

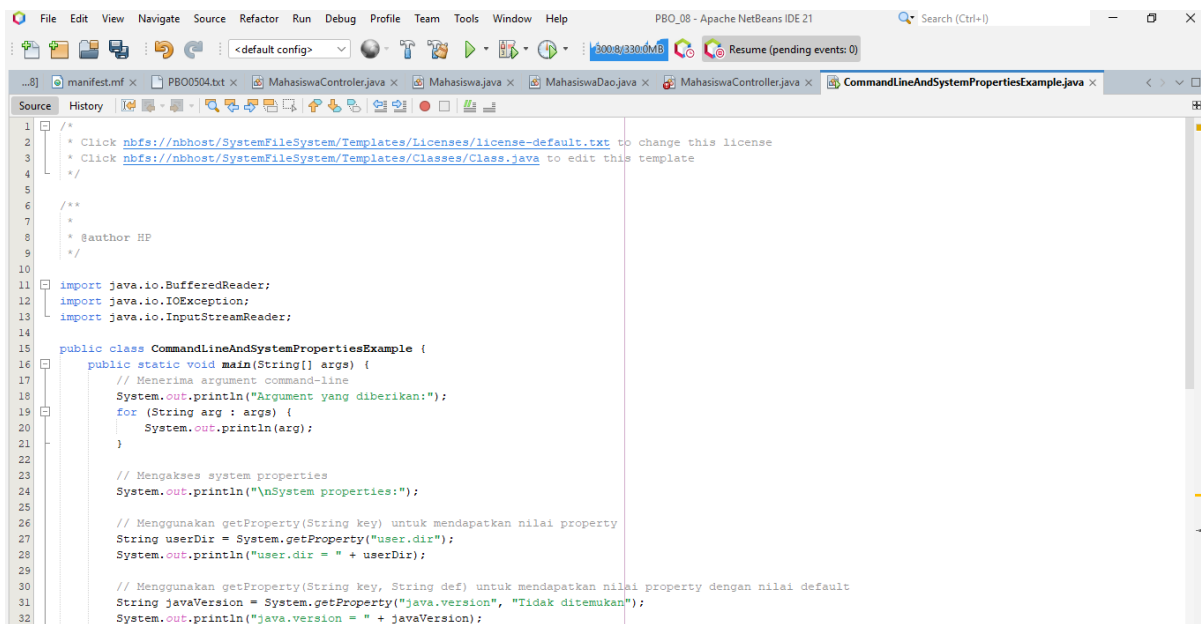
Nama : Desvita Putri

Nim : 2301092008

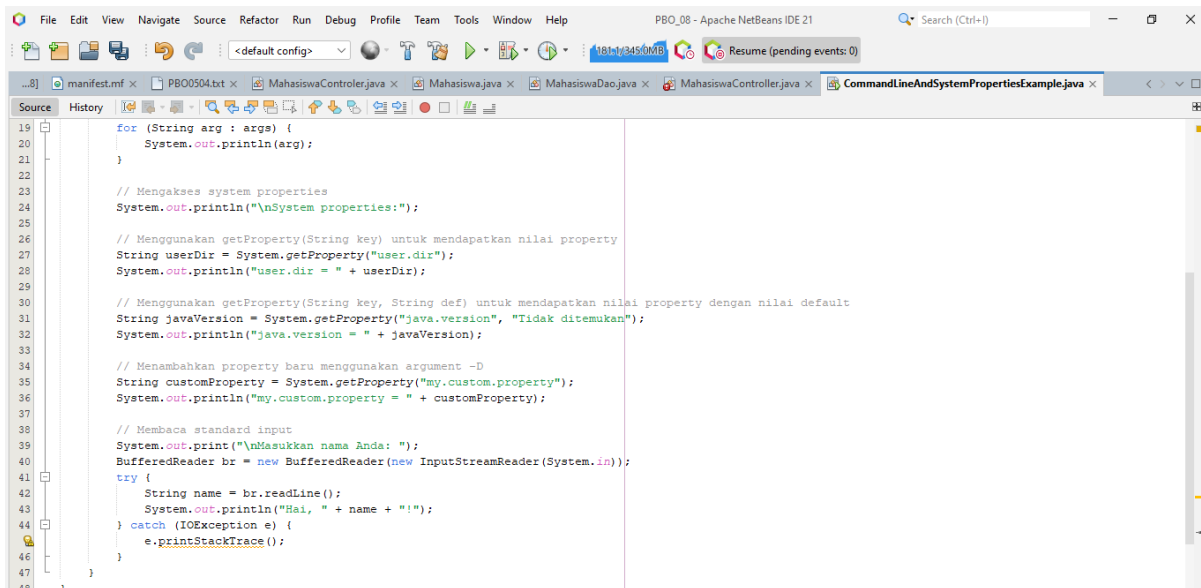
Kelas : MI IC

Tugas : Teori PBO

1. Buatlah sebuah program yang memuat dua String sebagai argument, sumber dan nama file tujuan. Kemudian, baca file sumber dan tuliskan isi dari file tersebut terhadap file tujuan, seluruh spasi yang ada (' ') diubah menjadi underscore ('_ ').



```
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6   /**
7    * @author HP
8    */
9
10
11  import java.io.BufferedReader;
12  import java.io.IOException;
13  import java.io.InputStreamReader;
14
15  public class CommandLineAndSystemPropertiesExample {
16      public static void main(String[] args) {
17          // Menerima argument command-line
18          System.out.println("Argument yang diberikan:");
19          for (String arg : args) {
20              System.out.println(arg);
21          }
22
23          // Mengakses system properties
24          System.out.println("\nSystem properties:");
25
26          // Menggunakan getProperty(String key) untuk mendapatkan nilai property
27          String userDir = System.getProperty("user.dir");
28          System.out.println("user.dir = " + userDir);
29
30          // Menggunakan getProperty(String key, String def) untuk mendapatkan nilai property dengan nilai default
31          String javaVersion = System.getProperty("java.version", "Tidak ditemukan");
32          System.out.println("java.version = " + javaVersion);
```



```
19      for (String arg : args) {
20          System.out.println(arg);
21      }
22
23      // Mengakses system properties
24      System.out.println("\nSystem properties:");
25
26      // Menggunakan getProperty(String key) untuk mendapatkan nilai property
27      String userDir = System.getProperty("user.dir");
28      System.out.println("user.dir = " + userDir);
29
30      // Menggunakan getProperty(String key, String def) untuk mendapatkan nilai property dengan nilai default
31      String javaVersion = System.getProperty("java.version", "Tidak ditemukan");
32      System.out.println("java.version = " + javaVersion);
33
34      // Menambahkan property baru menggunakan argument -D
35      String customProperty = System.getProperty("my.custom.property");
36      System.out.println("my.custom.property = " + customProperty);
37
38      // Membaca standard input
39      System.out.print("\nMasukkan nama Anda: ");
40      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
41      try {
42          String name = br.readLine();
43          System.out.println("Hai, " + name + "!");
44      } catch (IOException e) {
45          e.printStackTrace();
46      }
47  }
48  }
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class CommandLineAndSystemPropertiesExample {
    public static void main(String[] args) {
        // Menerima argument command-line
        System.out.println("Argument yang diberikan:");
        for (String arg : args) {
            System.out.println(arg);
        }

        // Mengakses system properties
        System.out.println("\nSystem properties:");

        // Menggunakan getProperty(String key) untuk mendapatkan nilai property
        String userDir = System.getProperty("user.dir");
        System.out.println("user.dir = " + userDir);

        // Menggunakan getProperty(String key, String def) untuk mendapatkan nilai property dengan nilai default
        String javaVersion = System.getProperty("java.version", "Tidak ditemukan");
        System.out.println("java.version = " + javaVersion);

        // Menambahkan property baru menggunakan argument -D
        String customProperty = System.getProperty("my.custom.property");
        System.out.println("my.custom.property = " + customProperty);

        // Membaca standard input
        System.out.print("\nMasukkan nama Anda: ");
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        try {
            String name = br.readLine();
            System.out.println("Hai, " + name + "!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

1.Menerima argument command-line:

- Program Java dapat menerima argument dari command-line saat dijalankan.
- Argument-argument tersebut disimpan dalam array String args yang main().
- Kita dapat mengakses nilai-nilai argument tersebut melalui indeks array args[].
- Pada contoh, kita m

2.Mengakses system properties:

- System properties adalah pemetaan ant
- Kita dapat mengakses system properties menggunakan method getProperty() dari class System.
- Ada dua bentuk method getProperty():
 - getProperty(String key): Mengembalikan nilai property jika ada, atau null jika tidak ditemukan.
 - getProperty(String key, String def): Mengembalikan nilai property jika ada, atau nilai default def jika tidak ditemukan.
- Pada contoh, kita mengakses dua system properties:
 - user.dir: Mengembalikan direktori kerja saat ini.
 - java.version: Mengembalikan versi Java yang digunakan, atau "Tidak ditemukan" jika tidak tersedia.
- Kita juga menunjukkan cara menambahkan property baru menggunakan argument -D saat menjalankan program.
 - Property my.custom.property ditambahkan saat menjalankan program, dan kita mengakses nilainya.

3.Membaca standard input:

- Selain menerima input dari command-line, kita juga dapat membaca input dari user saat program berjalan.
- Untuk ini, kita menggunakan byte stream System.in dan character stream BufferedReader.
- Kita membuat instance BufferedReader dengan membungkus InputStreamReader(System.in).
- Memanggil br.readLine() untuk membaca satu baris input dari user.
- Pada contoh, kita meminta user memasukkan namanya, lalu mencetak salam.