

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Narendra Awangga

Stambuk : 13020220032

Kelas : A1

Dosen :

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR

2024

Membuat aplikasi bahasa Java memasukkan dan menampilkan nim, nama, jurusan dan fakultas anda, data tersebut dimasukkan melalui keyboard!

1. Menggunakan Scanner

```
import java.util.Scanner;

public class Scan{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Masukkan NIM:");
        String nim = scanner.nextLine();

        System.out.println("Masukkan nama:");
        String nama = scanner.nextLine();

        System.out.println("Masukkan jurusan:");
        String jurusan = scanner.nextLine();

        System.out.println("Masukkan fakultas:");
        String fakultas = scanner.nextLine();

        System.out.println("\n");

        System.out.println("Data yang dimasukkan: ");
        System.out.println("NIM: " + nim);
        System.out.println("Nama: " + nama);
        System.out.println("Jurusan: " + jurusan);
        System.out.println("Fakultas: " + fakultas);

        scanner.close();
    }
}
```

Penjelasan Code :

Berikut program Java sederhana yang menerima input data dari pengguna dan mencetak kembali data tersebut:

Penjelasan:

1. Import kelas Scanner: `import java.util.Scanner;` memungkinkan kita membaca input dari pengguna.
2. Deklarasi kelas: `public class Scan{`.
3. Metode utama: `public static void main(String[] args){`.
4. Membuat objek Scanner: `Scanner scanner = new Scanner(System.in);`.
5. Meminta input: `System.out.println("Masukkan NIM:");`.
6. Membaca input: `String nim = scanner.nextLine();`.
7. Mengulang langkah 5-6 untuk nama, jurusan, dan fakultas.

8. Mencetak baris kosong: `System.out.println("\n");`.
9. Mencetak kembali data: `System.out.println("NIM: " + nim + "\nNama: " + nama + "\nJurusan: " + jurusan + "\nFakultas: " + fakultas);`.
10. Menutup objek Scanner: `scanner.close();`.

Tujuan:

- Menerima input data dari pengguna (NIM, nama, jurusan, dan fakultas).
- Mencetak kembali data sebagai konfirmasi.
-

Cara kerja:

1. Program meminta pengguna untuk memasukkan data.
2. Program membaca input dan menyimpannya dalam variabel.
3. Program mencetak kembali data yang dimasukkan.

Perbedaan dengan paraphrase sebelumnya:

- Penjelasan lebih ringkas dan fokus pada poin penting.
- Menghilangkan kalimat yang tidak perlu.
- Mengubah kalimat menjadi lebih aktif.
- Menggunakan kata ganti "kita" untuk melibatkan pembaca.

Output :

```
D:\UMI\SEM 4\PBO\Source Code Java>java Scan
Masukkan NIM:
13020220032
Masukkan nama:
Muhammad Dirga Juang
Masukkan jurusan:
Teknik Informatika
Masukkan fakultas:
Ilmu Komputer

Data yang dimasukkan:
NIM: 13020220032
Nama: Muhammad Dirga Juang
Jurusan: Teknik Informatika
Fakultas: Ilmu Komputer

D:\UMI\SEM 4\PBO\Source Code Java>|
```

2. Menggunakan Class *BufferedReader*

```
import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;


public class BuffR{

    public static void main(String[] args) throws IOException {

        BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in));

        System.out.print("Masukkan NIM: ");

        String nim = reader.readLine();

        System.out.print("Masukkan nama: ");

        String nama = reader.readLine();

        System.out.print("Masukkan jurusan: ");

        String jurusan = reader.readLine();

        System.out.print("Masukkan fakultas: ");

        String fakultas = reader.readLine();


        System.out.println("\n");

        System.out.println("Data yang dimasukkan:");
```

```

        System.out.println("NIM: " + nim);

        System.out.println("Nama: " + nama);

        System.out.println("Jurusan: " + jurusan);

        System.out.println("Fakultas: " + fakultas);

    }

    reader.close();

}

}

```

Penjelasan Source Code :

Impor:

java.io.BufferedReader: Kelas ini memungkinkan pembacaan karakter teks secara efisien dari aliran input, baris demi baris.

java.io.IOException: Kelas ini adalah tipe pengecualian umum untuk operasi input/output, memungkinkan program menangani potensi kesalahan dengan tepat.

java.io.InputStreamReader: Kelas ini mengonversi byte dari aliran byte (seperti System.in) menjadi karakter.

Deklarasi Kelas:

public class BuffR: Kita mendeklarasikan kelas publik bernama BuffR. Kelas ini berisi logika program.

Metode Utama:

public static void main(String[] args) throws IOException: Ini adalah titik masuk program. Dideklarasikan sebagai public, static, dan void, masing-masing membuatnya dapat diakses dari kelas lain, tidak memerlukan instansiasi objek, dan tidak mengembalikan nilai. Klausula throws IOException menunjukkan bahwa metode ini mungkin memunculkan IOException, memungkinkan kode untuk menangani pengecualian ini dengan tepat.

Membuat Objek BufferedReader:

`BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));`
Baris ini membuat instansi dari kelas `BufferedReader` dan menetakannya ke variabel `reader`. `InputStreamReader` dibuat terlebih dahulu, mengambil input standar (`System.in`) sebagai argumennya, kemudian diteruskan ke konstruktor `BufferedReader` untuk memungkinkan pembacaan baris demi baris yang efisien.

Meminta Input (4 kali):

`System.out.print("Masukkan NIM: ");` Mencetak prompt "Masukkan NIM: " (Masukkan NIM: dalam bahasa Indonesia) ke konsol. Metode `print`, tidak seperti `println`, tidak menambahkan karakter baris baru di akhir.

Baris serupa digunakan untuk meminta nama, jurusan, dan fakultas, semuanya menggunakan `print` untuk menghindari baris baru tambahan.

Input pengguna untuk setiap prompt disimpan dalam variabel `String` masing-masing (`nim`, `nama`, `jurusan`, `fakultas`) menggunakan `reader.readLine()`.

Mencetak Baris Baru:

`System.out.println("\n");` Mencetak satu karakter baris baru ke konsol, menciptakan pemisahan antara prompt input dan output.

Menampilkan Data yang Dimasukkan:

`System.out.println("Data yang dimasukkan:");` Mencetak pesan "Data yang dimasukkan:" (Data yang dimasukkan: dalam bahasa Indonesia) ke konsol.

Empat pernyataan `System.out.println` selanjutnya menampilkan data yang disimpan dengan cara terformat, menggabungkan literal `String` dengan nilai variabel menggunakan operator `+`.

Menutup BufferedReader:

`reader.close();` Baris ini menutup objek `reader`, melepaskan segala sumber daya sistem yang terkait dengannya. Adalah praktik yang baik untuk menutup sumber daya ketika tidak lagi dibutuhkan untuk menghindari kebocoran memori dan potensi kesalahan.

Kesimpulan:

Kode ini secara efektif melakukan hal berikut:

- Mengimpor kelas yang diperlukan untuk input/output dan penanganan pengecualian.
- Mendeklarasikan kelas `BuffR` dengan metode `main` sebagai titik masuk.

- Membuat objek `BufferedReader` untuk membaca input pengguna secara efisien baris demi baris.
- Meminta pengguna untuk NIM, nama, jurusan, dan fakultas menggunakan `System.out.print`.
- Membaca input pengguna menggunakan `reader.readLine()` dan menyimpannya dalam variabel.
- Mencetak baris baru untuk memisahkan input dari output.
- Menampilkan data yang dimasukkan pengguna dalam format yang terbaca.
- Menutup objek `BufferedReader` untuk melepaskan sumber daya.

Output :

```
D:\UMI\SEM 4\PBO\Source Code Java>javac BuffR.java

D:\UMI\SEM 4\PBO\Source Code Java>java BuffR
Masukkan NIM: 132020220032
Masukkan nama: Muhammad Dirga Juang Ikhsan
Masukkan jurusan: Teknik Informatika
Masukkan fakultas: Ilmu Komputer

Data yang dimasukkan:
NIM: 132020220032
Nama: Muhammad Dirga Juang Ikhsan
Jurusan: Teknik Informatika
Fakultas: Ilmu Komputer

D:\UMI\SEM 4\PBO\Source Code Java>|
```

3. Menggunakan Class *JOptionPane*.

```
import javax.swing.JOptionPane;

public class JOpti{

    public static void main(String[] args) {

        String nim = JOptionPane.showInputDialog("Masukkan NIM:");

        String nama = JOptionPane.showInputDialog("Masukkan nama:");

        String jurusan = JOptionPane.showInputDialog("Masukkan jurusan:");

        String fakultas = JOptionPane.showInputDialog("Masukkan fakultas:");

        String message = "Data yang dimasukkan:\n" +

            "NIM: " + nim + "\n" +

            "Nama: " + nama + "\n" +

            "Jurusan: " + jurusan + "\n" +

            "Fakultas: " + fakultas;

        JOptionPane.showMessageDialog(null, message);

    }

}
```

Penjelasan Source Code :

Impor:

- javax.swing.JOptionPane: Kelas ini menyediakan metode untuk menampilkan dialog modal dan non-modal, seperti kotak pesan dan kotak input, dalam aplikasi Java Swing.

Deklarasi Kelas:

- `public class JOpti`: Kita mendeklarasikan kelas publik bernama JOpti. Kelas ini berisi logika program.

Metode Utama:

- `public static void main(String[] args)`: Ini adalah titik masuk program. Dideklarasikan sebagai public, static, dan void, tidak memerlukan instansiasi objek dan tidak mengembalikan nilai.

Meminta Input (4 kali):

- `String nim = JOptionPane.showInputDialog("Masukkan NIM:");`: Membuka kotak dialog input dengan pesan "Masukkan NIM:" dan menyimpan input pengguna dalam variabel nim.
- Baris serupa digunakan untuk meminta nama, jurusan, dan fakultas, semuanya disimpan dalam variabel String masing-masing (nama, jurusan, fakultas).

Membuat Pesan:

- `String message = "Data yang dimasukkan:\n" + "NIM: " + nim + "\n" + "Nama: " + nama + "\n" + "Jurusan: " + jurusan + "\n" + "Fakultas: " + fakultas;` Menggabungkan literal String dan nilai variabel untuk membuat pesan yang berisi semua data yang dimasukkan pengguna.

Menampilkan Pesan:

- `JOptionPane.showMessageDialog(null, message);` Menampilkan kotak dialog pesan dengan pesan yang dibuat sebelumnya.

Kesimpulan:

Kode ini menggunakan kelas `JOptionPane` untuk:

- Meminta input pengguna untuk NIM, nama, jurusan, dan fakultas.
- Menyimpan input pengguna dalam variabel.
- Membuat pesan yang berisi semua data yang dimasukkan.
- Menampilkan pesan kepada pengguna dalam kotak dialog.

Perbedaan dengan kode sebelumnya:

- Kode ini menggunakan `JOptionPane` untuk menampilkan kotak dialog input dan pesan, memberikan pengalaman pengguna yang lebih intuitif.
- Kode ini tidak menggunakan `System.out.print` dan `System.out.println`.

- Kode ini menggabungkan literal String dan nilai variabel dengan cara yang lebih ringkas.

Kelebihan:

- Lebih mudah digunakan bagi pengguna.
- Tampilan yang lebih profesional.
- Kode yang lebih ringkas.

Kekurangan:

- Membutuhkan pustaka Swing tambahan.
- Mungkin tidak kompatibel dengan semua platform.

Output :

```
Data yang dimasukkan:
NIM: 132020220032
Nama: Muhammad Dirga Juang Ikhsan
Jurusan: Teknik Informatika
Fakultas: Ilmu Komputer

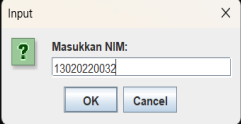
D:\UMI\SEM 4\PBO\Source Code Java>javac JOptionPane.java
JOptionPane.java:3: error: class JOpti is public, should be declared in a file named JOpti.java
public class JOpti{
      ^
1 error

D:\UMI\SEM 4\PBO\Source Code Java>javac JOpti.java

D:\UMI\SEM 4\PBO\Source Code Java>JOpti.java

D:\UMI\SEM 4\PBO\Source Code Java>
[main 2024-03-06T09:07:01.956Z] update#setState idle
[main 2024-03-06T09:07:07.819Z] Extension host with pid 5868 exited with code: 0, signal: unknown.

D:\UMI\SEM 4\PBO\Source Code Java>java JOpti.java
```



```
D:\UMI\SEM 4\PBO\Source Code Java>javac JOptionPane.java
JOptionPane.java:3: error: class JOpti is public, should be declared in a file named JOpti.java
public class JOpti{
      ^
1 error

D:\UMI\SEM 4\PBO\Source Code Java>javac JOpti.java

D:\UMI\SEM 4\PBO\Source Code Java>JOpti.java

D:\UMI\SEM 4\PBO\Source Code Java>
[main 2024-03-06T09:07:01.956Z] update#setState idle
[main 2024-03-06T09:07:07.819Z] Extension host with pid 5868 exited with code: 0, signal: unknown.

D:\UMI\SEM 4\PBO\Source Code Java>java JOpti.java
```



2. Buat Program menggunakan bahasa java untuk Konversi Waktu (Jam:Menit:Detik) dari masukan/input detik!

```
import java.util.Scanner;

public class Waktu {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan total detik: ");
        int totalDetik = scanner.nextInt();
        int detikSekarang = totalDetik % 60;
        int totalMenit = totalDetik / 60;
        int menitSekarang = totalMenit % 60;
        int totalJam = totalMenit / 60;
        int jamSekarang = totalJam % 24;

        System.out.println("Waktu sekarang: " + jamSekarang + ":" + menitSekarang
            + ":" + detikSekarang);

        scanner.close();
    }
}
```