang tidak valid, maka akan memunculkan exception yang telah diatur sebelumnya, seperti InvalidAge, InvalidGender, dan InvalidEmail.

Selain itu, dalam kode tersebut juga terdapat penggunaan try-except untuk menangani exception dan memberikan pesan kesalahan yang sesuai. Jika semua input valid, maka kode akan menghasilkan pesan "Informasi benar!".

Penggunaan exception dan validasi seperti ini sangat penting untuk memastikan data yang dimasukkan ke dalam program sudah benar dan valid. Dengan begitu, program akan dapat berjalan dengan baik dan dapat menghasilkan output yang diinginkan.