



SET

Set

- **Set** dalam bahasa pemrograman **Python** adalah salah satu metode penanganan kumpulan data dengan tipe data gabungan, selain *list*, *tuple*, dan *dictionary*.
- Ada 2 jenis set, yaitu:
 1. Set dinamis (*Mutable* set) → Ukurannya bisa diubah → Key: **set**
 2. Set statis (*Immutable* set) → Isinya tidak bisa diubah → Key: **frozenset**
- **Set** tidak menyimpan data yang **duplikasi/ganda** (*double*).

Format Penulisan Set

```
variable = set(index-0, index-1, ..., index-n)
```

```
variable = frozenset(index-0, index-1, ..., index-n)
```

Contoh:

A = **set**("Hallo sahabat")

Set dinamis

B = **frozenset**("Robonesia")

Set statis

Membuat Set

```
A = set("Belajar Python")          # Set dinamis
print(A)
print("Tipe data dari A =", type(A))
print('')
B = frozenset("More Than Robotics Learning")    # Set statis
print(B)
print("Tipe data dari B =", type(B))
```

```
# Output
{'j', 'a', 'y', 'h', 'o', 'P', 'B', 'e', 'r', 'n', 'l', 't', ' '}
Tipe data dari A = <class 'set'>

frozenset({'t', 'R', 'M', 'a', 'c', 'L', 's', 'i', 'b', 'g', 'o', 'h', 'e', 'r', 'T', 'n', ' '})
Tipe data dari B = <class 'frozenset'>
```

Operasi Pada Set



1. Menambah Elemen Set

Untuk menambah elemen pada suatu **set** dapat dilakukan dengan menggunakan fungsi **add()** dan **update()**.

```
C = set("Robonesia")           # Sebuah set
print("Nilai set C = ", C)
C.add("Oke!")                  # Cara 1, Menambah elemen set
print("Nilai set C baru = ", C)
print('')
D = set("Belajar Pemrograman")
print("Nilai set D = ", D)
D.update("Python")             # Cara 2, Menambah elemen set
print("Nilai set D baru = ", D)
```

```
# Output
Nilai set C = {'e', 'i', 's', 'b', 'a', 'n', 'o', 'R'}
Nilai set C baru = {'e', 'Oke!', 'i', 's', 'b', 'a', 'n', 'o', 'R'}

Nilai set D = {'e', 'g', 'l', 'P', 'B', 'm', ' ', 'j', 'a', 'n', 'o', 'r'}
Nilai set D baru = {'e', 'g', 'l', 'y', 'P', 'B', 'm', 't', ' ', 'j', 'a', 'h', 'n', 'o', 'r'}
```

2. Menghapus Elemen Set & Set (1)

Untuk menghapus elemen pada suatu **set** dapat dilakukan dengan menggunakan fungsi **remove()** dan untuk menghapus suatu set dapat dilakukan dengan menggunakan fungsi **del**.

```
E = set("Welcome to Python")      # Set dinamis (mutable)
print(E)
E.remove("t")                    # Hapus elemen set E, yaitu "t"
print(E)
del E                            # Hapus set E
print(E)
```

```
# Output
{'W', 't', 'c', ' ', 'P', 'l', 'e', 'y', 'o', 'h', 'm', 'n'}
{'W', 'c', ' ', 'P', 'l', 'e', 'y', 'o', 'h', 'm', 'n'}
Traceback (most recent call last):
  File "E:\PROGRAMMING\Pemrograman Python - 2 - PyCharm\pycharm_workspace\set.py", line 36, in <module>
    print(E)
NameError: name 'E' is not defined
```

2. Menghapus Elemen Set & Set (2)

```
F = frozenset("Selamat Datang") # Set statis (immutable)
print(F)
F.remove("D") # Hapus elemen set F, yaitu "D"
print(F)
F.add('Temanku') # Menambah elemen ke dalam set F
print(F)
```

```
# Output
frozenset({'g', 'l', ' ', 'S', 'a', 'e', 'm', 't', 'D', 'n'})
Traceback (most recent call last):
  File "E:\PROGRAMMING\Pemrograman Python - 2 - PyCharm\pycharm_workspace\set.py", line 42, in <module>
    F.remove("D") # Hapus elemen set F, yaitu "D"
AttributeError: 'frozenset' object has no attribute 'remove'
```

```
Traceback (most recent call last):
  File "E:\PROGRAMMING\Pemrograman Python - 2 - PyCharm\pycharm_workspace\set.py", line 45, in <module>
    F.add('Temanku') # Menambah elemen ke dalam set F
AttributeError: 'frozenset' object has no attribute 'add'
```

Catatan:

Operasi **menghapus** dan **menambahkan** elemen **tidak dapat dilakukan** pada **set statis** (*immutable set*)

3. Operasi Keanggotaan (Membership)

Untuk mengetahui suatu data adalah anggota suatu **set** atau bukan, dapat dilakukan menggunakan operator **in** atau **not in**. Nilai balik operator **in** adalah “**True**,” sedangkan operator **not in** adalah “**False**.”

```
H = set("Selamat Pagi")
print("Set H =", H)
print("Apakah S anggota set H? =", "S" in H)
print("Apakah P bukan anggota set H? =", "P" not in H)
print("Apakah u anggota set H? =", "u" in H)
print("Apakah g anggota set H? =", "g" in H)
```

```
# Output
Set H = {' ', 'g', 'm', 'a', 'P', 'i', 't', 'e', 'l', 'S'}
Apakah S anggota set H? = True
Apakah P bukan anggota set H? = False
Apakah u anggota set H? = False
Apakah g anggota set H? = True
```

4. Operasi Gabungan (*Union*) & Irisan (*Intersection*)

Operasi **gabungan** (*Union*) merupakan operasi menggabungkan dua buah set atau lebih dengan menggunakan operator “|”. Sedangkan **irisan** (*Intersection*) merupakan operasi menampilkan elemen-elemen yang sama antara dua buah set atau lebih dengan menggunakan operator “&”.

```
I = set("Belajar Pemrograman")
J = frozenset("Bahasa Python")
print("Set I = ", I)
print("Set J = ", J)
K = I | J          # Operasi gabungan (Union)
L = I & J          # Operasi irisan (Intersection)
print("Union set I | J = ", K)
print("Intersection set I & J = ", L)
```

```
Set I = {'a', 'm', 'n', ' ', 'j', 'B', 'P', 'e', 'r', 'o', 'g', 'l'}
Set J = frozenset({'a', 't', 'n', ' ', 's', 'B', 'P', 'o', 'y', 'h'})
Union set I | J = {'m', 't', 'n', 's', 'P', 'o', 'h', 'y', 'g', 'a', ' ', 'j', 'B', 'e', 'r', 'l'}
Intersection set I & J = {'a', 'n', ' ', 'B', 'P', 'o'}
```

5. Operasi Perbandingan (1)

Operasi **perbandingan** merupakan operasi membandingkan dua buah set atau lebih. Operator yang digunakan yaitu sama dengan (**==**) dan tidak sama dengan (**!=**).

```
M = set("Belajar Python")
N = set("alerBaj nohtyP")          # acak
print("Set M = ", M)
print("Set N = ", N)
print("Apakah M == N? - ", M == N)
print("Apakah M != N? - ", M != N)
```

```
# Output
Set M = {' ', 'l', 'y', 'a', 'h', 'n', 'j', 'e', 'P', 'r', 't', 'B', 'o'}
Set N = {' ', 'l', 'y', 'a', 'h', 'n', 'j', 'e', 'P', 'r', 't', 'B', 'o'}
Apakah M == N? - True
Apakah M != N? - False
```

5. Operasi Perbandingan (2)

>> Superset & Subset

Operasi **perbandingan superset** dan **subset** merupakan operasi membandingkan dua buah set, apakah lebih besar, lebih kecil, atau sama dengan. Operator yang digunakan yaitu **>**, **>=**, **<** dan **<=**.

```
O = set("Python Coding")
P = set("Coding")
print("Set O = ", O)
print("Set P = ", P)
print("Apakah O superset P? - ", O > P)      # O superset P
print("Apakah O superset P? - ", O >= P)     # O superset P
print("Apakah O subset P? - ", O < P)        # O subset P
print("Apakah O subset P? - ", O <= P)       # O subset P
```

O dikatakan **superset P** →
jika semua elemen set P
termasuk dalam set O.

O dikatakan **subset P** →
jika semua elemen set O
termasuk dalam set P.

```
# Output
Set O = {'i', 'C', 'h', 'n', 't', 'o', 'y', 'd', 'P', ' ', 'g'}
Set P = {'i', 'C', 'n', 'o', 'd', 'g'}
Apakah O superset P? - True
Apakah O superset P? - True
Apakah O subset P? - False
Apakah O subset P? - False
```

6. Operasi Komplemen (*Difference*)

Komplemen suatu himpunan/set merupakan elemen-elemen yang hanya **terdapat pada satu set saja** dan **tidak terdapat pada set yang lain**. Operator yang digunakan yaitu tanda minus (-).

```
Q = set("Kota Jogja")
R = frozenset("Kota Batam")
print(" Set Q = ", Q)
print(" Set R = ", R)
S = Q - R          # Q komplemen R
T = R - Q          # R komplemen Q
print(" Q komplemen R = ", S)
print(" R komplemen Q = ", T)
```

```
# Output
Set Q =  {'j', ' ', 't', 'a', 'K', 'o', 'g', 'J'}
Set R =  frozenset({'B', ' ', 't', 'a', 'K', 'o', 'm'})
Q komplemen R =  {'g', 'j', 'J'}
R komplemen Q =  frozenset({'m', 'B'})
```

7. Operasi *Symetric Difference*

Operasi *symmetric difference* merupakan operasi logika **XOR** pada sebuah set, di mana akan menghasilkan semua **elemen berbeda** dari dua buah set atau lebih. Operator yang digunakan adalah “**^**”

```
U = set("Kota Batam")
V = set("Kota Bogor")
print("Set U = ", U)
print("Set V = ", V)
W = U^V          # U symmetric difference V
print("U symmetric difference V = ", W)
```

```
# Output
Set U =  {'K', 'm', 't', 'a', 'o', 'B', ' '}
Set V =  {'K', 'r', 'g', 't', 'a', 'o', 'B', ' '}
U symmetric difference V =  {'m', 'r', 'g'}
```

Operasi Pada Set Menggunakan Fungsi Built-in



Fungsi **Built-In** Untuk Set & Frozenset

Fungsi	Keterangan	Contoh
<code>issubset()</code>	Fungsi subset	<code>X.issubset(A)</code>
<code>issuperset()</code>	Fungsi superset	<code>X.issuperset(A)</code>
<code>union()</code>	Fungsi gabungan	<code>X.union(A)</code>
<code>intersection()</code>	Fungsi irisan	<code>X.intersection(A)</code>
<code>difference()</code>	Fungsi komplemen	<code>X.difference(A)</code>
<code>symmetric_difference</code>	Fungsi symmetric difference	<code>X.symetric_difference(A)</code>

Operasi Menggunakan Fungsi Built-In Pada Set

```
X = set("Robotika")
Y = set("Robonesia")
print("Set X = ", X)
print("Set Y = ", Y)
print('')
print("Apakah X subset Y? - ", X.issubset(Y))
print("Apakah X superset Y? - ", X.issuperset(Y))
print("X Union Y = ", X.union(Y))
print("X Intersection Y = ", X.intersection(Y))
print("X Komplemen Y = ", X.difference(Y))
print("X symetric_difference Y = ", X.symmetric_difference(Y))
```

Output

```
Set X = {'i', 't', 'R', 'a', 'k', 'o', 'b'}
```

```
Set Y = {'i', 'R', 'a', 's', 'b', 'o', 'n', 'e'}
```

```
Apakah X subset Y? - False
```

```
Apakah X superset Y? - False
```

```
X Union Y = {'i', 'e', 't', 'a', 'R', 's', 'k', 'o', 'n', 'b'}
```

```
X Intersection Y = {'i', 'a', 'R', 'o', 'b'}
```

```
X Komplemen Y = {'k', 't'}
```

```
X symetric_difference Y = {'t', 's', 'k', 'n', 'e'}
```

Terima Kasih