**Rangkuman Ekspresi**

Kita sudah berada di penghujung materi Ekspresi. Sampai sini, Anda memiliki pemahaman mendasar mengenai ekspresi yang akan sering Anda jumpai ketika membuat program Python. Mari kita rangkum secara saksama.

**Pengertian Ekspresi**

Ekspresi pada pemrograman merupakan kombinasi dari satu atau lebih variabel, konstanta, operator, dan fungsi yang bermakna untuk menghasilkan suatu nilai dalam tipe tertentu.

**Jenis-Jenis Ekspresi**

Umumnya ekspresi dibagi menjadi dua, yakni **menurut arity dari operator**dan **menurut tipe data yang dihasilkan**.

1. **Ekspresi Menurut Arity dari Operator**
   1. Biner  
      Jenis ekspresi yang memiliki dua operan. Beberapa contohnya adalah **(x + y), (x - y), (x \* y)**, dan sebagainya.
   2. Uner  
      Jenis ekspresi yang memiliki bentuk dasar operasi dengan satu operan. Contohnya adalah increment (x+=1), decrement (x-=1), dan negasi (not x).
2. **Ekspresi Menurut Tipe Data yang Dihasilkan**  
   Jenis ekspresi yang dikelompokkan berdasarkan tipe data yang dihasilkan. Berikut adalah penjelasan lebih detailnya.

| **Jenis** | **Contoh** |
| --- | --- |
| Ekspresi aritmetika:  <numerik> <operator> <numerik> = <numerik> | 2 + 2 = 4, 2 - 2 = 0 |
| Ekspresi relasional:  <numerik> <operator> <numerik> = <boolean> | 3 < 10 = True, 1 > 10 = False |
| Ekspresi logika:  <boolean> <operator> <boolean> = <boolean> | True or False = True |

**Jenis-Jenis Operator**

Selain ekspresi yang memiliki beragam jenis, operator pun memiliki berbagai jenis yang dikelompokkan menjadi operator aritmetika, operator relasional, operator logika, dan operator assignment.

1. **Operator Aritmetika**  
   Operator aritmetika merupakan jenis operator untuk melakukan operasi aritmetika. Perhatikan tabel di bawah ini untuk memahami contoh penerapan operator aritmetika. **Asumsikan x bernilai 11 dan y bernilai 5.**

| **Operator** | **Deskripsi** | **Contoh** |
| --- | --- | --- |
| Penjumlahan (+) | Menambahkan nilai dari kedua operan. | x + y = 16 |
| Pengurangan (-) | Mengurangi nilai dari kedua operan. | x - y = 6 |
| Perkalian (\*) | Mengalikan nilai dari kedua operan. | x \* y = 55 |
| Pembagian bulat (//) | Membagi nilai dari kedua operan. Jika operan adalah integer, hasil operasi adalah bilangan bulat. | x // y = 2 |
| Pembagian riil (/) | Membagi nilai dari kedua operan. Jika operan adalah float, hasil operasi adalah bilangan riil. | x / y = 2.2 |
| Modulo (%) | Sisa hasil pembagian nilai dari dua operan. | x % y = 1 |
| Pangkat (\*\*) | Memangkatkan nilai dari dua operan. | x \*\* y = 161051 |

1. **Operator Relasional**  
   Operator relasional merupakan operator perbandingan antara dua operan yang berupa integer, float, string, ataupun boolean. Hasil akhir dari operator ini adalah nilai bertipe boolean. Perhatikan tabel di bawah untuk memahami contoh penerapan operator relasional. **Asumsikan kedua variabel bertipe numerik atau float dengan x bernilai 5 dan y bernilai 10.**  
     
   Jika variabel bertipe integer atau float, berikut adalah penjelasan detailnya.

| **Operator** | **Deskripsi** | **Contoh** |
| --- | --- | --- |
| Sama dengan (==) | Menghasilkan True, jika kedua operan bukan merupakan bilangan riil dan bernilai sama. | x == y, menghasilkan False. |
| Tidak Sama dengan (!=) | Menghasilkan True, Jika kedua operan tidak bernilai sama. | x != y, menghasilkan True. |
| Lebih Besar dari (>) | Menghasilkan True, jika operan kiri lebih besar dari operan kanan. | x > y, menghasilkan False. |
| Kurang dari (<) | Menghasilkan True, jika operan kanan lebih besar dari operan kiri. | x < y, menghasilkan True. |
| Lebih Besar dari Sama dengan (>=) | Menghasilkan True, jika operan kiri lebih besar atau sama dengan operan kanan. | x >= y, menghasilkan False |
| Kurang dari Sama dengan (<=) | Menghasilkan True, jika operan kanan lebih besar atau sama dengan operan kiri. | x <= y, menghasilkan True |

1. Jika variabel adalah bertipe string, berikut penjelasan detailnya. **Asumsikan x bernilai "Dicoding" dan y bernilai "Indonesia".**

| **Operator** | **Deskripsi** | **Contoh** |
| --- | --- | --- |
| Sama dengan (==) | Menghasilkan True, jika kedua string memiliki nilai yang identik/sama persis. | x == y, menghasilkan False. |
| Tidak Sama dengan (!=) | Menghasilkan True, jika kedua string memiliki nilai yang tidak sama. | x != y, menghasilkan True. |
| Lebih Besar dari (>) | Menghasilkan True, jika **huruf pertama pada string pertama** lebih BESAR dari **huruf pertama pada string kedua** dalam urutan alfabet. | x > y, menghasilkan False. |
| Kurang dari (<) | Menghasilkan True, jika **huruf pertama pada string pertama** lebih KECIL dari **huruf pertama pada string kedua** dalam urutan alfabet. | x < y, menghasilkan True. |
| Lebih Besar dari Sama dengan (>=) | Menghasilkan True, jika **huruf pertama pada string pertama** lebih besar atau sama dengan dari **huruf pertama pada string kedua** dalam urutan alfabet. | x >= y, menghasilkan False. |
| Kurang dari Sama dengan (<=) | Menghasilkan True, jika **huruf pertama pada string pertama** lebih kecil atau sama dengan dari **huruf pertama pada string kedua** dalam urutan alfabet. | x <= y, menghasilkan True. |

1. **Operator Logika**  
   Operator logika merupakan jenis operator untuk melakukan operasi logika dengan kedua operannya bertipe boolean. Hasil akhir dari operasi ini adalah nilai bertipe boolean. Perhatikan kode di bawah ini untuk memahami contoh penerapannya, **asumsikan bahwa p bernilai True dan q bernilai False.**

| **Operator** | **Deskripsi** | **Contoh** |
| --- | --- | --- |
| "AND" atau "&" | Logika yang hanya menghasilkan True jika kedua operan bernilai True. | p and q = False, p & q = False |
| "OR" atau "|" | Logika yang menghasilkan True jika salah satu dari kedua operan bernilai True. | p or q = True, p | q = True |
| NOT | Logika yang bertujuan untuk membalikkan nilai logika dari operannya. | not q = True |

1. **Operator Assignment**  
   Operator ini bertujuan untuk melakukan proses assignment atau pemberian nilai pada suatu variabel dengan nilai tetap. Perhatikan tabel di bawah ini untuk memahami contoh penerapan operator assignment. **Asumsikan x bernilai 11 dan y bernilai 5.**

| **Operator** | **Deskripsi** | **Contoh** |
| --- | --- | --- |
| += | Menyederhanakan operasi x = x + y. | x += y, menghasilkan nilai 16. |
| -= | Menyederhanakan operasi x = x - y. | x -= y, menghasilkan nilai 6. |
| \*= | Menyederhanakan operasi x = x \* y. | x \*= y, menghasilkan nilai 55. |
| /= | Menyederhanakan operasi x = x / y. | x /= y, menghasilkan nilai 2.2. |
| %= | Menyederhanakan operasi x = x % y. | x %= y, menghasilkan nilai 1 |