

RUNNING CODE EXAMPLE: Interface Segregation Principle (ISP)

Nama : Tangguh Chairunnisa

NIM : 21120122140103

Kelas : RPLBK C

Group : D

A. Problem

a. Code:

```
// Pelanggaran Interface Segregation Principle

interface PeralatanRumahTangga {
    void hidupkan();
    void matikan();
    void panaskan();
    void bekukan();
    void haluskan();
}

// Blender dipaksa untuk mengimplementasikan metode yang tidak perlu
class Blender implements PeralatanRumahTangga {
    @Override
    public void hidupkan() {
        System.out.println("Blender dihidupkan");
    }

    @Override
    public void matikan() {
        System.out.println("Blender dimatikan");
    }

    @Override
    public void haluskan() {
        System.out.println("Blender menghaluskan bahan");
    }

    @Override
    public void panaskan() {
        // Tidak diperlukan, tapi harus diimplementasikan
    }

    @Override
    public void bekukan() {
        // Tidak diperlukan, tapi harus diimplementasikan
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Blender blender = new Blender();
        blender.hidupkan();
    }
}
```

```

        blender.haluskan();
    }
}

```

b. Output:

```

// Main.java
public class Main {
    public static void main(String[] args) {
        Blender blender = new Blender();
        blender.hidupkan();
        blender.haluskan();
    }
}

// Run Output
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Users\USER\IntelliJ IDEA Community Edition 2022.3.2\lib\idea_rt.jar=64693:C:\Users\USER
Blender dihidupkan
Blender menghaluskan bahan

Process finished with exit code 0

```

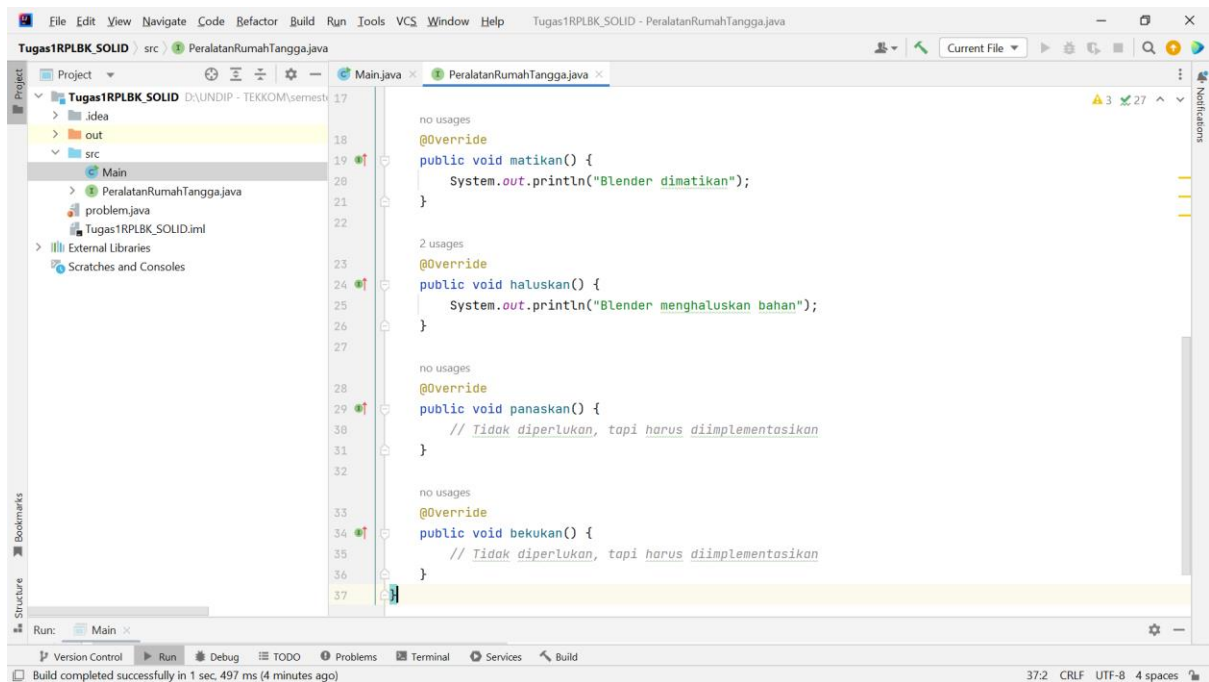
```

// PeralatanRumahTangga.java
// Pelanggaran Interface Segregation Principle

interface PeralatanRumahTangga {
    void hidupkan();
    void matikan();
    void panaskan();
    void bekukan();
    void haluskan();
}

// Blender implements PeralatanRumahTangga
class Blender implements PeralatanRumahTangga {
    @Override
    public void hidupkan() {
        System.out.println("Blender dihidupkan");
    }
}

```



c. Penjelasan

Pada contoh ini, sebuah antarmuka besar digunakan untuk berbagai peralatan rumah tangga. Kelas Blender dipaksa mengimplementasikan metode yang tidak relevan seperti panaskan dan bekukan. Blender dipaksa untuk mengimplementasikan metode panaskan dan bekukan, padahal metode tersebut tidak relevan dengan fungsi Blender.

B. Solver

a. Code

```

// Solusi menggunakan Interface Segregation Principle

interface Mesin {
    void hidupkan();
    void matikan();
}

interface BlenderFungsi {
    void haluskan();
}

interface KulkasFungsi {
    void bekukan();
}

interface KomporFungsi {
    void panaskan();
}

// Blender hanya mengimplementasikan metode yang relevan
class Blender implements Mesin, BlenderFungsi {
    @Override

```

```

    public void hidupkan() {
        System.out.println("Blender dihidupkan");
    }

    @Override
    public void matikan() {
        System.out.println("Blender dimatikan");
    }

    @Override
    public void haluskan() {
        System.out.println("Blender menghaluskan bahan");
    }
}

```

```

public class Main {
    public static void main(String[] args) {
        Blender blender = new Blender();
        blender.hidupkan();
        blender.haluskan();
    }
}

```

b. Output:

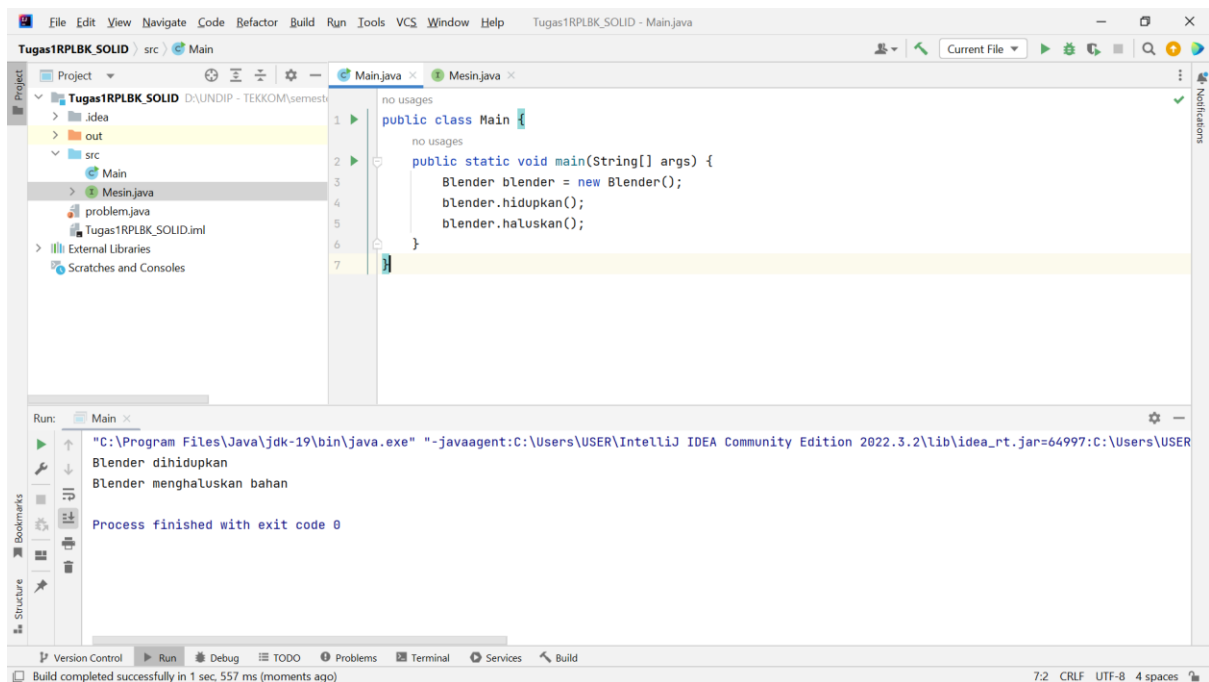
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:

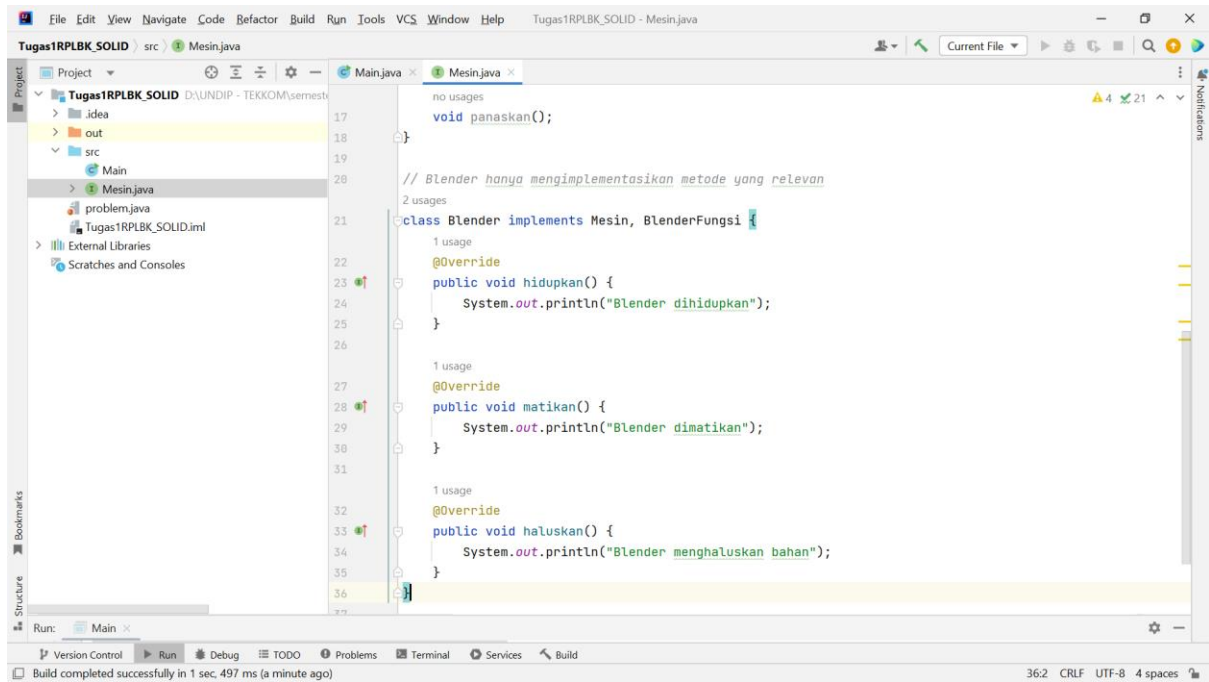
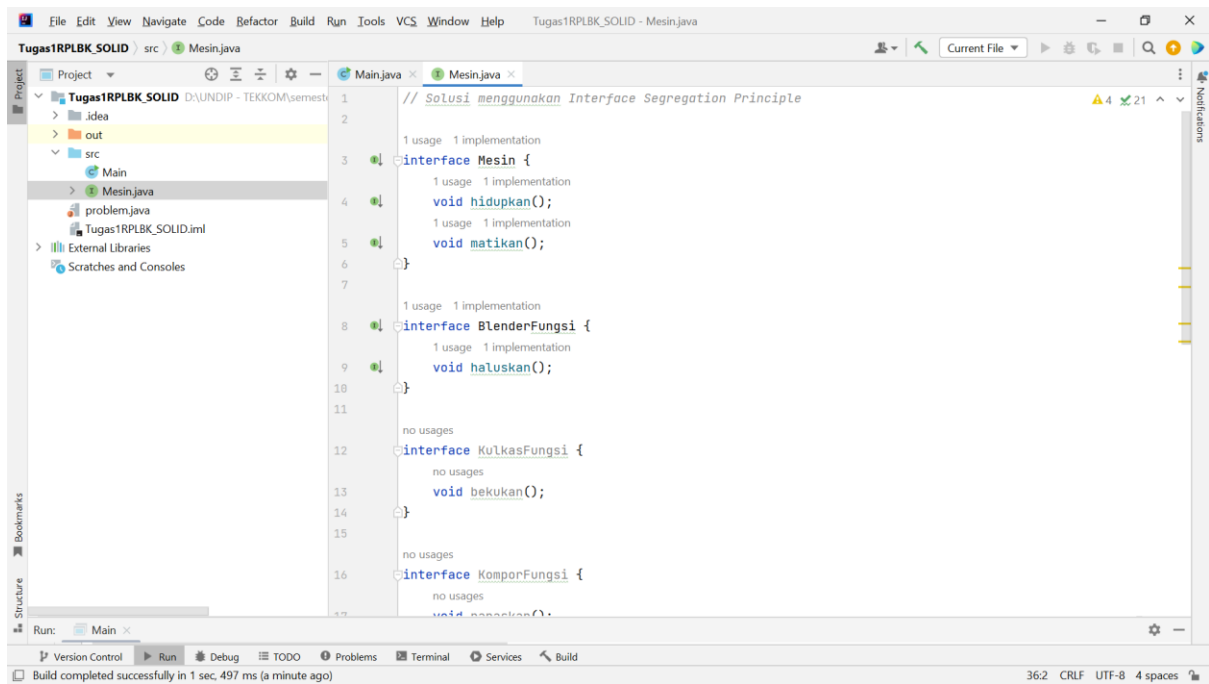
Blender dihidupkan

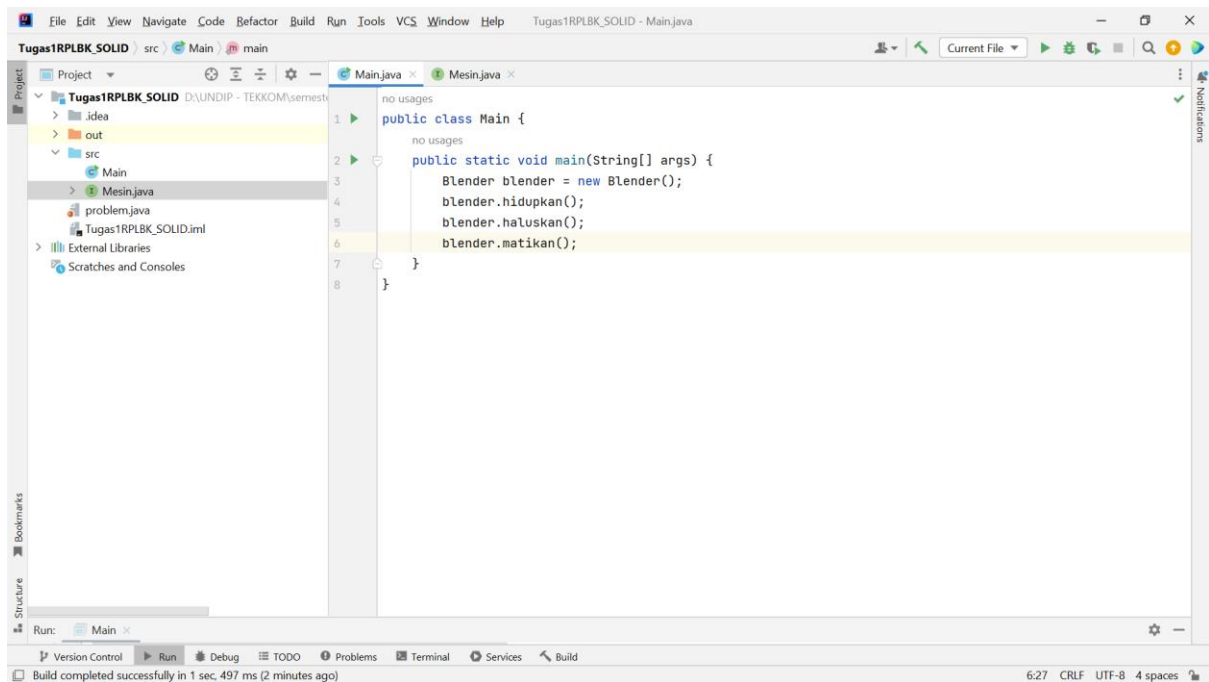
Blender menghaluskan bahan

Blender dimatikan

Process finished with exit code 0







c. Penjelasan

Di sini, antarmuka dipecah menjadi beberapa antarmuka kecil sesuai kebutuhan. Blender hanya mengimplementasikan metode yang relevan. Antarmuka dipecah menjadi BlenderFungsi, KulkasFungsi, dan KomporFungsi. Blender hanya mengimplementasikan antarmuka Mesin dan BlenderFungsi, tanpa harus terlibat dengan metode yang tidak dibutuhkannya seperti panaskan dan bekukan.