**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN KOMPUTER**

**MODUL 3**

**OPERATOR**

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah

Praktikum Basis Data

**A logo for a university

Description automatically generated**

Disusun Oleh :

AHSAN MAULANA RIZQI

104062400071

S1 BD-04-B

**PROGRAM STUDI S1 BISNIS DIGITAL**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

1. **Dasar Teori**
   1. Pengantar Operator

Operator adalah simbol yang memerintahkan komputer untuk melakukan satu tugas sederhana. *Operand* adalah data, ekspresi , atau nilai yang menjadi dasar tindakan atau pekerjaan operator. Operator Python adalah simbol khusus yang melakukan operasi tertentu pada nilai atau variabel yangdisebut operand, seperti operator aritmatika (+), operator perbandingan (==), operator logika (*and*), operator keanggotaan (in), operator penugasan (+=), dan operator identitas (is). Beberapa operator yang terdapat pada bahasa pemrograman python adalah:

* + - 1. Operator Aritmatika (*Arithmetic Operators*).
      2. Operator Perbandingan (*Comparison / Relational Operators*).
      3. Operator Penugasan (*Assignment Operators*).
      4. Operator Logika (*Logical Operators*).
      5. Operator Keanggotaan (*Membership Operators*).
      6. Operator Identitas (*Identity Operators*).
  1. Jenis-Jenis Operator
     + 1. Operator Aritmatika (*Arithmetic Operators*).

Operator aritmatika adalah operator umum yang biasa digunakan untuk melakukan perhitungan aritmatika. Operator ini terdiri dari:

*Table 1. Jenis-jenis operator aritmatika*

|  |  |  |  |
| --- | --- | --- | --- |
| *Operator* | *Symbol* | Contoh | Keterangan |
| Penjumlahan | + | 2 + 5 = 7 | Menjumlahkan nilai dari masing masing *operand* |
| Pengurangan | - | 5 – 2 = 3 | Mengurangi nilai *operand* sebelah kiri dengan nilai *operand* sebelah kanan |
| Perkalian | \* | 2 \* 5 = 10 | Mengalikan nilai *operand* |
| Pembagian | / | 10 / 5 = 2 | Membagi nilai *operand* di sebelah kiri dengan nilai *operand* di sebelah kanan |
| Sisa bagi | % | 11% 2 = 1 | Sisa bagi dari nilai *operand* di sebelah kiri ketika dibagi dengan nilai *operand* di sebelah kanan |
| Pangkat | \*\* | 8 \*\* 2 = 64 | Memangkatkan nilai *operand* sebelah kiri dengan nilai *operand* di sebelah kanan |
| Pembagian Bulat | // | 10 // 3 = 3 | Seperti halnya operator pembagian tetapi dibulatkan ke bawah atau angka di belakang koma dihilangkan |

* + - 1. Operator Perbandingan (*Comparison / Relational Operators*).

Perbandingan adalah operator yang digunakan untuk membandingkan nilai dari masing-masing operand. Hasil dari operasi menggunakan operator perbandingan hanya ada 2 yaitu *True* dan *False*. Operator ini terdiri dari:

*Table 2. jenis-jenis operator perbandingan*

|  |  |  |  |
| --- | --- | --- | --- |
| *Operator* | *Symbol* | Contoh | Keterangan |
| Kurang dari | < | 2 < 5  (*False*) | Bernilai *False* karena nilai *operand* di sebelah kiri lebih besar dari nilai *operand* di sebelah kanan sedangkan *operator* yang digunakan adalah < |
| Lebih dari | > | 2 > 5  (*True)* | Bernilai *True* karena nilai *operand* di sebelah kiri lebih besar dari nilai *operand* di sebelah kanan dan operator yang digunakan adalah > |
| Kurang dari atau sama dengan | <= | 2 <= 5  (*True)* | Bernilai *True* karena nilai *operand* di sebelah kiri kurang dari atau sama dengan nilai *operand* di sebelah kanan dan operator yang digunakan <= |
| Lebih dari atau sama dengan | >= | 5 >= 10  (*False*) | Bernilai *False* karena nilai *operand* di sebelah kiri lebih kecil dari atau sama dengan nilai *operand* di sebelah kanan dan operator yang digunakan >= |
| Sama dengan | == | 10 == 5 | Bernilai *False* karena nilai operand di sebelah kiri tidak sama dengan nilai *operand* di sebelah kanan dan operator yang digunakan adalah == |
| Tidak sama dengan | != | 8 != 2 | Bernilai True karena nilai operand di sebelah kiri tidak sama dengan nilai operand di sebelah kanan dan operator yang digunakan adalah != |

* + - 1. Operator Penugasan (*Assignment Operators*).

Operator penugasan adalah operator yang digunakan untuk memberikan atau memodifikasi sebuah nilai dalam sebuah variabel. Operator ini terdiri dari:

*Table 3. jenis-jenis operator penugasan*

|  |  |  |  |
| --- | --- | --- | --- |
| *Operator* | *Symbol* | Contoh | Keterangan |
| Sama dengan | = | a = 5 | Memberikan nilai 5 pada variabel a |
| Tambah sama dengan | += | a += 5 | Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri ditambah dengan nilai 5 atau (a = a + 5) |
| Kurang sama dengan | -= | a -= 5 | Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dikurang dengan nilai 5 atau (a = a - 5) |
| Kali sama dengan | \*= | a \*= 5 | Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dikali dengan nilai 10 atau (a = a \* 10) |
| Bagi sama  dengan | **/=** | a /= 2 | Memberikan nilai kepada variabel **a** dengan nilai variabel itu sendiri dibagi dengan nilai 2 atau (a = a / 2) |
| Sisa bagi sama dengan | **%=** | a %= 2 | Memberikan nilai kepada variabel **a** dengan nilai variabel itu sendiri dibagi dengan nilai **2** atau (a = a % 2) tetapi hasil yang ditampilkan adalah sisa bagi |
| Pangkat sama  dengan | **\*\*=** | a \*\*= 3 | Memberikan nilai kepada variabel **a** dengan nilai variabel itu sendiri dipangkatkan dengan nilai **3** atau (a = a \*\* 3) |
| Pembagian bulat sama dengan | **//=** | a //= 3 | Memberikan nilai kepada variabel **a** dengan nilai variabel itu sendiri dibagi dengan nilai **3** atau (a = a // 3) tetapi hasil yang ditampilkan adalah nilai pembagian yang dibulatkan |

* + - 1. Operator Logika (*Logical Operators*).

Operator Logika digunakan untuk membandingkan dua *Operand* atau dua nilai yang bertipe *Boolean* dan akan menghasilkan nilai *Boolean* yaitu *True* atau *False*. Berikut beberapa jenis operatornya:

*Table 4. jenis-jenis operator logika*

|  |  |  |
| --- | --- | --- |
| *Operator* | Contoh | Keterangan |
| and | 7 > 2 and 2 > 3  (False) | Bernilai *True* karena kedua *operand* bernilai True. 7 > 2 = *True*, 2 > 1 = *True*. Jika salah satu atau kedua *operand* tersebut bernilai *False* maka hasilnya *False* |
| or | 7 < 2 or 2 > 1  (False) | Bernilai *True* karena salah satu *operand* bernilai *True*. 7 < 2 = *False*, 2 > 1 = *True.* Jika salah satu atau kedua *operand* tersebut bernilai True maka hasilnya *True*. Hasil akan bernilai *False* ketika kedua *operand* bernilai *False*. |
| not | a = not True  (False) | Bernilai *False* karena nilai dari *operand* bernilai *True*. Jika nilai *operand* awalnya bernilai *False* maka hasilnya akan bernilai *True*, begitupun sebaliknya |

* + - 1. Operator Keanggotaan (*Membership Operators*).

Operator keanggotaan adalah operator yang bisa digunakan untuk memeriksa apakah ada sebuah nilai yang menjadi bagian keanggotaan dari sebuah *sequence*. *Sequence* bisa terdiri dari sebuah *list, string* atau *tuple*. Hasil dari operasi menggunakan operator keanggotaan hanya ada 2 yaitu *True* dan *False*. Operator ini terdiri dari:

*Table 5. jenis-jenis operator keanggotaan*

|  |  |  |
| --- | --- | --- |
| *Operator* | Contoh | Keterangan |
| in | list\_data = [2,3,4,5]  5 in list\_data  (True) | Dilakukan pemeriksaan terhadap sebuah list apakah ada nilai 5 di dalam list dengan nama variabel list\_data. Karena nilai 5 terdapat dalam list tersebut, hasil yang ditampilkan adalah True |
| not in | list\_data = [2,4,5,8]  8 not in list\_data  (False) | Dilakukan pemeriksaan terhadap sebuah list apakah nilai 8 tidak ada di dalam list dengan nama variabel list\_data. Karena nilai 8 ada di dalam list tersebut, hasil yang ditampilkan adalah False |

* + - 1. Operator Identitas (*Identity Operators*).

*Table 6. jenis-jenis operator identitas*

|  |  |  |
| --- | --- | --- |
| *Operator* | Contoh | Keterangan |
| is | a = 10  b = 910  a is b  (False) | Dilakukan pemeriksaan apakah nilai *operand* di sebelah kiri memiliki identitas memori yang sama dengan nilai *operand* yang berada di sebelah kanan. Jika sama maka akan bernilai *True*, jika tidak akan bernilai *False*. |
| is not | a = 10  b = 10  a is not b  (True) | Dilakukan pemeriksaan apakah nilai *operand* di sebelah kiri memiliki identitas memori yang berbeda dengan nilai *operand* yang berada di sebelah kanan. Jika sama maka akan bernilai *True*, jika tidak akan bernilai *False*. |

1. **Pembahasan Tugas Guided**
   1. Operator Aritmatika.

A screenshot of a computer code

AI-generated content may be incorrect.

*Gambar 1. Code operator aritmatika*

*Gambar 2. Hasil output operator aritmatika*

* 1. Operator Perbandingan (Comparison / Relational Operators).

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 3. Code operator perbandingan*

*Gambar 4. Hasil output operator perbandingan*

* 1. Operator Penugasan (Assignment Operators).

A screenshot of a computer program

AI-generated content may be incorrect.

*Gambar 5. Code operator penugasan*

*Gambar 6. Hasil output operator penugasan*

* 1. Operator Logika (Logical Operators).

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 7. Code operator logika*

*Gambar 8. Hasil output operator*

* 1. Operator Keanggotaan (Membership Operators).

A screenshot of a computer code

AI-generated content may be incorrect.

*Gambar 9. Code operator logika*

*Gambar 10. Hasil output operator logika*

* 1. Identitas (Identity Operators).

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 11. Code Identity Operators*

*Gambar 12. Hasil output Identity Operators*

1. **Pembahasan Tugas Unguided**
   1. Jika nilai a adalah 9 dan b adalah 5, buatlah kodingan yang menjawab pertanyaan berikut:
2. Apakah a sama dengan b?
3. Apakah a < b?
4. Apakah a > b?
5. Apakah a <= b?
6. Apakah a >= b?
7. Apakah a != b?

A screenshot of a computer program

AI-generated content may be incorrect.

Gambar 13. Code unguided 1

Gambar 14. Hasil output unguided 1

* 1. Jika nilai a adalah 5 dan b adalah 6, buat kodingan untuk menjawab pertanyaan berikut:
* a + b = ?
* b - a = ?
* a \* b = ?
* a / b = ?

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 15. Code unguided 2*

*Gambar 16. Hasil output unguided 2*

1. **Ringkasan**