**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN KOMPUTER**

**MODUL 4**

**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 > 1  (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*).

Operator identitas adalah operator yang digunakan untuk memeriksa identitas dari nilai apakah memiliki identitas memori yang sama atau tidak. Hasil dari operasi menggunakan operator keanggotaan hanya ada 2 yaitu *True* dan *False*. Operator ini terdiri dari:

*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.

Pada gambar 1 terdapat kode program berupa operasi aritmatika dengan dua variabel yaitu variabel x dengan nilai 9 dan y dengan nilai 2. Setiap operasi aritmatika yang dilakukan dengan operand varibel x dan variabel y akan ditampilkan ke layer menggunakan fungsi print().

A screenshot of a computer code

AI-generated content may be incorrect.

*Gambar 1. Code operator aritmatika*

Operasi penjumlahan x + y menghasilkan nilai 11. Operasi pengurangan x - y menghasilkan nilai 7. Operasi perkalian x \* y mengasilkan nilai 18. Operasi pembagian x / y menghasilkan nilai 4,5. Operasi pembagian bulat x // y menghasilkan nilai 4. Operasi modulus x % y menghasilkan nilai 1. Operasi perpangkatan x \*\* y menghasilkan nilai 81. Hasil Output sesuai dengan gambar 2.

*Gambar 2. Hasil output operator aritmatika*

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

Pada gambar 3 terdapat kode program berupa operasi perbandingan dengan dua variabel yaitu variabel x dengan nilai 5 dan y dengan nilai 4. Setiap operasi perbandingan yang dilakukan dengan operand varibel x dan variabel y akan mengahasilkan tipe data Boolean yang berpa *True* atau *False*.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 3. Code operator perbandingan*

Operator lebih besar pada code x > y menghasilkan nilai *True*, karena 5 lebih besar dari 4. Operator kurang pada code x < y menghasilkan nilai *False*, karena 5 tidak lebih kecil dari 4. operator sama dengan pada code x = y menghasilkan nilai *False*, karena 5 tidak sama dengan 4. operator tidak sama dengan pada kode x != y menghasilkan nilai *True*, karena 5 tidak sama dengan 4. operator lebih besar sama dengan pada kode x >= y menghasilkan nilai *True*, karena 5 lebih dari sama dengan 4. Operator lebih kuranga dari sama dengan pada kode x <= y menghasilkan nilai *False*, karena 5 tidak kurang dari sama dengan 4. Pada terminal akan menghasilkan output sesuai gambar 4.

*Gambar 4. Hasil output operator perbandingan*

* 1. Operator Penugasan (Assignment Operators).

Kode pada gambar 5 merupakan implementasi penggunaan operator penugasan (assignment operators) dalam Python. Operator ini digunakan untuk melakukan operasi aritmatika sekaligus dengan memperbarui nilai dari variabel dalam satu langkah.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 5. Code operator penugasan*

Variabel a yang bernilai 8 sebagai nilai awal untuk melakukan operasi aritmatika secara berurutan. Nilai ini akan dimodifikasi secara bertahap menggunakan operator penugasan. Operator += menambahkan nilai 2 ke a dengan niali a yang awalnya 8 di tambahkan dengan 2 menghasi lkan niali baru menjadi 10. Operator -= mengurangkan nilai a yang baru itu 10 dengan 3 menhasilkan nilai baru menjadi 7. Operator \*= mengkalikan nilai a yang baru yaitu 7 dengan 10 akan menghasilkan nilai baru menjadi 70. Operator /= membagi nilai a yang baru yaitu 70 dengan 4 akan menghasilkan nilai baru yaitu 17,5. Operator \*\*= mempangkatkan nilai a baru yaitu 17,5 dengan 2 akan menghasilkan nilai akhir menjadi 306,25.

*Gambar 6. Hasil output operator penugasan*

* 1. Operator Logika (Logical Operators).

Kode pada gambar 7 merupakan implementasi dari operator logika (*and*, *or*, dan *not*) dalam Bahasa pemprogaman Python. Operator ini digunakan untuk membandingkan dua nilai boolean. Menghsilkan nilai *output* berupa *True* atau *False*

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 7. Code operator logika*

Variabel x yang bernilai *True* dan variabel y yang bernilai *False*. Operator *and* digunakan untuk memvalidasi apakah kedua nilai mernilai *True*. Karena x bernilai *True* dan y bernilai *False* maka x and y menghasilkan nilai *False* karena kedua niai harus *True* untuk menghasilkan nilai *True*. Operator *or* digunakan untuk memvalidasi apakah ada salah satu yang nilai bernilai *True*. Karena x bernilai *True* dan y bernilai *False* maka x *or* y menghasilkan nilai *True* karena ada salah satu yang bernilai *True*. Operator *not* digunakan untuk menghasilkan kebalikan dari nilai *True* atau *False*. Karena x bernillai *True* maka *not* x menghasilkan nilai *False*.

*Gambar 8. Hasil output operator*

* 1. Operator Keanggotaan (Membership Operators).

Kode pada gambar 9 menunjukan inplementasi dari penggunaan operator keanggotaan (*in* dan *not* *in*) dalam Bahasa pemprogaman python. Operator ini digunakan memeriksa apakah ada sebuah nilai yang menjadi bagian keanggotaan dari sebuah *sequence*. *Sequence* bisa terdiri dari sebuah *list, string* atau *tuple.* Menghasilkan nilai *output* berupa *True* atau *False*.

A screenshot of a computer code

AI-generated content may be incorrect.

*Gambar 9. Code operator logika*

Variabel x merupakan *list* yang beranggotakan [1,5,9] dan variabel y yang bernilai 5. Operator *in* digunakan untuk memeriksa apakah y termasuk kedalam anggota variabel x. Kode y *in* x akan menghasilkan nilai *True,* karena y termasuk anggota x. operator not in digunakan untuk memeriksa apakah y tidak termasuk kedalam anggota variabel x. Kode y *not* *in* x menghasilkan nilai *False,* karena y termasuk anggota x.

*Gambar 10. Hasil output operator logika*

* 1. Operator Identitas (Identity Operators).

Kode pada gambar 11 menunjukan implementasi penggunaan operator identitas (*is* dan *is* *not*) dalam python. Operator ini digunakan untuk memeriksa identitas dari nilai apakah memiliki identitas memori yang sama atau tidak. Menghasilkan *output* benrupa nilai *True* atau *False*.

A screenshot of a computer

AI-generated content may be incorrect.

*Gambar 11. Code Identity Operators*

Variabel x bernilai 4 dan variabel y yang diinisialisasi nilai x, sehingga y memiliki nilai 4. Operator is untuk memeriksa variabel x dan y memiliki identitas memori yang sama. Kode x *is* y menghasilkan nilai *True*, karena y memiliki identitas memori yang sama.

Variabel x bernilai 10 dan variabel y bernilai 4, nilai x digunakan untuk nilai yang divadidasi dan nilai y digunakan untuk nilai yang memvadidasi. Operator *is* *not* untuk memeriksa identitas memori yang tidak sama. Kode y *is* *not* x menghasilkan nilai *True*, karena y tidaak memiliki identitas memori yang sama.

*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 screen shot of a computer

AI-generated content may be incorrect.

*Gambar 13. Code unguided 1*

Operator sama dengan pada code a = b menghasilkan nilai *False*, karena 9 tidak sama dengan 5. Operator kurang dari pada code a < b menghasilkan nilai *False*, karena 9 tidak kurang dari 5. Operator lebih dari pada code a > b menghasilkan nilai *True*, karena 9 lebih dari 5. operator kurang dari sama dengan pada kode a <= b menghasilkan nilai *False*, karena 9 lebih dari sama dengan 5. operator lebih dari sama dengan pada kode a >= b menghasilkan nilai *True*, karena 9 lebih dari sama dengan 5. Operator tidak sama dengan pada kode a != b menghasilkan nilai *True*, karena 9 tidak sama dengan 5. Pada terminal akan menghasilkan output sesuai gambar 14.

*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*

Variabel a bernilai 5 dan variabel b bernilai 6. Operasi penjumlahan a + b menghasilkan nilai 11. Operasi pengurangan b - a menghasilkan nilai 1. Operasi perkalian x \* y mengasilkan nilai 30. Operasi pembagian x / y menghasilkan nilai 0,833…4. Hasil Output sesuai dengan gambar 16.

*Gambar 16. Hasil output unguided 2*

1. **Ringkasan**