

**UNIVERSITAS IPWIJA**  
PROGRAM STUDI INFORMATIKA (S1)  
Semester Ganjil 2023/2024

Mata Kuliah  
Dosen  
Hari/Tanggal  
Waktu

**LEMBAR JAWABAN - UJIAN TENGAH SEMESTER (UTS)**

: Konsep Pemrograman

: M.Maulana Rachman

: Rabu, 08 November 2023

: 09:00 s/d 11:00

NAMA : RENI DEARNI KRISTI PURBA MATKUL : PEMOGRAMAN

NIM : 202301110029 SEMESTER : SATU

PRODI : INFORMATIKA TANGGAL : 15 NOVEMBER 2023

**1**. **Variabel**

Variabel atau variable dapat diartikan sebagai sesuatu yang dapat berubah. Dalam pemrograman, variabel digunakan untuk menampung suatu nilai dan isinya dapat berubah ketika program dieksekusi. Contohnya:

Ketika kita mendeklarasikan variabel a dengan tipe data integer dan dalam perjalanannya terdapat proses assignment  dari variabel lain, maka isi variabel a akan berubah. Gambarannya adalah sebagai berikut (sintaks yang digunakan adalah bahasa C):

**Kode:**

void main(){

int a, b, c;

a = 10;  b = 20; c = 30;

b = c;      //variabel b berubah nilainya menjadi 30 karena mendapat nilai dari variabel c

a = b + c;      //variabel a berubah nilai menjadi 60 karena mendapat nilai dari operasi variabel b yang saat ini bernilai 30 dijumlahkan dengan variabel c yang juga berilai 30

printf(a);

}

**Output:**

60       //output yang akan keluar adalah 60 dan bukan 30 karena nilai variabel a sudah mengalami perubahan

**Konstanta**

Konstanta atau dalam bahasa Inggris dikenal dengan constant adalah sesuatu yang nilainya tetap. Dalam pemrograman, konstanta juga diartikan sebagai sesuatu yang nilainya tidak dapat berubah. Contohnya:

Dalam bahasa C, konstanta umumnya dapat didefinisikan menggunakan #define dan dapat juga menggunakan const. Sehingga, apabila dalam prosesnya kita mencoba mengisi konstanta yang sudah dideklarasikan dengan nilai lain, saat eksekusi, output yang dihasilkan adalah error karena sifat konstanta yang tidak dapat diubah. Untuk dapat memahami hal tersebut, berikut adalah contoh konstanta dalam program (sintaks yang digunakan adalah bahasa C):

**Kode:**

#define a 10      //mendefinisikan konstanta dengan cara #define

void main(){

const int b = 20;     //mendefinisikan konstanta dengan const

int result;     //result disini merupakan variabel sehingga nilainya dapat berubah, berbeda dengan a dan b yang merupakan konstanta

result = a \* b;

printf(result);

}

**Output:**

200     //Hasil ini diperoleh dari perkalian konstanta a dengan b

**2**. A. Bilangan Bulat (Integer)

Tipe bilangan bulat (Integer) adalah tipe data numerik yang biasa digunakan apabila bertemu dengan bilangan bulat, seperti 1, 27, 100, dll. Bilangan ini juga mengenal nilai positif dan negatif (signed number). Tipe data numerik yang termasuk ke dalam bilangan bulat adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| **Tipe Data** | **Ukuran (Bit)** | ***Range*** |
| Byte | 8 | -128 s/d 127 |
| Short | 16 | -32768 s/d 32767 |
| Int | 32 | -2147483648 s/d 2147483647 |
| Long | 64 | -9223372036854775808 s/d 9223372036854775807 |

Dari keempat tipe data di atas, yang sering digunakan adalah “int” atau Integer. Tipe “byte” dan “short” hanya digunakan pada aplikasi khusus yang berkaitan dengan memori. Sedangkan tipe “long” sangat jarang digunakan karena dirasa tidak memerlukan bilangan yang berkapasitas besar, seperti kapasitas yang diberikan oleh tipe “long”.

#### 1. Byte

Tipe byte biasa digunakan pada saat kita bekerja dengan data stream pada file maupun jaringan. Tipe ini diperlukan saat kita melakukan proses seperti membaca dan menulis. Selain itu, byte juga digunakan saat kita menggunakan data biner yang tidak kompatibel dengan tipe lainnya pada program bahasa Java.

**Contoh 1.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **byte** value = 1;  *//Output*        System.out.println("Nilai dari byte = " + value);  } |

**Contoh 2 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **byte** value;  *//Inisialisasi variabel*        value = 1;  *//Output*        System.out.println{"Nilai dari byte = " + value);  } |

#### 2. Short

Tipe short sangat umum digunakan pada komputer yang berkapasitas 16-bit, sehingga sangat jarang sekali kita temui dengan kapasitas laptop saat ini.

**Contoh 1 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **short** value = 2;  *//Output*        System.out.println("Nilai dari short = " + value);  } |

**Contoh 2 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **short** value;  *//Inisialisasi variabel*        value = 2;  *//Output*        System.out.println("Nilai dari short= " + value);  } |

#### 3. Int

Tipe int termasuk tipe yang sangat sering dipakai saat ini karena merepresentasikan angka dalam pemrograman. Sebab tipe data int dianggap paling efisien daripada tipe bilangan bulat lainnya. Tipe int digunakan pada indeks dalam struktur perulangan maupun dalam array.

Secara teori, setiap ekspresi yang melibatkan tipe integer byte, short, dan long, semuanya harus melalui prosedur int untuk dipromosikan terlebih dahulu sebelum dilakukan perhitungan.

**Contoh 1 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **int** value = 20;  *//Output*        System.out.println("Nilai dari int = " + value);  } |

**Contoh 2 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **int** value;  *//Inisialisasi variabel*        value = 20;  *//Output*        System.out.println("Nilai dari int = " + value);  } |

Tipe data integer dapat juga digabungkan melalui beberapa tipe data seperti berikut ini :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | **Public** **class** Main {  **public** **static** **void** main (**String** []args)  *//Membuat variabel dengan beberapa tipe data yang berbeda*  **byte** value = 1;  **short** anotherValue = 2;  **int** result = 3;  **int** anotherResult;  *//Menjumlahkan variabel byet, short, dan int*        anotherResult = value + anotherValue + result;  *//Mencetak atau menampilkan hasilnya*        System.out.println("Nilai dari result = " + anotherResult);  } |

#### 4. Long

Tipe long biasa digunakan saat nilainya berada di luar kapasitas rentang tipe int karena tipe long punya range sangat tinggi dibanding tipe data lainnya. Dengan kata lain, tipe long dibutuhkan saat data memiliki range di luar jangkauan tipe int, short, maupun byte.

**Contoh 1 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **long** value = 22121;  *//Output*         System.out.println("Nilai dari long = " + value);  } |

**Contoh 2 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **long** value;  *//Inisialisasi variabel*        value = 22121;  *//Output*        System.out.println("Nilai dari long = " + value);  } |

### B. Bilangan Pecahan (Floating Point)

Tipe bilangan pecahan atau floating point adalah bilangan yang menangani bilangan desimal atau perhitungan secara detail. Karena kemampuannya, float point berbanding terbalik dengan integer. Terdapat dua tipe pada bilangan pecahan ini

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipe** | **Ukuran** | | **Range** | **Presisi (Jumlah Digit)** |
| ***bytes*** | ***bit*** |
| Float | 4 | 32 | +/- 3.4 x 1038 | 6 – 7 |
| Double | 8 | 64 | +/- 1.8 x 10308 | 15 |

#### 1. Float

Tipe float biasa digunakan untuk menandai nilai yang presisi seperti ketelitian tunggal (single precision) dengan menggunakan penyimpanan 32-bit. Tipe ini memiliki kemampuan yang lebih cepat jika digunakan pada prosesor-prosesor tertentu dan pastinya memakan ruang penyimpanan yang lebih kecil dari tipe double.

Pada tipe float, akan bermasalah pada saat nilainya terlalu kecil atau terlalu besar, karena pada penerapannya nilai tersebut menjadi tidak akurat.

**Contoh 1 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **float** ip = 3,7;  *//Output*        System.out.println("IP saya = " + ip);  } |

**Contoh 2 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **float** ip;  *//Inisialisasi variabel*        ip = 3,7;  *//Output*        System.out.ptintln("IP saya = " + ip);  } |

#### 2. Double

Tipe double memiliki tingkat ketelitian secara ganda atau double precision dengan menggunakan ruang penyimpanan 64-bit dalam menyimpan nilainya. Tipe tersebut pastinya memberikan kemampuan menghitung matematis secara lebih cepat dari tipe float. Dalam perhitungan yang bersifat bilangan riil dan menginginkan hasil yang lebih akurat, sebaiknya menggunakan tipe ini.

**Contoh 1 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **double** pi = 3.14285714286;  *//Output*        System.out.println("Nilai pi = " + pi);  } |

**Contoh 2 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **int** pi;  *//Inisialisasi variabel*        pi = 3.14285714286;  *//Output*        System.out.println("Nilai pi = " + pi);  } |

### C. Karakter (Char)

Tipe data karakter tunggal yang biasa didefinisikan dengan tanda petik (‘) di awal dan di akhir karakternya. Tipe ini mengikuti aturan “unicode” sehingga bilangan harus diawali kode “/u”. Tetapi juga biasa menggunakan bilangan heksadesimal dari 0000 sampai FFFF.

**Contoh 1 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **char** value = 'a';  *//Output*        System.out.println("Huruf pertama adalah " + value);  } |

**Contoh 2 :**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **public** **class** Main {  **public** **static** **void** main (**String** [] args)  *//Deklarasi variabel*  **char** value;  *//Inisialisasi variabel*        value = 'a';  *//Output*        System.out.println("Huruf Pertama adalah " + value);  } |

Pada tipe ini terdapat juga karakter-karakter yang tidak dapat diketik melalui keyboard. Pada bahasa java kita bisa menemukan istilah “escape sequence” (pasangan karakter yang dianggap karakter tunggal). Berikut akan dijelaskan contoh escape sequence:

|  |  |  |
| --- | --- | --- |
| ***Escape Squence*** | **Keterangan** | **Nilai Unicode** |
| \ddd | Karakter Octal (ddd) |  |
| \uxxxx | Karakter Unicode Heksadesimal (xxxx) |  |
| \’ | Petik tunggal | \u0027 |
| \\* | Double Quote | \u0022 |
| \\ | Backslash | \u005c |
| \r | Carriage return | \u000d |
| \n | Baris baru (line feed) | \u000a |
| \f | Form feed |  |
| \t | Tab | \u0009 |
| \b | Backspace | \u0008 |

### D. Boolean

Tipe data boolean merupakan tipe yang memiliki dua nilai yaitu benar (true) atau salah (false). Nilai yang digunakan pada tipe ini sangat penting dalam mengambil keputusan suatu kejadian tertentu.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **public** **class** Main {  **public** **class** **void** main (**String** [] args)  *//Deklarasi variabel*  **boolean** value = **true**;  *//Output*          System.out.println("Nilai boolean = " + value);      }  } |

Nah, secara keseluruhan akan jadi seperti ini:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | **class** Main {  **public** **static** **void** main (**String** [] args){  *//Tipe data primitif*  **long** sampleOfLong = 546767226531;  **int** sampleOfInteger = 2235641;  **short** sampleOfShort = 741;  **byte** sampleOfByte = 34;  **float** sampleOfFloat = 1.733;  **double** sampleOfDouble = 4.967;  **char** sampleOfChar = 'C';  **boolean** sampleOfBoolean = **true**;            System.out.println("Nilai Long : " + sampleOfLong);          System.out.println("Nilai Integer : " + sampleOfInteger);          System.out.println("Nilai Short : " + sampleOfShort);          System.out.println("Nilai Byte : " + sampleOfByte);          System.out.println("Nilai Float : " + sampleOfFloat);          System.out.println("Nilai Double : " + sampleOfDouble);          System.out.println("Nilai Char : " + sampleOfChar);          System.out.println("Nilai Boolean : " + sampleOfBoolean);      }  } |

3.