

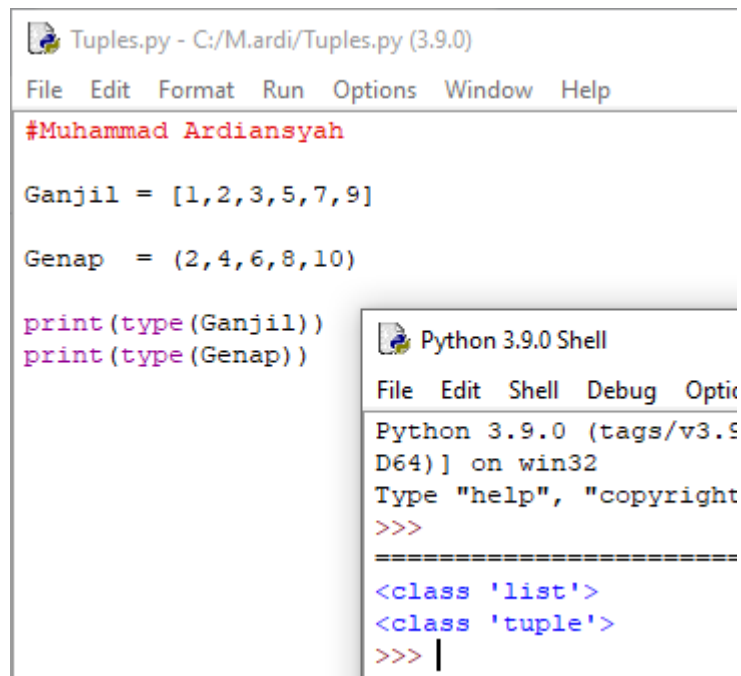
Muhammad Ardiansyah	Jobsheet 1 Tipe Data Tuple dan Sets	Sabtu, 13 Maret 2021
		SaaS
XII SIJA A		Antoni Budiman, S.Pd Maya Yashbir, ST.
SMKN 1 CIMAHI		Paraf:

Tipe Data Tuple dan Sets

Pembahasan

1. Tuple

Cara membuat tuple itu sama seperti cara membuat list, tetapi bedanya adalah jika list menggunakan kurung siku ([]), sedangkan tuple itu menggunakan kurung biasa (()). Bagaimana cara mengetahui tipe data tersebut list atau tuple? Hal ini dapat dilakukan dengan cara melakukan print pada kedua tipe data tersebut dengan perintah “**print(type(NamaVariabel))**” seperti pada contoh berikut.



The image shows a screenshot of a Python script editor and a terminal window. The script, named 'Tuples.py', defines a list 'Ganjil' and a tuple 'Genap'. It then prints the types of these variables. The terminal output shows that 'Ganjil' is of type 'list' and 'Genap' is of type 'tuple'.

```

Tuples.py - C:/M.ardi/Tuples.py (3.9.0)
File Edit Format Run Options Window Help

#Muhammad Ardiansyah

Ganjil = [1,2,3,5,7,9]

Genap = (2,4,6,8,10)

print(type(Ganjil))
print(type(Genap))

```

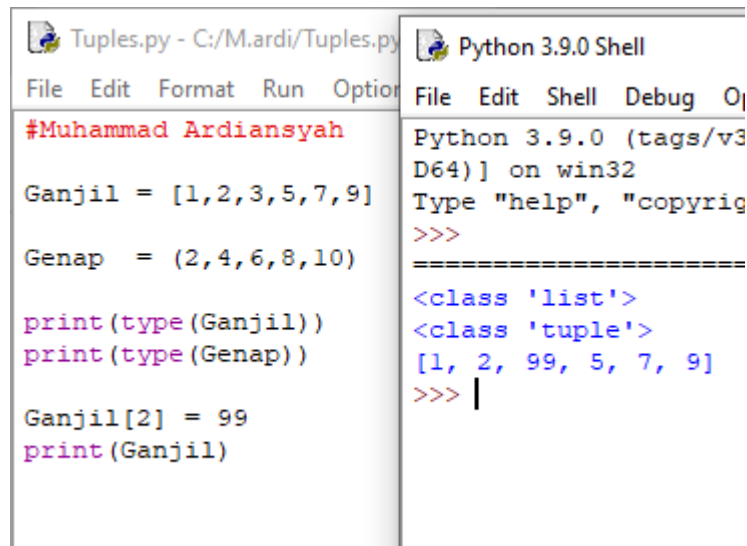
```

Python 3.9.0 Shell
File Edit Shell Debug Optic
Python 3.9.0 (tags/v3.9.0
D64) on win32
Type "help", "copyright
>>>
=====
<class 'list'>
<class 'tuple'>
>>> |

```

Tuple ini tidak bisa diubah nilainya dan tidak bisa di tambah nilainya (memberinya) karena tuple ini bersifat **fix** berbeda dengan list yang dapat diubah maupun ditambah

nilainya (memberinya). Contohnya ketika saya mau mengubah nilai pada list, maka nilai bisa diubah seperti pada gambar dibawah.



```
Tuples.py - C:/M.ardi/Tuples.py
File Edit Format Run Options
#Muhammad Ardiansyah

Ganjil = [1,2,3,5,7,9]

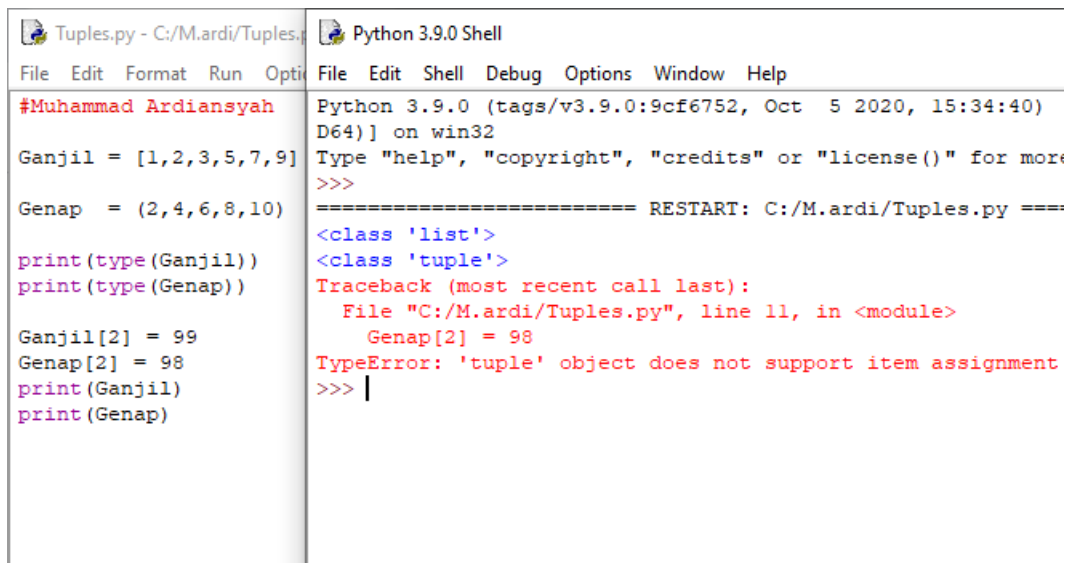
Genap = (2,4,6,8,10)

print(type(Ganjil))
print(type(Genap))

Ganjil[2] = 99
print(Ganjil)
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
=====
<class 'list'>
<class 'tuple'>
[1, 2, 99, 5, 7, 9]
>>> |
```

Dan ketika saya ingin mengubah nilai pada tuple, maka akan terjadi error karena tuple ini tidak support dengan item assignment sehingga kita tidak bisa mengubah nilai dari si tuple itu sendiri. Tampilan errornya adalah seperti berikut.



```
Tuples.py - C:/M.ardi/Tuples.py
File Edit Format Run Options
#Muhammad Ardiansyah

Ganjil = [1,2,3,5,7,9]

Genap = (2,4,6,8,10)

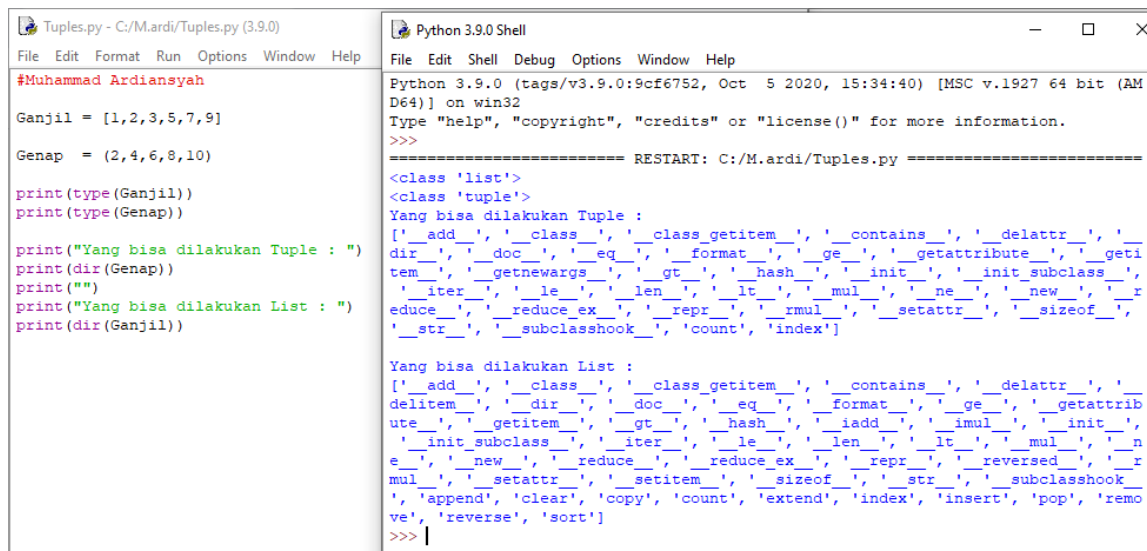
print(type(Ganjil))
print(type(Genap))

Ganjil[2] = 99
Genap[2] = 98
print(Ganjil)
print(Genap)
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
===== RESTART: C:/M.ardi/Tuples.py =====
<class 'list'>
<class 'tuple'>
Traceback (most recent call last):
  File "C:/M.ardi/Tuples.py", line 11, in <module>
    Genap[2] = 98
TypeError: 'tuple' object does not support item assignment
>>> |
```

Contoh lainnya mengenai tuple ini bersifat fix dan berbeda dengan list adalah ketika kita dapat menambahkan sebuah data pada akhir list dengan attribute 'append', namun pada tuple hal tersebut tidak dapat dilakukan melainkan akan muncul error .

Apa saja yang bisa digunakan dari tuple dan list ini? Untuk mengetahuinya, kita bisa melakukan print dengan menambahkan option dir, contoh perintahnya **'print(dir>NamaVariabel))'**.



```
Tuples.py - C:/M.ardi/Tuples.py (3.9.0)
File Edit Format Run Options Window Help
#Muhammad Ardiansyah

Ganjil = [1,2,3,5,7,9]

Genap = (2,4,6,8,10)

print(type(Ganjil))
print(type(Genap))

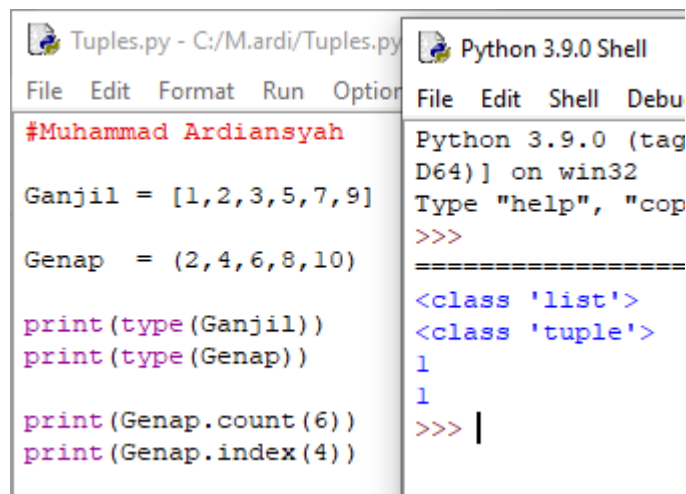
print("Yang bisa dilakukan Tuple : ")
print(dir(Genap))
print("")
print("Yang bisa dilakukan List : ")
print(dir(Ganjil))

Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/M.ardi/Tuples.py =====
<class 'list'>
<class 'tuple'>
Yang bisa dilakukan Tuple :
['_add_', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'count', 'index']
Yang bisa dilakukan List :
['_add_', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
>>>
```

Dapat dilihat dari gambar diatas, bahwa Tuple hanya memiliki ‘count’, dan ‘index’, berbeda dengan list memiliki cukup banyak attribute seperti ‘append’, ‘insert’, ‘reverse’, dll.

Apa fungsi ‘count’ dan ‘index’ pada tuple? Count difungsikan untuk menghitung jumlah sebuah data yang telah ditentukan pada sebuah tuple, contoh perintahnya adalah

“print>NamaVariabel.count(Data))”. Sedangkan Index difungsikan untuk mengetahui index dari sebuah data, contoh perintahnya adalah **“print>NamaVariabel.index(Data))”**.



```
Tuples.py - C:/M.ardi/Tuples.py
File Edit Format Run Option
#Muhammad Ardiansyah

Ganjil = [1,2,3,5,7,9]

Genap = (2,4,6,8,10)

print(type(Ganjil))
print(type(Genap))

print(Genap.count(6))
print(Genap.index(4))

Python 3.9.0 Shell
File Edit Shell Debu
Python 3.9.0 (tag
D64)] on win32
Type "help", "cop
>>>
=====
<class 'list'>
<class 'tuple'>
1
1
>>> |
```

Tuple ini berguna ketika kita memiliki sebuah data yang tidak bisa diubah, misalnya data sensus penduduk dan data KTP. Tuple ini lebih ringan daripada list untuk diproses.

<pre>Tuples.py - C:/M.ardi/Tuples.py (3.9.0) File Edit Format Run Options Window Help #Muhammad Ardiansyah import sys data_list = [1,2,3,4,5,"PPKN","Saas","Indo",False,3.14] data_tuple = (1,2,3,4,5,"PPKN","Saas","Indo",False,3.14) besar_datalist = sys.getsizeof(data_list) besar_datatuple = sys.getsizeof(data_tuple) print("Besar data list: ", besar_datalist) print("Besar data tuple: ", besar_datatuple)</pre>	<pre>Python 3.9.0 Shell File Edit Shell Debug Optio Python 3.9.0 (tags/v3.9 D64)] on win32 Type "help", "copyright >>> Besar data list: 152 Besar data tuple: 120 >>> </pre>
---	---

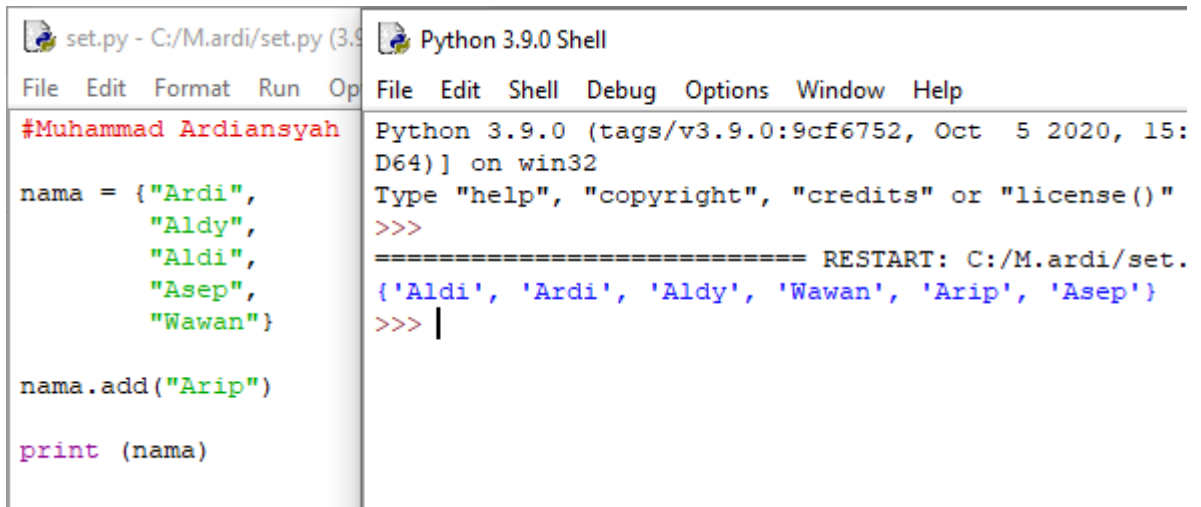
Dapat dilihat pada gambar diatas data list lebih besar dibandingkan data tuple, hal ini dikarenakan pada tuple itu tidak banyak fungsi yang dapat dipakai, berbeda dengan list yang memiliki banyak fungsi sehingga list memerlukan memori lebih banyak dan memiliki waktu proses yang lebih lama dibandingkan tuple. Dan berikut adalah buktinya.

<pre>Tuples.py - C:/M.ardi/Tuples.py (3.9.0) File Edit Format Run Options Window Help #Muhammad Ardiansyah import timeit data_list = [1,2,3,4,5,"PPKN","Saas","Indo",False,3.14] data_tuple = (1,2,3,4,5,"PPKN","Saas","Indo",False,3.14) waktu_list = timeit.timeit(stmt="(1,2,3,4,5,6,7,8,9)",number=1000000) waktu_tuple = timeit.timeit(stmt="(1,2,3,4,5,6,7,8,9)",number=1000000) print("Waktu untuk memproses list: ", waktu_list) print("Waktu untuk memproses tuple: ", waktu_tuple)</pre>	<pre>Python 3.9.0 Shell File Edit Shell Debug Options Window Help Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15: D64)] on win32 Type "help", "copyright", "credits" or "license()" >>> ===== RESTART: C:/M.ardi/Tuples Waktu untuk memproses list: 0.022983499999999999 Waktu untuk memproses tuple: 0.024252500000000001 >>> </pre>
---	--

Pada gambar diatas membuktikan bahwa untuk memproses tuple diperlukan waktu hampir 10x lebih cepat dibandingkan dengan list.

2. Set

Apa itu set? Set dalam bahasa indonesia itu sama dengan himpunan yang memiliki karakteristik tidak mempunyai urutan, frekuensi datanya ketika ada data yang sama hanya akan dihitung satu. Cara membuat tipe data set/himpunan ini ada dua, yang pertama menggunakan kurung kurawa ({ }). Contoh perintahnya “**NamaSet = {“data”, “yang”, “dimasukkan”}**”



The screenshot shows a Python IDE with two windows. The left window, titled 'set.py - C:/M.ardi/set.py (3.9.0)', contains the following code:

```
#Muhammad Ardiansyah

nama = {"Ardi",
        "Aldy",
        "Aldi",
        "Asep",
        "Wawan"}

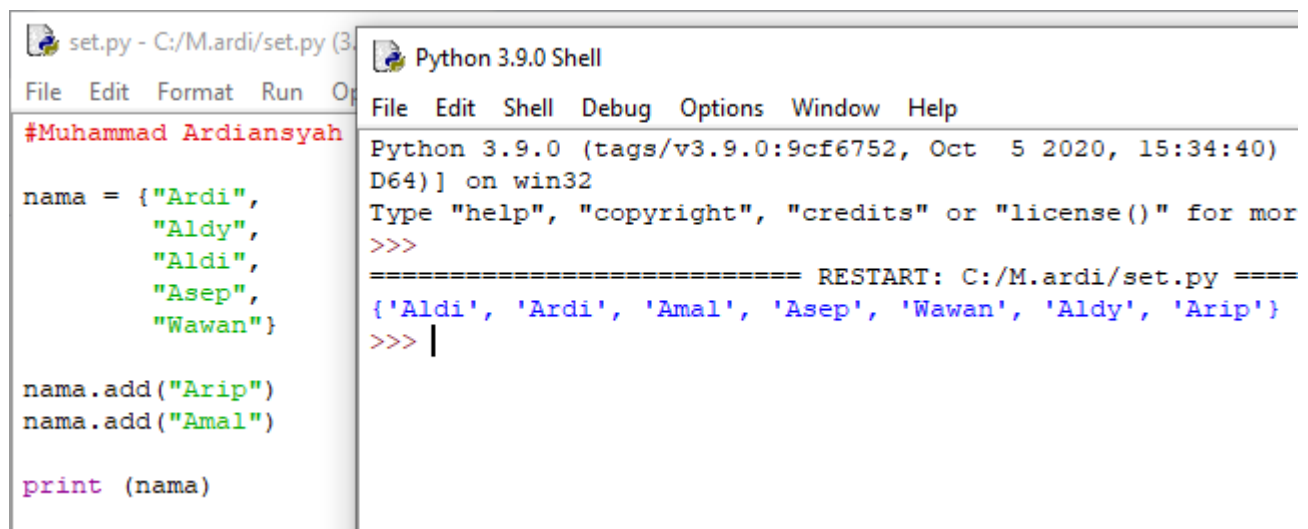
nama.add("Arip")

print (nama)
```

The right window, titled 'Python 3.9.0 Shell', shows the execution output:

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:
D64)] on win32
Type "help", "copyright", "credits" or "license()"
>>>
===== RESTART: C:/M.ardi/set.
{'Aldi', 'Ardi', 'Aldy', 'Wawan', 'Arip', 'Asep'}
>>> |
```

Dapat dilihat pada gambar diatas yang membuktikan bahwa sebuah himpunan itu memiliki karakteristik yang tidak mempunyai/tidak memperdulikan urutan. Dan ketika kita menambahkan data yang sama ke dalam sebuah set, maka data tersebut juga hanya akan ditampilkan satu kali seperti pada gambar berikut.



The screenshot shows a Python IDE with two windows. The left window, titled 'set.py - C:/M.ardi/set.py (3.9.0)', contains the following code:

```
#Muhammad Ardiansyah

nama = {"Ardi",
        "Aldy",
        "Aldi",
        "Asep",
        "Wawan"}

nama.add("Arip")
nama.add("Amal")

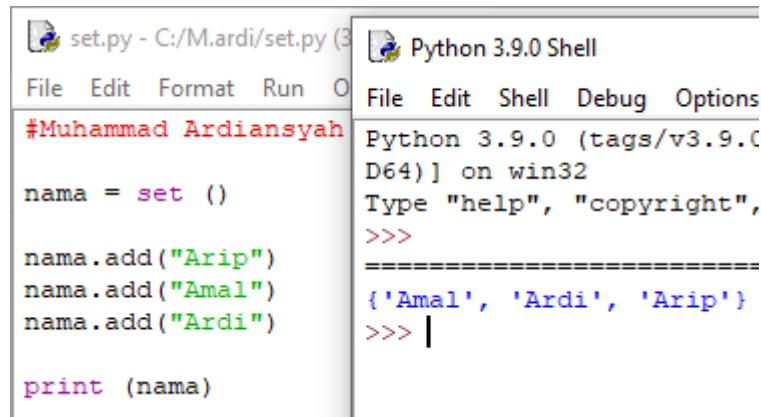
print (nama)
```

The right window, titled 'Python 3.9.0 Shell', shows the execution output:

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40)
D64)] on win32
Type "help", "copyright", "credits" or "license()" for mor
>>>
===== RESTART: C:/M.ardi/set.py =====
{'Aldi', 'Ardi', 'Amal', 'Asep', 'Wawan', 'Aldy', 'Arip'}
>>> |
```

Cara kedua untuk membuat set/himpunan adalah dengan cara menyatakan variable sebagai set terlebih dahulu, dengan contoh perintah “**NamaSet = set()**”. Dan diikuti dengan perintah ‘add’ untuk menambahkan data kedalam set dengan perintah

“**NamaSet.add(“Data yang ditambahkan”)**” seperti pada gambar berikut



```
set.py - C:/M.ardi/set.py (3)
File Edit Format Run O
#Muhammad Ardiansyah

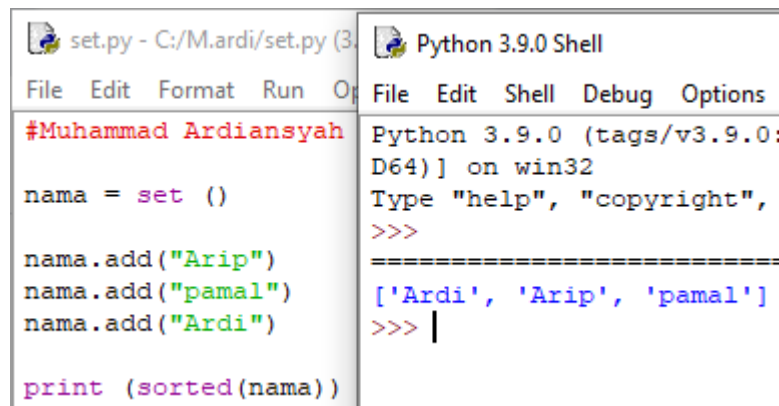
nama = set ()

nama.add("Arip")
nama.add("Amal")
nama.add("Ardi")

print (nama)
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options
Python 3.9.0 (tags/v3.9.0:
D64)] on win32
Type "help", "copyright",
>>>
=====
{'Amal', 'Ardi', 'Arip'}
>>> |
```

Kita juga sebenarnya bisa menampilkan set secaraurut berdasarkan abjad dengan perintah “**print(sorted>NamaSet))**”



```
set.py - C:/M.ardi/set.py (3)
File Edit Format Run O
#Muhammad Ardiansyah

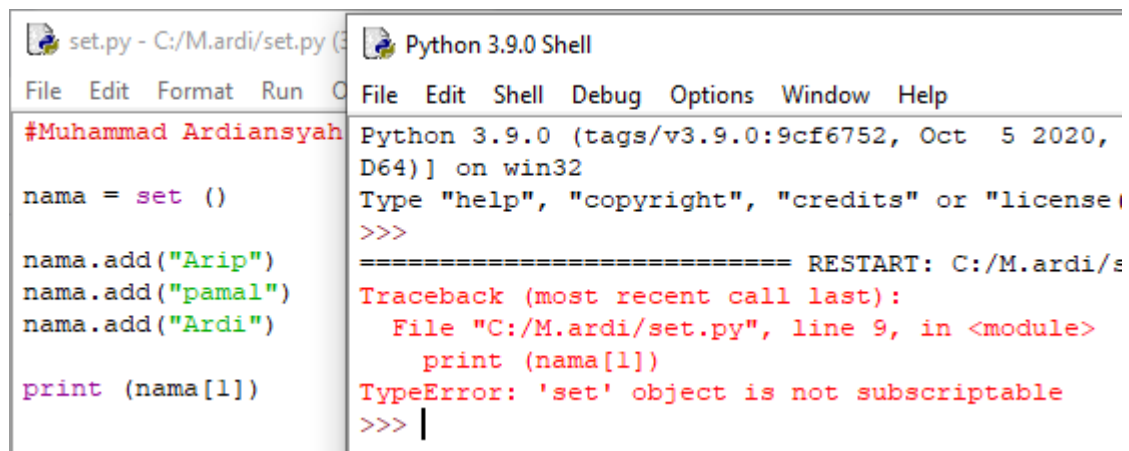
nama = set ()

nama.add("Arip")
nama.add("pamal")
nama.add("Ardi")

print (sorted(nama))
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options
Python 3.9.0 (tags/v3.9.0:
D64)] on win32
Type "help", "copyright",
>>>
=====
['Ardi', 'Arip', 'pamal']
>>> |
```

Selain itu, karena set ini tidak memperdulikan urutan, maka set tidak mengenal yang namanya index seperti pada gambar berikut.



```
set.py - C:/M.ardi/set.py (3)
File Edit Format Run O
#Muhammad Ardiansyah

nama = set ()

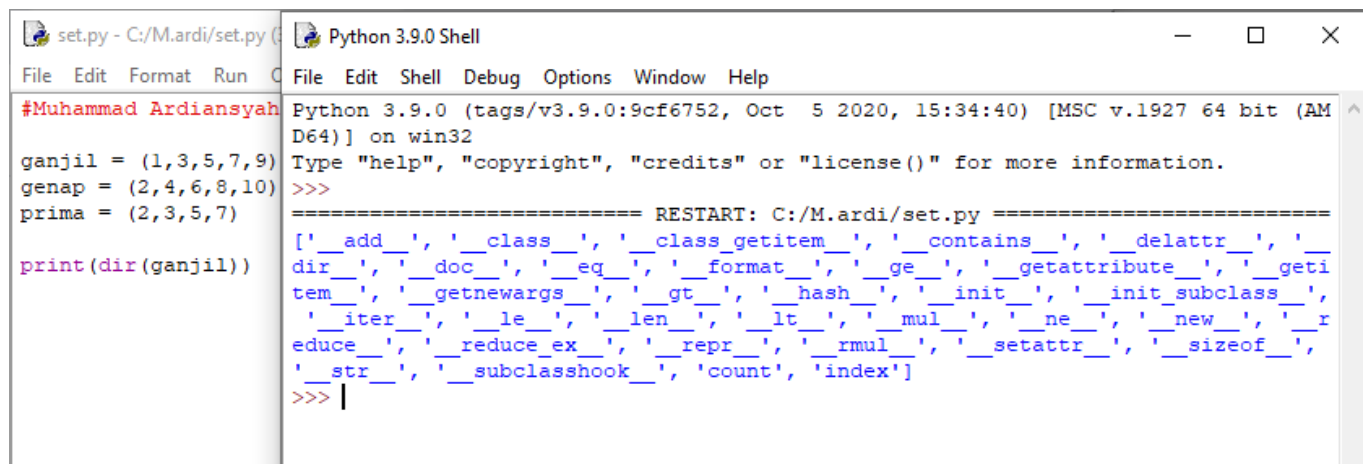
nama.add("Arip")
nama.add("pamal")
nama.add("Ardi")

print (nama[1])
```

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020,
D64)] on win32
Type "help", "copyright", "credits" or "license()
>>>
===== RESTART: C:/M.ardi/s
Traceback (most recent call last):
  File "C:/M.ardi/set.py", line 9, in <module>
    print (nama[1])
TypeError: 'set' object is not subscriptable
>>> |
```

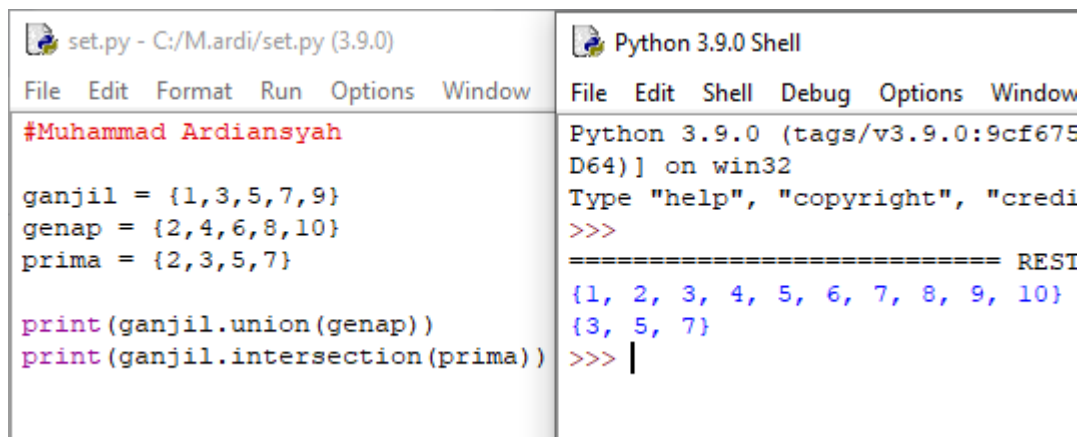
Beberapa hal yang dapat kita lakukan dengan himpunan/set adalah kita bisa menggunakan macam – macam istilah pada himpunan matematika, seperti Union

(Gabungan), Intersection (Irisan), dan lain – lain. Berikut adalah fungsi – fungsi yang dapat dilakukan oleh set.



```
set.py - C:/M.ardi/set.py Python 3.9.0 Shell
File Edit Format Run C File Edit Shell Debug Options Window Help
#Muhammad Ardiansyah Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/M.ardi/set.py =====
['_add_', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'count', 'index']
>>> |
```

Contoh penggunaan set dengan menggunakan fungsi union (gabungan) dan intersection (irisan) adalah dengan perintah “**print>NamaSet.union>NamaSetLain))**” dan “**print>NamaSet.intersection>NamaSetLain))**” seperti gambar berikut.



```
set.py - C:/M.ardi/set.py (3.9.0) Python 3.9.0 Shell
File Edit Format Run Options Window File Edit Shell Debug Options Window
#Muhammad Ardiansyah Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/M.ardi/set.py =====
ganjil = {1,3,5,7,9}
genap = {2,4,6,8,10}
prima = {2,3,5,7}
print(ganjil.union(genap))
print(ganjil.intersection(prima))
>>> |
```

Kesimpulan

Tuple adalah sebuah tipe data yang mirip dengan list yang memiliki kelebihan dan kekurangan sendiri jika dibandingkan dengan list. Keunggulannya diantaranya yaitu dapat diproses lebih cepat dan memiliki memori yang lebih kecil dibandingkan list. Sedangkan kekurangannya yaitu memiliki fungsi yang sangat sedikit, berbeda dengan list yang memiliki cukup banyak fungsi. Perbedaan yang sangat jelas adalah pada sintaknya, tuple menggunakan kurung biasa (()), sedangkan pada list menggunakan kurung siku ([]).

Set dalam bahasa indonesia itu sama dengan himpunan yang memiliki karakteristik tidak mempunyai urutan, frekuensi datanya ketika ada data yang sama hanya akan dihitung satu. Ada dua cara dalam membuat set. Pertama, secara langsung dibuat seperti membuat list/tuple tetapi dengan menggunakan kurung kurawa ({ }). Kedua, dengan cara menyatakan variable sebagai set terlebih dahulu, dan diikuti dengan perintah 'add' untuk menambahkan data kedalam set. Karena set tidak mengenal urutan, sehingga tidak ada istilah 'index' dalam set. Walaupun begitu, kita tetap bisa mengurutkan berdasarkan abjad dengan perintah 'sorted'.