

LKPD 2.1

MENERAPKAN VARIABEL DAN TIPE DATA

Tujuan:

1. Siswa mampu menerapkan tipe data yang sesuai dan memahami konsekuensi penggunaan tipe data yang salah
2. Siswa mampu menerapkan tipe data dan operator, serta mengintegrasikannya dengan fungsi input dan output

Instruksi:

1. Cobalah skenario berikut ini (pilih minimal 1 bahasa pemrograman), dan simpulkan konsekuensi yang terjadi jika Anda menggunakan tipe data yang salah dalam program.

Skenario Penggunaan Tipe Data yang Salah

Contoh sederhana dalam bahasa Python yang menunjukkan akibat dari penggunaan tipe data yang salah:

```
1. # Contoh 1: Penjumlahan angka dan string
2. angka = 10
3. teks = "5"
4. hasil = angka + teks
5. print(hasil)
```

Pada contoh di atas, variabel `angka` bertipe *integer* dan `teks` bertipe *string*. Ketika mencoba menjumlahkannya, akan terjadi kesalahan tipe (*TypeError*) karena Python tidak dapat menambahkan *integer* dan *string* secara langsung.

Untuk menghindari kesalahan tersebut, pastikan kedua variabel memiliki tipe data yang sesuai sebelum operasi dilakukan.

Output

Clear

```
ERROR!
/tmp/Bge3Hc16yg/main.c:1:3: error: invalid preprocessing directive #Contoh
 1 | # Contoh 1: Penjumlahan angka dan string
   | ^~~~~~
/tmp/Bge3Hc16yg/main.c:2:1: warning: data definition has no type or storage class
 2 | angka = 10
   | ^~~~~
/tmp/Bge3Hc16yg/main.c:2:1: error: type defaults to 'int' in declaration of 'angka' [-Wimplicit-int]
/tmp/Bge3Hc16yg/main.c:3:1: error: expected ',', ' or ';' before 'teks'
 3 | teks = "5"
   | ^~~~

=== Code Exited With Errors ===
```

Penulisan yang benar:

```
1. # Konversi string ke integer
2. angka = 10
3. teks = "5"
4. hasil = angka + int(teks)
5. print(hasil) # Output: 15
6.
```

Dengan mengonversi `teks` menjadi *integer* menggunakan `int(teks)`, operasi penjumlahan dapat dilakukan tanpa error.

Contoh dalam JavaScript:

```
1. // Contoh 2: Penggabungan string dan angka
2. let angka = 10;
3. let teks = "5";
4. let hasil = angka + teks;
5. console.log(hasil); // Output: "105"
```

Output

105

=== Code Execution Successful ===

Pada contoh JavaScript di atas, operator `+` digunakan untuk penjumlahan dan penggabungan string. Karena salah satu operannya adalah *string*, JavaScript akan mengkonversi `angka` menjadi string dan menggabungkannya dengan `teks`, menghasilkan "105" sebagai *output*.

Solusi:

Pastikan tipe data sesuai sebelum melakukan operasi.

```
1. // Konversi string ke number
2. let angka = 10;
3. let teks = "5";
4. let hasil = angka + Number(teks);
5. console.log(hasil); // Output: 15
```

Dengan mengonversi `teks` menjadi number menggunakan `Number(teks)`, operasi penjumlahan menghasilkan angka yang diharapkan.

Contoh dalam C:

```
1. #include <stdio.h>
2.
3. int main() {
4.     // Contoh 3: Pembagian integer
5.     int a = 5;
6.     int b = 2;
7.     float hasil = a / b;
8.     printf("Hasil: %f\n", hasil);
9.     return 0;
10. }
```

Pada contoh di atas, `a` dan `b` adalah *integer*. Operasi `a / b` akan menghasilkan pembagian *integer*, yaitu 2, bukan 2.5. Meskipun `hasil` bertipe *float*, nilai yang disimpan sudah berupa 2.0.

Solusi:

Lakukan *casting* salah satu operand ke *float* untuk mendapatkan hasil yang tepat.

```
1. #include <stdio.h>
2.
3. int main() {
4.     // Pembagian dengan casting
5.     int a = 5;
6.     int b = 2;
7.     float hasil = (float)a / b;
8.     printf("Hasil: %f\n", hasil); // Output: 2.500000
9.     return 0;
10. }
```

Dengan melakukan casting `(float)a`, operasi pembagian akan menghasilkan nilai desimal yang akurat.

Kesimpulan:

2. Deklarasikan beberapa variabel dengan tipe data yang berbeda!
3. Gunakan input pengguna untuk memberi nilai ke dalam variabel yang telah Anda deklarasikan, dan tampilkan nilainya menggunakan fungsi output.

Contoh Solusi:

Bahasa C	Bahasa Python	Bahasa JavaScript
		

Contoh Eksekusi

Input:

```
Usia: 20  
Tinggi: 1.68  
Nama: Budi
```

Output:

```
=== HASIL ===  
Usia: 20 tahun  
Tinggi: 1.68 meter  
Nama: Budi
```

