Tugas 2 PBO Makassar, 27 Februari 2024

**TUGAS PEMROGRAMAN**

**BEROIENTASI OBJEK**



Nama : Muh. Reza

Stambuk : 13020220070

Dosen : Mardiyyah Hasnawi, S.Kom.,M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MUSLIM INDONESIA**

**MAKASSAR**

**2024**

1. **Metode Scanner:**

1. `**import java.util.Scanner;`:**

- Ini adalah pernyataan untuk mengimpor kelas `Scanner` dari paket `java.util`. Ini memungkinkan kita untuk menggunakan kelas Scanner untuk membaca input dari pengguna.

2. `**public class Scan {`:**

- Ini adalah deklarasi kelas utama. Kelas ini diberi nama "Scan" dan dideklarasikan sebagai `public`, yang berarti dapat diakses dari kelas lain.

3. `**public static void main(String[] args) {`:**

- Ini adalah deklarasi metode utama dalam kelas. Setiap program Java harus memiliki metode `main`, yang merupakan titik awal eksekusi program.

4. `**Scanner scanner = new Scanner(System.in);`:**

- Ini adalah pembuatan objek Scanner baru dengan parameter `System.in`. Objek Scanner ini digunakan untuk membaca input dari pengguna melalui konsol.

5. `**String nama = scanner.nextLine();`:**

- Ini adalah pernyataan untuk membaca baris teks dari input pengguna dan menyimpannya ke dalam variabel `nama`.

6. `**String nim = scanner.nextLine();`:**

- Ini adalah pernyataan yang mirip dengan sebelumnya, namun kali ini digunakan untuk membaca NIM (Nomor Induk Mahasiswa) dari input pengguna.

7. `**int tahunMasuk = scanner.nextInt();`:**

- Ini adalah pernyataan untuk membaca integer dari input pengguna dan menyimpannya ke dalam variabel `tahunMasuk`.

8. `**System.out.println("\nData Mahasiswa:");`:**

- Ini adalah pernyataan yang mencetak teks "Data Mahasiswa:" ke konsol.

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

- Ini adalah pernyataan yang mencetak nama mahasiswa yang telah dimasukkan oleh pengguna.

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

- Ini adalah pernyataan yang mencetak NIM mahasiswa yang telah dimasukkan oleh pengguna.

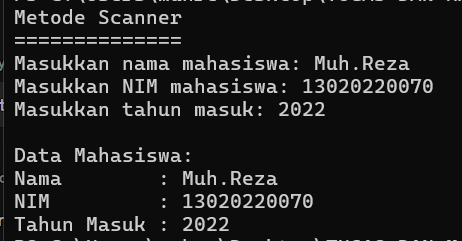
11. **`System.out.println("Tahun Masuk : " + tahunMasuk);`:**

- Ini adalah pernyataan yang mencetak tahun masuk mahasiswa yang telah dimasukkan oleh pengguna.

12. `**scanner.close();`:**

- Ini adalah pernyataan yang menutup objek Scanner setelah selesai digunakan untuk mencegah kebocoran sumber daya.

Output:



1. **Metode BufferedReader:**

1. **`import java.io.BufferedReader;`:**

- Ini adalah pernyataan untuk mengimpor kelas `BufferedReader` dari paket `java.io`. Kelas ini digunakan untuk membaca input karakter dari suatu sumber.

2. **`import java.io.IOException;`:**

- Ini adalah pernyataan untuk mengimpor kelas `IOException` dari paket `java.io`. Kelas ini merupakan kelas pengecualian yang dilemparkan saat terjadi kesalahan I/O.

3**. `import java.io.InputStreamReader;`:**

- Ini adalah pernyataan untuk mengimpor kelas `InputStreamReader` dari paket `java.io`. Kelas ini digunakan untuk membaca input byte dari input stream dan menerjemahkannya menjadi karakter menggunakan encoding yang ditentukan atau default encoding jika tidak ada yang ditentukan.

4. **`public class Reader {`:**

- Ini adalah deklarasi kelas utama. Kelas ini diberi nama "Reader" dan dideklarasikan sebagai `public`, yang berarti dapat diakses dari kelas lain.

5. **`public static void main(String[] args) throws IOException {`:**

- Ini adalah deklarasi metode utama dalam kelas. Setiap program Java harus memiliki metode `main`, yang merupakan titik awal eksekusi program. Metode ini dideklarasikan melempar IOException, yang artinya jika terjadi kesalahan I/O, penanganan kesalahan harus dilakukan di tempat lain.

6**. `BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));`:**

- Ini adalah pembuatan objek BufferedReader baru dengan menggunakan InputStreamReader sebagai input stream dan System.in sebagai sumber inputnya. Objek BufferedReader ini digunakan untuk membaca input dari pengguna melalui konsol.

7. `**System.out.println("Metode BufferedReader \n");`:**

- Ini adalah pernyataan yang mencetak teks "Metode BufferedReader" diikuti dengan baris baru ke konsol.

8. `**System.out.println("===================== \n");`:**

- Ini adalah pernyataan yang mencetak baris garis putus diikuti dengan baris baru ke konsol.

9. `**String nama = reader.readLine();`:**

- Ini adalah pernyataan yang membaca baris teks dari input pengguna dan menyimpannya ke dalam variabel `nama`.

10. `**String nim = reader.readLine();`:**

- Ini adalah pernyataan yang mirip dengan sebelumnya, namun kali ini digunakan untuk membaca NIM (Nomor Induk Mahasiswa) dari input pengguna.

11. `**int tahunMasuk = Integer.parseInt(reader.readLine());`:**

- Ini adalah pernyataan yang membaca baris teks dari input pengguna, kemudian mengonversinya menjadi tipe data integer menggunakan `Integer.parseInt()`, dan menyimpannya ke dalam variabel `tahunMasuk`.

12. **`System.out.println("\nData Mahasiswa:");`:**

- Ini adalah pernyataan yang mencetak teks "Data Mahasiswa:" diikuti dengan baris baru ke konsol.

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

- Ini adalah pernyataan yang mencetak nama mahasiswa yang telah dimasukkan oleh pengguna.

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

- Ini adalah pernyataan yang mencetak NIM mahasiswa yang telah dimasukkan oleh pengguna.

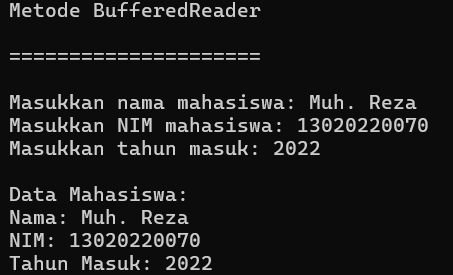
15. `**System.out.println("Tahun Masuk: " + tahunMasuk);`:**

- Ini adalah pernyataan yang mencetak tahun masuk mahasiswa yang telah dimasukkan oleh pengguna.

16. `**reader.close();`:**

- Ini adalah pernyataan yang menutup objek BufferedReader setelah selesai digunakan untuk mencegah kebocoran sumber daya.

Output:



1. **Metode Console**

1. **`import java.io.Console;`:**

- Mengimpor kelas `Console` dari paket `java.io` yang memungkinkan kita untuk membaca input dari pengguna dan menampilkan output ke konsol.

2. `**Console console = System.console();`:**

- Membuat objek `Console` dengan menggunakan metode `System.console()`. Jika `System.console()` mengembalikan `null`, berarti console tidak tersedia.

3**. `if (console == null) { ... }`:**

- Menggunakan pengecekan kondisi untuk memastikan bahwa console tersedia sebelum melanjutkan. Jika console tidak tersedia, program akan mencetak pesan kesalahan dan keluar dari program.

4. **`String nama = console.readLine("Masukkan nama mahasiswa: ");`:**

- Membaca baris teks dari pengguna menggunakan metode `readLine()` dari objek `Console`, dan menyimpannya ke dalam variabel `nama`.

5. **`String nim = console.readLine("Masukkan NIM mahasiswa: ");`:**

- Membaca NIM dari pengguna menggunakan metode `readLine()` dari objek `Console`, dan menyimpannya ke dalam variabel `nim`.

6. `**int tahunMasuk = Integer.parseInt(console.readLine("Masukkan tahun masuk: "));`:**

- Membaca tahun masuk dari pengguna menggunakan metode `readLine()` dari objek `Console`, dan mengonversinya menjadi tipe data integer menggunakan `Integer.parseInt()`, kemudian menyimpannya ke dalam variabel `tahunMasuk`.

7. `**System.out.println("\nData Mahasiswa:");`:**

- Mencetak teks "Data Mahasiswa:" diikuti dengan baris baru ke konsol.

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

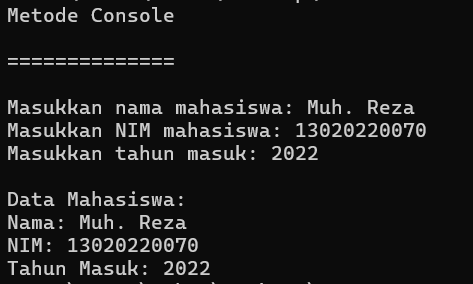
- Mencetak nama mahasiswa yang dimasukkan oleh pengguna.

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

- Mencetak NIM mahasiswa yang dimasukkan oleh pengguna.

10. `**System.out.println("Tahun Masuk: " + tahunMasuk);`:**

- Mencetak tahun masuk yang dimasukkan oleh pengguna.

Output:

1. **Studi Kasus 1:**

1**. `import java.util.Scanner;`:**

- Ini adalah pernyataan untuk mengimpor kelas `Scanner` dari paket `java.util`. Kelas ini digunakan untuk membaca input dari pengguna melalui konsol.

2. **`public class StudiKasus {`:**

- Ini adalah deklarasi kelas utama. Kelas ini diberi nama "StudiKasus" dan merupakan tempat program utama berjalan.

3. **`public static void main(String[] args) {`:**

- Ini adalah deklarasi metode utama dalam kelas. Metode ini merupakan titik awal eksekusi program.

4. `**Scanner scanner = new Scanner(System.in);`:**

- Ini adalah pembuatan objek `Scanner` baru dengan menggunakan `System.in` sebagai argumen. Objek `Scanner` ini digunakan untuk membaca input dari pengguna melalui konsol.

5. **`System.out.print("Masukkan total detik: ");`:**

- Ini adalah pernyataan yang mencetak teks "Masukkan total detik: " ke konsol tanpa membuat baris baru.

6. `**long totalDetik = scanner.nextLong();`:**

- Ini adalah pernyataan yang menggunakan objek `Scanner` untuk membaca input berikutnya dari pengguna sebagai bilangan bulat long dan menyimpannya dalam variabel `totalDetik`.

7. `**long detikSekarang = totalDetik % 60;`:**

- Ini adalah pernyataan yang menghitung detik saat ini dengan mengambil sisa pembagian dari `totalDetik` dengan 60.

8. `**long totalMenit = totalDetik / 60;`:**

- Ini adalah pernyataan yang menghitung total menit dengan membagi `totalDetik` dengan 60.

9. **`long menitSekarang = totalMenit % 60;`:**

- Ini adalah pernyataan yang menghitung menit saat ini dengan mengambil sisa pembagian dari `totalMenit` dengan 60.

10. `**long totalJam = totalMenit / 60;`:**

- Ini adalah pernyataan yang menghitung total jam dengan membagi `totalMenit` dengan 60.

11. **`long jamSekarang = totalJam % 24;`:**

- Ini adalah pernyataan yang menghitung jam saat ini dengan mengambil sisa pembagian dari `totalJam` dengan 24.

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

- Ini adalah pernyataan yang mencetak waktu saat ini (jam, menit, detik) ke konsol. Waktu ini ditampilkan dalam format "jam:menit:detik".

Output:

