

LAPORAN

PRAKTIKUM STRUKTUR DATA

MODUL 1



Disusun oleh :

Nama : Fidelia Ping

NIM : 245410012

Kelas : Informatika 1

PROGRAM STUDI INFORMATIKA
PROGRAM SARJANA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA
YOGYAKARTA
2025

MODUL 1

TIPE-TIPE DATA DALAM JAVA

A. DASAR TEORI

Setiap bahasa pemrograman memiliki tipe data yang spesifik. Tipe data akan digunakan untuk mendeklarasikan variable yang digunakan. Tipe data digunakan untuk menentukan bentuk data yang dapat ditampung oleh sebuah variabel.

Dalam java terdapat dua jenis tipe data. Yang pertama adalah tipe data primitive yang merupakan tipe data bawaan dari compiler java. Tipe data ini akan anda pelajari pada modul 1 ini.

Sedangkan tipe data yang kedua adalah tipe data buatan yang baru akan anda pelajari pada modul 2.

Dalam bahasa Java, tipe data primitive dibedakan menjadi tiga bagian yaitu :

1. Tipe Data Alphabetic

- Char
- String

2. Tipe Data Alphanumeric

a. Tipe data Bilangan Bulat

- Byte
- Short
- Int
- Long

b. Tipe Data Bilangan Pecahan

- Float
- Double

3. Tipe Data Boolean

1. Tipe Data Alphabetic

Char (Karakter)

Tipe data char merupakan tipe yang digunakan untuk menyatakan sebuah karakter, bisa berupa huruf/ tandabaca/ simbol, yang didefinisikan dengan diawali dan diakhiri dengan tanda ' (petik tunggal).

Untuk merepresentasikan semua karakter yang ada bahasa Java menggunakan karakter *Unicode* yaitu sekumpulan karakter umum yang terdapat pada semua bahasa, seperti English, Latin, Arab, Yunani dan lain-lainnya. Karakter *Unicode* yang membutuhkan ukuran 16-bit dan memiliki 1680 jenis karakter.

Berikut ini disajikan contoh program sederhana menggunakan tipe data char.

```
public class tipeData {
    public static void main(String[] args) {
        char data1 = 'C';
        System.out.println("Nilai Char : " + data1);
    }
}
Hasil Eksekusi :
Nilai Char : C
Press any key to continue . . .
```

Program 1.1 Contoh pemakaian char

Dalam Java, data bertipe char juga dapat diinputkan melalui keyboard untuk disimpan dalam sebuah variabel char. Namun java tidak menyediakan fungsi khusus untuk membaca masukan bertipe char sehingga perlu dibuat pembacaan karakter menggunakan pembacaan kode unicode menggunakan fungsi `System.in.read()` dan kemudian mengkonversinya menggunakan fungsi `(char)` untuk dapat membacanya. Berikut ini adalah program untuk membaca masukannya.

```

import java.util.Scanner;
public class inputViaKeyboard
{ public static void main(String[] args)
{
    Scanner masukan = new Scanner(System.in);
    int bacaTombol=0;
    char huruf;
    System.out.print("Silakan masukkan sebuah huruf: ");
    try
    { bacaTombol = System.in.read();      }
    catch(java.io.IOException e)
    {   }
    huruf = (char)bacaTombol;
    System.out.println("Huruf yang anda entri adalah : " + huruf);
}
}

Hasil Eksekusi :
Silakan masukkan sebuah huruf: Z □
Huruf yang anda entri adalah : Z
Press any key to continue . .

```

Program 1.2 Contoh pemakaian char Via Keyboard

String

Tipe data String merupakan kumpulan dari tipe data char. Karena merupakan kumpulan char, maka tipe data String dapat digunakan untuk menyimpan kalimat.

Jika dilihat dari unsur pembentuknya, tipe data string bukan merupakan *tipe data primitif*, tetapi sudah merupakan sebuah *objek* yang berisi kumpulan tipe data char.

Berikut ini disajikan contoh program sederhana menggunakan tipe data string.

```

public class tipeData {
    public static void main(String[] args) {
        String data2 = "Namaku Agung Budi Prasetyo";
        System.out.println("Nilai String : " + data2);
    }
}

Hasil Eksekusi :
Nilai String : Namaku Agung Budi Prasetyo
Press any key to continue . .

```

Program 1.3 Contoh pemakaian String

Dalam bahasa java, data bertipe string juga dapat diinputkan melalui keyboard untuk disimpan dalam sebuah variabel string. Untuk keperluan tersebut java telah menyediakan sebuah fungsi untuk membaca masukan yaitu `next()` dan `nextline()`. Namun dari kedua fungsi di atas penulis lebih menyarankan untuk menggunakan fungsi `next()`.

Berikut ini adalah program untuk membaca masukannya.

```
import java.util.Scanner;
public class inputViaKeyboard
{
    public static void main(String[] args)
    {
        Scanner masukan = new Scanner(System.in);
        String kalimat;
        System.out.print("Silakan masukkan sebuah kalimat : ");
        kalimat = masukan.next();
        System.out.println("Kalimat yang anda entri adalah :" + kalimat);
    }
}
Hasil Eksekusi :
Silakan masukkan sebuah kalimat: Namaku Agung Budi Prasetyo□
Kalimat yang anda entri adalah : Namaku Agung Budi Prasetyo
Press any key to continue . . .
```

Program 1.4 Contoh pemakaian String via keyboard

Untuk karakter-karakter yang tidak dapat diketikkan secara langsung melalui keyboard, java menyediakan beberapa *escape sequence* (pasangan karakter yang dianggap sebagai karakter tunggal). *Escape sequence* tidak dianggap sebagai *String*, melainkan tetap sebagai tipe karakter khusus. Berikut ini adalah beberapa contoh *escape sequence*.

Contoh penggunaan escape sequence

```
public class tipeData {
    public static void main(String[] args) {
        String kalimat = "Halo \"Agung Budi Prasetyo\" apa kabar..";
        System.out.println("Nilai String : "+ kalimat);
    }
}
Hasil Eksekusi :
Nilai String : Halo "Agung Budi Prasetyo" apa kabar..
Press any key to continue . . .
```

Program 1.5 Contoh pemakaian escape sequence

2. Tipe Data Aplhanumeric

A. Tipe Bilangan Bulat

Ada empat macam tipe bilangan bulat, di mana masing-masing memiliki jangkauan nilai yang berbeda yaitu byte, short, int dan long.

Tipe	Ukuran	Jangkauan Nilai
byte	8 bit	-128 s/d 127
short	16 bit	-32.768 s/d 32.767
int	32 bit	-2.147.483.648 s/d 2.147.483.647
long	64 bit	-9.223.372.036.854.775.808 s/d 9.223.372.036.854.775.807

Byte

Type data *byte* umumnya digunakan pada saat kita bekerja dengan sebuah data *stream* dari suatu file, memory, maupun jaringan komputer yaitu untuk keperluan proses membaca/menulis berkas. Selain itu, tipe ini juga digunakan saat bekerja dengan data biner. Namun tidak jarang pula tipe *byte* digunakan untuk menyimpan bilangan yang tidak terlalu besar (bilangan di bawah bilangan 127, contohnya umur pegawai).

Short

Tipe data short memiliki ukuran yang sedikit lebih besar dibandingkan byte. Pada umumnya tipe data short digunakan pada komputer-komputer 16-bit yang memang memiliki ukuran bilangan yang terbatas. Tipe short digunakan pula pada aplikasi khusus yang memperhatikan penggunaan memori.

Int

Tipe data int merupakan tipe yang paling banyak dipakai dalam merepresentasikan angka dalam Java. Hal ini karena tipe data int dianggap paling efisien dibandingkan dengan tipe-tipe integer lainnya. Tipe *Int* banyak digunakan untuk indeks dalam struktur pengulangan maupun dalam konstruksi sebuah *array*. Selain itu, secara teori setiap ekspresi yang melibatkan tipe integer *byte*, *short*, *int*, *long* semuanya itu akan dipromosikan ke *int* terlebih dahulu sebelum dilakukan proses perhitungan

Long

Tipe ini digunakan untuk kasus-kasus tertentu yang nilainya berada di luar rentang tipe int, karna tipe ini punya range paling tinggi dibanding Integer lainnya. Dengan kata lain, tipe long terpaksa digunakan jika data memiliki range diluar range int.

Semua bilangan bulat dalam Java secara default dianggap sebagai tipe int. Sedangkan bilangan yang ingin dikategorikan sebagai long harus diakhiri dengan huruf L. Misalnya : 18102006L. Berikut ini disajikan contoh program sederhana menggunakan tipe data byte, short, int, dan long.

```
public class tipeData {
    public static void main(String[] args) {
        // Tipe data primitif
        byte data3 = 34;
        short data4 = 714;
        int data5 = 2235641;
        long data6 = 546767226531L;
        System.out.println("Nilai Byte      : " + data3);
        System.out.println("Nilai Short     : " + data4);
        System.out.println("Nilai Int       : " + data5);
        System.out.println("Nilai Long      : " + data6);
    }
}
Hasil Eksekusi :
Nilai Byte      : 34
Nilai Short     : 714
Nilai Int       : 2235641
Nilai Long      : 546767226531
Press any key to continue . . .
```

Program 1.6 Contoh pemakaian byte, short, int dan long

Dalam bahasa java, data bertipe byte, short, int dan long juga dapat diinputkan melalui keyboard untuk disimpan dalam sebuah variabel dengan tipe yang sama. Untuk keperluan tersebut java telah menyediakan sebuah fungsi untuk membaca masukan yaitu *nextByte()* untuk tipe data byte, *nextShort()* untuk tipe data short, *nextInt()* untuk tipe data int, dan *nextLong()* untuk tipe data long.

Berikut ini adalah program untuk membaca masukannya.

```
import java.util.Scanner;
public class inputViaKeyboard
{
    public static void main(String[] args)
    {
        Scanner masukan = new Scanner(System.in);
        byte bilanganByte;
        short bilanganShort;
        int bilanganInt;
        long bilanganLong;
```

3. Tipe Data Boolean

Dalam Java dikenal tipe data boolean yang terdiri dari dua nilai saja, yaitu true dan false. Boolean sangat penting dalam mengevaluasi suatu kondisi, dan sering digunakan untuk menentukan alur program.

```
public class tipeData {
    public static void main(String[] args) {
        // Tipe data primitif
        boolean data9 = true;
        boolean data10 = false;
        System.out.println("Nilai data9 : " + data9);
        System.out.println("Nilai data10 : " + data10);
    }
}
Hasil Eksekusi :
Nilai data9 : true
Nilai data10 : false
Press any key to continue . . .
```

Program 1.9 Contoh pemakaian boolean

Tipe Data Array/ Larik

Selain ketiga tipe data di atas, dalam java terdapat juga tipe data Array/ Larik. Tipe data Array adalah tipe data untuk membuat variabel secara bersusun.

Dengan adanya tipe data array/ larik kita dapat membuat variabel-wariabel kembar yang bertipe data sama. Sebagai contoh dapat kita lihat program berikut ini.

B. PEMBAHASAN PRAKTIK

Praktik 1 (percobaan tentang tipe data integer)

```
public class pembagian {
    public static void main(String[] args) {
        int banyaknyaApel = 5;
        int jumlahAnak = 2;
        int perolehan;
        perolehan = banyaknyaApel / jumlahAnak;
        System.out.println("Masing2 mendapat = " + perolehan);
    }
}
```

```
Masing2 mendapat = 2
Press any key to continue . . .
```

Pembasahan : Ketika program dieksekusi, hasil yang diperoleh Adalah : Masing2 mendapat = 2 Apakah hasilnya 2.5? Tidak, hasilnya bukan 2.5. Mengapa bisa demikian? Hal ini terjadi karena variabel banyaknyaApel, jumlahAnak, dan perolehan semuanya dideklarasikan menggunakan tipe data int (integer). Dalam Java (dan banyak bahasa pemrograman lainnya), ketika operasi pembagian (/) dilakukan antara dua bilangan bulat (int), hasilnya juga akan berupa bilangan bulat. Bagian desimal atau sisa pembagian akan diabaikan (dipotong/dihilangkan/truncated), bukan dibulatkan. $5 / 2 = 2$ dengan sisa \$1\$. Karena tipe data hasil (perolehan) adalah int, maka nilai yang disimpan hanya 2.

Praktik 2 (percobaan tentang tipe data Long)

```
public class cobaLong {
    public static void main(String[] args){
        long coba = 1234567890123;
        System.out.println(coba);
    }
}
```

```
Error: Could not find or load main class cobaLong
Caused by: java.lang.ClassNotFoundException: cobaLong
Press any key to continue . . .
```



```
public class cobaLong {
    public static void main(String[] args){
        long coba = 1234567890123;
        System.out.println(coba);
    }
}
```

```
1234567890123
Press any key to continue . . .
```

Pembasahan : Analisis Kode Program AwalApa yang terjadi sewaktu program di atas dieksekusi? Program akan menghasilkan kesalahan (error) kompilasi. Mengapa bisa demikian? Literal

angka 1234567890123 secara default dianggap sebagai tipe data int oleh kompiler Java. Namun, nilai 1,234,567,890,123 ini melebihi batas maksimum yang bisa disimpan oleh tipe data int (yaitu 2,147,483,647). Karena nilai literal ini terlalu besar untuk int, kompiler akan mengeluarkan pesan kesalahan.

Setelah ditambah L pada akhir angka pada baris 3 Program akan berhasil dikompilasi dan dieksekusi, mencetak: 1234567890123 Mengapa bisa demikian? Penambahan sufiks L (bisa huruf besar atau kecil, tetapi L disarankan untuk menghindari kebingungan dengan angka 1) pada akhir angka menandakan kepada kompiler bahwa literal angka tersebut harus diperlakukan sebagai tipe data long. Karena tipe data long mampu menyimpan angka yang sangat besar (hingga sekitar 9×10^{18}), nilai 1234567890123 valid untuk disimpan di dalamnya, sehingga program berjalan tanpa error.

Praktik 3 (percobaan tentang tipe data String dan Char)

```
cobaKalimat.java X cobalong.java pembagian.java
public class cobaKalimat {
    public static void main(String[] args){
        char coba = "HAI";
        System.out.println(coba);
    }
}

D:\FIDELIA PING\SEMESTER 3\PRAKTIKUM STRUKTUR DATA\MODUL 1>cobaKalimat.java:3: error: incompatible types: String cannot be converted to char
    char coba = "HAI";
               ^
1 error
Tool completed with exit code 1
```

```
TextPad - [D:\FIDELIA PING\SEMESTER 3\PRAKTIKUM STRUKTUR DATA\MODUL 1>coba... - File Edit Search View Tools Macros Configure Window Help
File Edit Search View Tools Macros Configure Window Help
Find Incrementally
cobakalimat.java X cobalong.java pembagian.java
public class cobakalimat {
    public static void main(String[] args){
        String coba = "HAI";
        System.out.println(coba);
    }
}

C:\Windows\system32\cmd.exe + - x
HAI
Press any key to continue . . .
```

Pembasanah : Apa yang terjadi sewaktu program di atas dieksekusi? Program akan menghasilkan kesalahan (error) kompilasi. Mengapa bisa demikian? Variabel coba dideklarasikan sebagai tipe data char. Tipe data char di Java hanya dapat menampung satu karakter tunggal, yang harus diapit oleh tanda kutip tunggal (' '). Anda mencoba mengisinya dengan tiga karakter ("HAI") yang diapit oleh tanda kutip ganda, yang menandakan sebuah String. Konflik tipe data ini menyebabkan error kompilasi.

Apa yang terjadi Ketika char pada baris 3 diganti dengan String? Program akan berhasil dikompilasi dan dieksekusi, mencetak: HAI. Mengapa bisa demikian? Mengganti tipe data variabel coba menjadi String menyelaraskan tipe data variabel dengan nilai literal yang diberikan ("HAI"). String adalah tipe data di Java yang digunakan untuk menyimpan urutan karakter (teks atau kalimat) dan diinisialisasi menggunakan tanda kutip ganda (" "). Karena tipe data variabel dan nilai sudah sesuai, program berjalan dengan sukses.

Praktik 4 (percobaan tentang lingkup variabel)

The screenshot shows a Java development environment with several windows:

- Code Editor:** Displays the following Java code with syntax highlighting and error markers:

```
public class Variabel {
    static int a;
    public static void main(String[] args) {
        int x;
        x = 10;
        a = 2;
        System.out.println("Nilai a : " + a);
        {
            int y;
            y = 5;
            System.out.println("Nilai x : " + x);
            System.out.println("Nilai a : " + a);
            {
                int z;
                z = 20;
                System.out.println("Nilai x + y + z + a :"
                    + (x + y + z + a));
            }
            System.out.println("Nilai Z : " + z);
            System.out.println("Nilai y : " + y);
        }
        System.out.println("Nilai Z : " + Z);
        System.out.println("Nilai y : " + y);
        System.out.println("Nilai x : " + x);
    }
}
```
- Tool Output:** Shows compilation errors:

```
D:\FIDELIA PING\SEMESTER 3\PRAKTIKUM STRUKTUR DATA\MODUL 1\Variabel.java:17: error: cannot find symbol
    System.out.println("Nilai Z : " + Z);
                           ^
symbol:  variable Z
location: class Variabel
D:\FIDELIA PING\SEMESTER 3\PRAKTIKUM STRUKTUR DATA\MODUL 1\Variabel.java:20: error: cannot find symbol
```
- Terminal Window:** Shows the execution of the program:

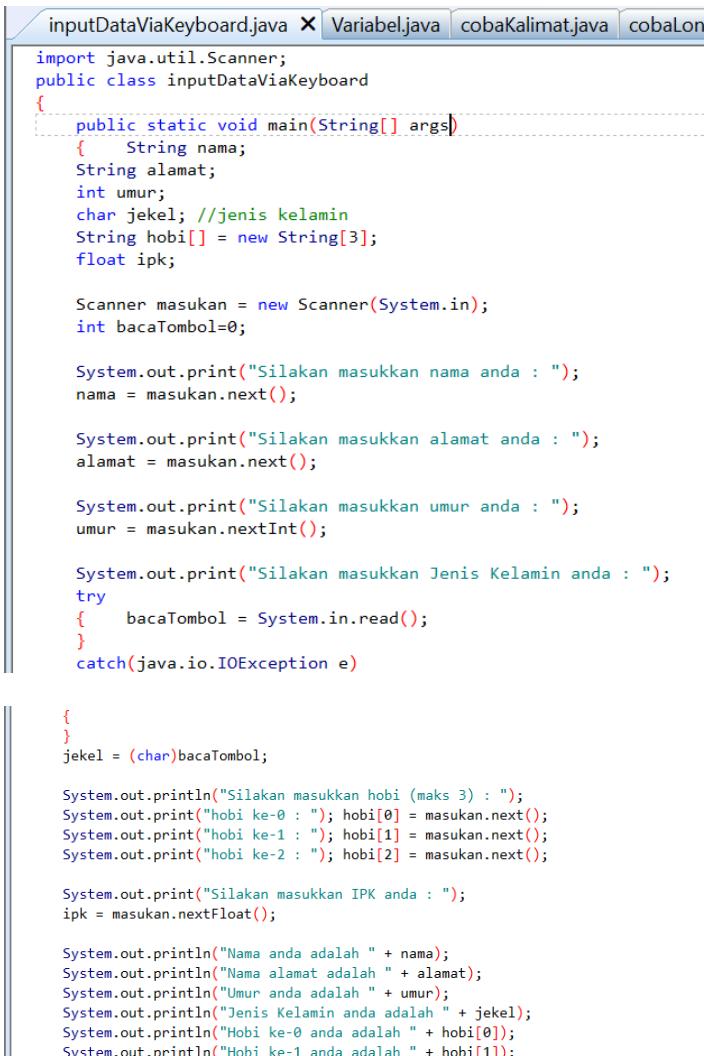
```
Nilai a : 2
Nilai x : 10
Nilai a : 2
Nilai x + y + z + a :37
Nilai y : 5
Nilai x : 10
Press any key to continue . . .
```
- TextPad Editor:** Shows the same Java code as the code editor.

Pembasanahan : Apakah yang terjadi sewaktu program tersebut dieksekusi? Program akan menghasilkan kesalahan (error) kompilasi. Mengapa bisa demikian? Kesalahan terjadi karena adanya pelanggaran lingkup variabel (variable scope): Baris 17 dan 20: Variabel z dideklarasikan di baris 12 di dalam blok kode terdalam. Lingkup (cakupan) z hanya berlaku di dalam blok tersebut (baris 12 hingga 16). Mencoba mengakses z (atau Z, karena Java case-sensitive) di baris 17 (di luar blok z) dan baris 20 (di luar blok y dan z) menyebabkan error. Baris 21: Variabel y dideklarasikan di baris 8 di dalam blok kode menengah. Lingkup y hanya berlaku di dalam blok tersebut (baris 8 hingga 19). Mencoba mengakses y di baris 21 (di luar blok y) menyebabkan error. Variabel x (dideklarasikan di baris 4) dan a (dideklarasikan statis di baris 2) dapat diakses dengan sukses karena berada di luar blok yang lebih kecil.

Setelah Penghapusan Baris 17, 20, dan 21 Program akan berhasil dikompilasi dan dieksekusi, mencetak output seperti diatas Mengapa bisa demikian? Dengan menghapus baris yang mencoba mengakses variabel di luar lingkupnya (z dan y), sisa program hanya mengakses variabel dalam batas lingkup yang diizinkan:

- a: Variabel statis, dapat diakses di mana saja.
- x: Dideklarasikan di dalam `main`, lingkupnya hingga akhir `main`.
- y: Dideklarasikan di Baris 8, diakses di Baris 18 (masih di dalam bloknya), sehingga valid.
- z: Hanya diakses di Baris 14-15 (di dalam bloknya), sehingga valid.

Praktik 5 (percobaan tentang menerima masukan keyboard)



```
import java.util.Scanner;
public class inputDataViaKeyboard
{
    public static void main(String[] args)
    {
        String nama;
        String alamat;
        int umur;
        char jekel; //jenis kelamin
        String hobi[] = new String[3];
        float ipk;

        Scanner masukan = new Scanner(System.in);
        int bacaTombol=0;

        System.out.print("Silakan masukkan nama anda : ");
        nama = masukan.next();

        System.out.print("Silakan masukkan alamat anda : ");
        alamat = masukan.next();

        System.out.print("Silakan masukkan umur anda : ");
        umur = masukan.nextInt();

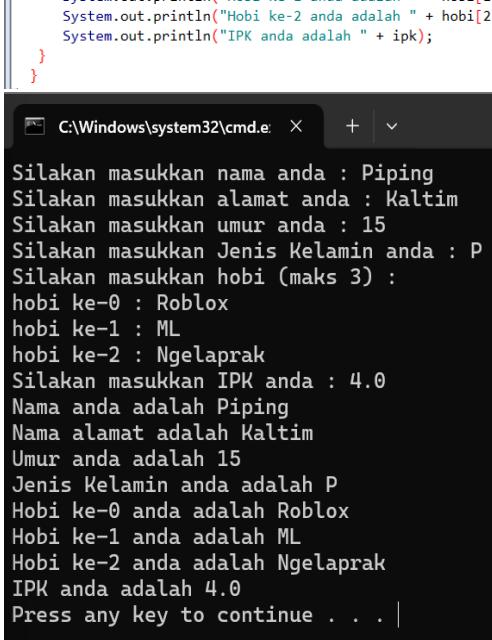
        System.out.print("Silakan masukkan Jenis Kelamin anda : ");
        try
        {
            bacaTombol = System.in.read();
        }
        catch(java.io.IOException e)

        {
            jekel = (char)bacaTombol;

            System.out.println("Silakan masukkan hobi (maks 3) : ");
            System.out.print("hobi ke-0 : "); hobi[0] = masukan.next();
            System.out.print("hobi ke-1 : "); hobi[1] = masukan.next();
            System.out.print("hobi ke-2 : "); hobi[2] = masukan.next();

            System.out.print("Silakan masukkan IPK anda : ");
            ipk = masukan.nextFloat();

            System.out.println("Nama anda adalah " + nama);
            System.out.println("Nama alamat adalah " + alamat);
            System.out.println("Umur anda adalah " + umur);
            System.out.println("Jenis Kelamin anda adalah " + jekel);
            System.out.println("Hobi ke-0 anda adalah " + hobi[0]);
            System.out.println("Hobi ke-1 anda adalah " + hobi[1]);
            System.out.println("Hobi ke-2 anda adalah " + hobi[2]);
            System.out.println("IPK anda adalah " + ipk);
        }
    }
}
```



```
C:\Windows\system32\cmd.e: Silakan masukkan nama anda : Piping
Silakan masukkan alamat anda : Kaltim
Silakan masukkan umur anda : 15
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : Roblox
hobi ke-1 : ML
hobi ke-2 : Ngelaprak
Silakan masukkan IPK anda : 4.0
Nama anda adalah Piping
Nama alamat adalah Kaltim
Umur anda adalah 15
Jenis Kelamin anda adalah P
Hobi ke-0 anda adalah Roblox
Hobi ke-1 anda adalah ML
Hobi ke-2 anda adalah Ngelaprak
IPK anda adalah 4.0
Press any key to continue . . . |
```

Pembasanah : program berhasil dijalankan dengan output nama Piping dengan Alamat Kaltim, umur 15, dengan jenis kelamin P dan hobi 3. Tidak terlihat error dalam programnya

C. PEMBAHASAN LATIHAN

- Buatlah sebuah program sederhana untuk menerima masukan bertipe string dari keyboard berupa "password". Selanjutnya program akan mencocokan password tersebut. Jika password yang dimasukkan sama dengan "AKAKOM" maka akan tampil pesan "password anda benar" tetapi jika tidak akan tampil pesan "password anda salah". T

The screenshot shows a Java code editor (TextPad) and a terminal window (cmd.exe). The code in TextPad is:

```
1 import java.util.Scanner;
2 public class Latihan {
3     public static void main(String[] args) {
4
5         Scanner input = new Scanner(System.in);
6         System.out.print("Password: ");
7         String password = input.nextLine();
8
9         if (password.equals("AKAKOM")) {
10
11             System.out.println("Password anda benar");
12         } else {
13             System.out.println("Password anda salah");
14         }
15     }
16 }
```

The terminal window shows the output of running the program:

```
 Password: AKAKOM
 Password anda benar
Press any key to continue . . .
```

Pembahasan : Ketika Program Dijalankan program akan berjalan dengan sukses dan menunggu masukan dari pengguna. Pertama, program akan mencetak prompt ke layar konsol: Password: Program kemudian berhenti (menunggu) pada baris String password = input.nextLine(); hingga pengguna mengetikkan teks dan menekan ENTER. Setelah pengguna menekan ENTER, program akan masuk ke struktur kontrol if-else dan membandingkan input pengguna (password) dengan string target "AKAKOM". Gambar menunjukkan bahwa pengguna memasukkan AKAKOM sebagai input. Input: password = "AKAKOM" Kondisi: if (password.equals("AKAKOM")) \$\rightarrow\$ if ("AKAKOM".equals("AKAKOM")) Hasil Kondisi: TRUE Output: Baris di dalam blok if dieksekusi: Password anda benar.

D. PEMBAHASAN TUGAS

- Dengan menggunakan struktur data seperti pada praktik 5 (nama, alamat, umur, jenis kelamin, hobi (3 buah), IPK), buatlah sebuah program untuk memasukkan biodata minimal untuk 5 orang mahasiswa.

```
import java.util.Scanner;

public class BiodataMahasiswa {

    public static void main(String[] args) {
        // Jumlah mahasiswa yang akan diinput
        final int JUMLAH_MAHASISWA = 5;

        // Deklarasi Array untuk menyimpan data 5 mahasiswa
        String[] nama = new String[JUMLAH_MAHASISWA];
        String[] alamat = new String[JUMLAH_MAHASISWA];
        int[] umur = new int[JUMLAH_MAHASISWA];
        char[] jekel = new char[JUMLAH_MAHASISWA];
        String[][] hobi = new String[JUMLAH_MAHASISWA][3]; // Array 2D:
        [Mahasiswa][Hobi ke-]
        float[] ipk = new float[JUMLAH_MAHASISWA];

        // Inisialisasi Scanner
        Scanner masukan = new Scanner(System.in);

        System.out.println("--- INPUT BIODATA 5 MAHASISWA ---");

        // Loop untuk memasukkan data 5 mahasiswa
        for (int i = 0; i < JUMLAH_MAHASISWA; i++) {
            System.out.println("\n--- Data Mahasiswa Ke-" + (i + 1) + " ---");

            // 1. Nama (String)
            System.out.print("Silakan masukkan Nama: ");
            nama[i] = masukan.nextLine();

            // 2. Alamat (String)
            System.out.print("Silakan masukkan Alamat: ");
            alamat[i] = masukan.nextLine();

            // 3. Umur (Integer)
            System.out.print("Silakan masukkan Umur: ");
            umur[i] = Integer.parseInt(masukan.nextLine());

            // 4. Jenis Kelamin (Char)
            System.out.print("Silakan masukkan Jenis Kelamin (L/P): ");
            jekel[i] = masukan.next().charAt(0);

            // 5. Hobi (String)
            System.out.print("Silakan masukkan Hobi (3 buah): ");
            hobi[i] = masukan.nextLine();
        }

        // Output
        System.out.println("Data berhasil dimasukkan!");
        System.out.println("Data Mahasiswa:");
        for (int i = 0; i < JUMLAH_MAHASISWA; i++) {
            System.out.println("Nama: " + nama[i]);
            System.out.println("Alamat: " + alamat[i]);
            System.out.println("Umur: " + umur[i]);
            System.out.println("Jenis Kelamin: " + jekel[i]);
            System.out.println("Hobi: " + hobi[i]);
        }
    }
}
```

```

// 2. Alamat (String)
System.out.print("Silakan masukkan Alamat: ");
alamat[i] = masukan.nextLine();

// 3. Umur (int)
System.out.print("Silakan masukkan Umur: ");
umur[i] = masukan.nextInt();
masukan.nextLine(); // Membersihkan buffer setelah nextInt()

// 4. Jenis Kelamin (char)
System.out.print("Silakan masukkan Jenis Kelamin (L/P): ");
// Mengambil karakter pertama dari input nextLine()
String inputJekel = masukan.nextLine();
if (!inputJekel.isEmpty()) {
    jekel[i] = inputJekel.charAt(0);
} else {
    jekel[i] = ' '; // Default jika input kosong
}

// 5. Hobi (3 String)
System.out.println("Silakan masukkan 3 Hobi:");
for (int j = 0; j < 3; j++) {
    System.out.print("Hobi ke-" + (j + 1) + ": ");
    hobi[i][j] = masukan.nextLine();
}

// 6. IPK (float)
System.out.print("Silakan masukkan IPK: ");
ipk[i] = masukan.nextFloat();
masukan.nextLine(); // Membersihkan buffer setelah nextFloat()
}

// Menutup Scanner
masukan.close();

// --- TAMPILAN DATA MAHASISWA ---
System.out.println("\n\n==== DATA BIODATA 5 MAHASISWA YANG TERSIMPAN
====");
for (int i = 0; i < JUMLAH_MAHASISWA; i++) {
    System.out.println("\n--- Mahasiswa Ke-" + (i + 1) + " ---");
    System.out.println("Nama: " + nama[i]);
    System.out.println("Alamat: " + alamat[i]);
    System.out.println("Umur: " + umur[i] + " tahun");
    System.out.println("Jenis Kelamin: " + jekel[i]);
    System.out.println("IPK: " + ipk[i]);
    System.out.println("Hobi:");
    System.out.println("  1. " + hobi[i][0]);
    System.out.println("  2. " + hobi[i][1]);
    System.out.println("  3. " + hobi[i][2]);
}
System.out.println("=====");
}
}

```

Output

```

--- INPUT BIODATA 5 MAHASISWA ---

--- Data Mahasiswa Ke-1 ---
Silakan masukkan Nama: piping
Silakan masukkan Alamat: kaltim
Silakan masukkan Umur: 19
Silakan masukkan Jenis Kelamin (L/P): P
Silakan masukkan 3 Hobi:
Hobi ke-1: makan
Hobi ke-2: tidur
Hobi ke-3: rapat
Silakan masukkan IPK: 4.0

--- Data Mahasiswa Ke-2 ---
Silakan masukkan Nama: pipi
Silakan masukkan Alamat: bengkulu
Silakan masukkan Umur: 20
Silakan masukkan Jenis Kelamin (L/P): P
Silakan masukkan 3 Hobi:
Hobi ke-1: lari
Hobi ke-2: makan
Hobi ke-3: mandi
Silakan masukkan IPK: 4.0

--- Data Mahasiswa Ke-3 ---
Silakan masukkan Nama: ing
Silakan masukkan Alamat: kaltim
Silakan masukkan Umur: 40
Silakan masukkan Jenis Kelamin (L/P): P
Silakan masukkan 3 Hobi:
Hobi ke-1: makan
Hobi ke-2: tidur
Hobi ke-3: roblox
Silakan masukkan IPK: 4.0

```

```
== DATA BIODATA 5 MAHASISWA YANG TERSIMPAN ==
```

```
--- Mahasiswa Ke-1 ---
```

```
Nama: piping
Alamat: kaltim
Umur: 19 tahun
Jenis Kelamin: P
IPK: 4.0
Hobi:
1. makan
2. tidur
3. rapat
```

```
--- Mahasiswa Ke-2 ---
```

```
Nama: pipi
Alamat: bengkulu
Umur: 20 tahun
Jenis Kelamin: P
IPK: 4.0
Hobi:
1. lari
2. makan
3. mandi
```

```
--- Mahasiswa Ke-3 ---
```

```
Nama: ing
Alamat: kaltim
Umur: 40 tahun
Jenis Kelamin: P
IPK: 4.0
Hobi:
1. makan
2. tidur
3. roblox
```

```
--- Data Mahasiswa Ke-4 ---
```

```
Silakan masukkan Nama: Ilham
Silakan masukkan Alamat: lombok
Silakan masukkan Umur: 20
Silakan masukkan Jenis Kelamin (L/P): L
Silakan masukkan 3 Hobi:
Hobi ke-1: lari
Hobi ke-2: ml
Hobi ke-3: tidur
Silakan masukkan IPK: 4.0
```

```
--- Data Mahasiswa Ke-5 ---
```

```
Silakan masukkan Nama: lany
Silakan masukkan Alamat: jakarta
Silakan masukkan Umur: 23
Silakan masukkan Jenis Kelamin (L/P): L
Silakan masukkan 3 Hobi:
Hobi ke-1: nyanyi
Hobi ke-2: karaoke
Hobi ke-3: nulis
Silakan masukkan IPK: 4.0
```

Pembahasan : Program ini berhasil meminta dan menerima semua data yang diperlukan untuk 5 entri mahasiswa. Data ini sekarang tersimpan di dalam program.

E. KESIMPULAN

Dari semua praktik yang telah dilakukan pemilihan tipe data di Java sangat krusial. Praktik pada tipe data *integer* (int), *long* (long), *character* (char), dan *String* menunjukkan bahwa setiap tipe memiliki batasan dan aturan penulisan yang spesifik; misalnya, pembagian dua int akan menghilangkan desimal, literal besar memerlukan sufiks L, dan perbandingan konten *String* harus menggunakan metode .equals(), bukan operator ==. Selain itu, praktik lingkup variabel (scope) menegaskan bahwa variabel hanya dapat diakses dalam blok kode tempatnya dideklarasikan.