TUGAS PRAKTIKUM 6
PEMROGRAMAN BERORIENTASI OBJEK

Oleh :

RAHMADITYA PUTRI LAILATUL ISMI
21091397036
2021B

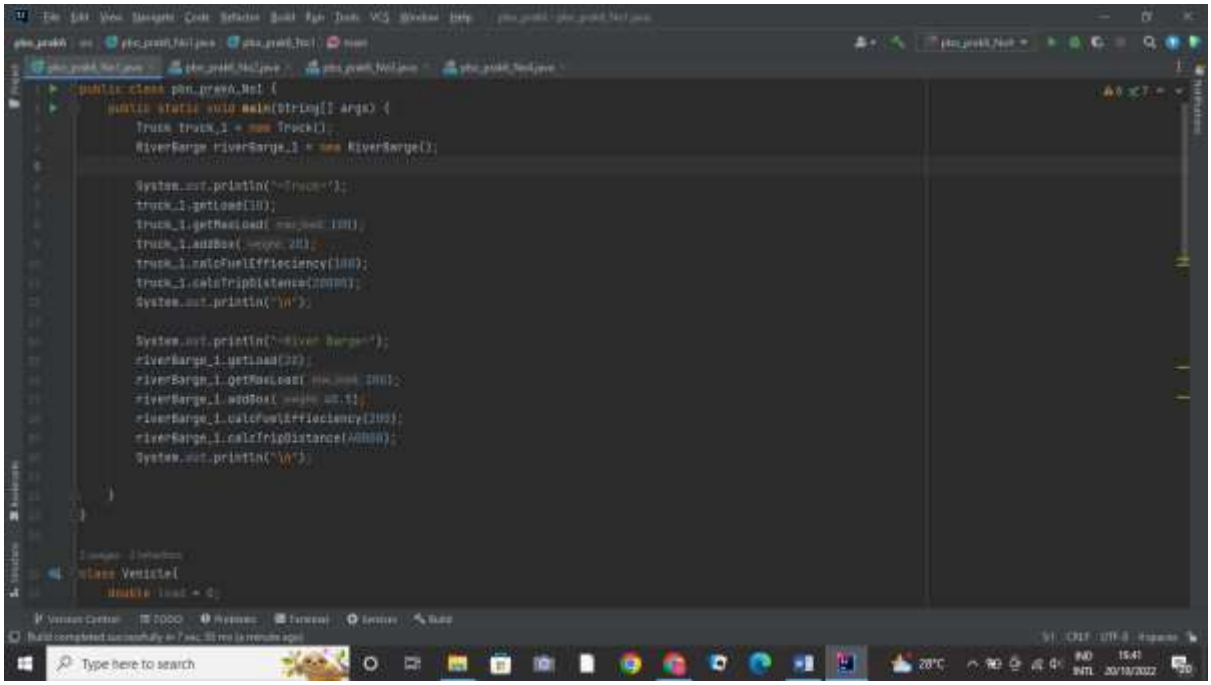
PROGRAM STUDI MANAJEMEN INFORMATIKA

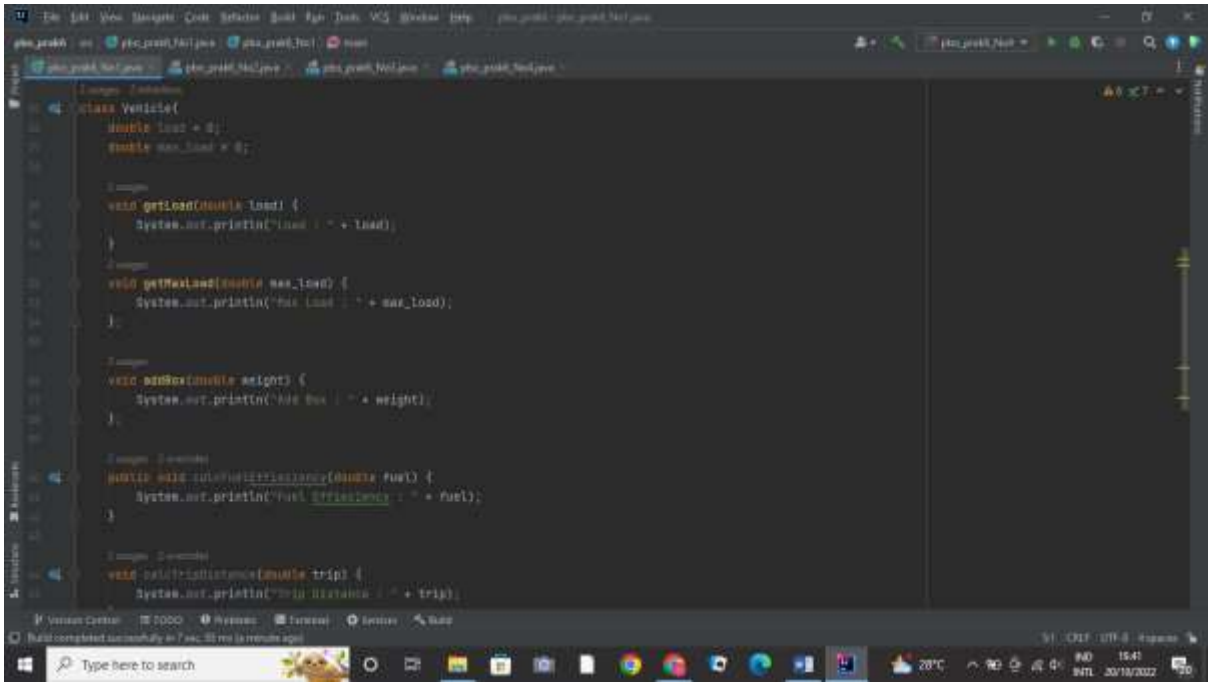
FAKULTAS VOKASI

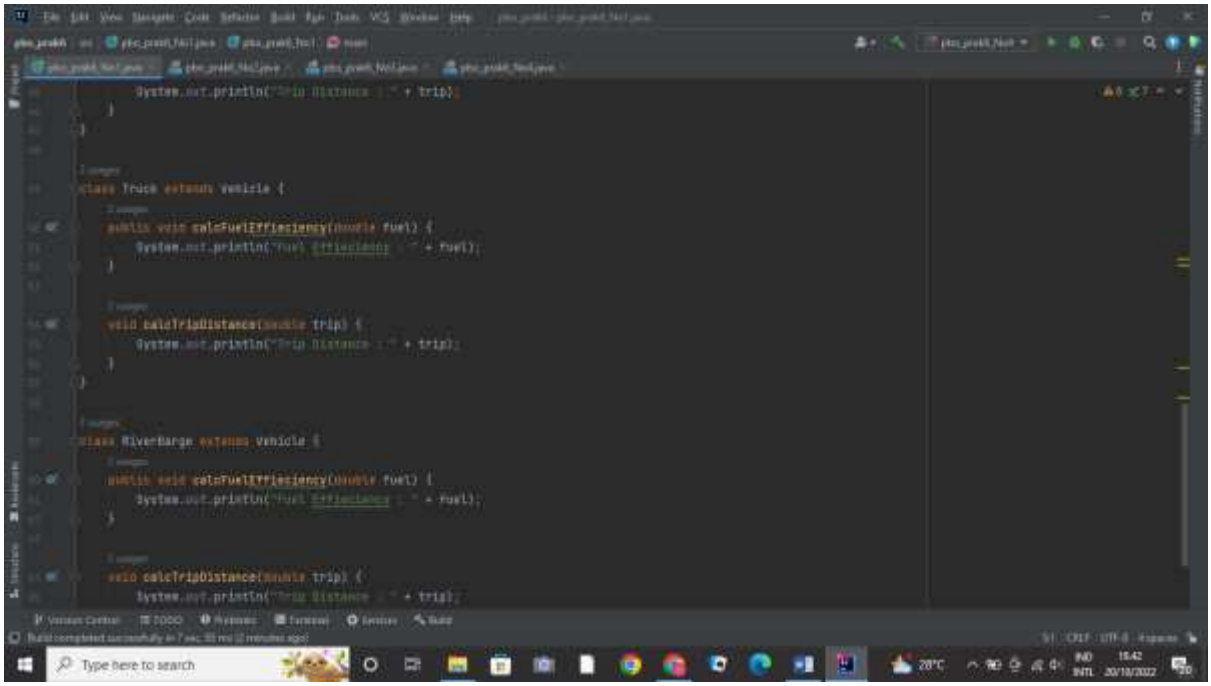
UNIVERSITAS NEGERI SURABAYA

2022

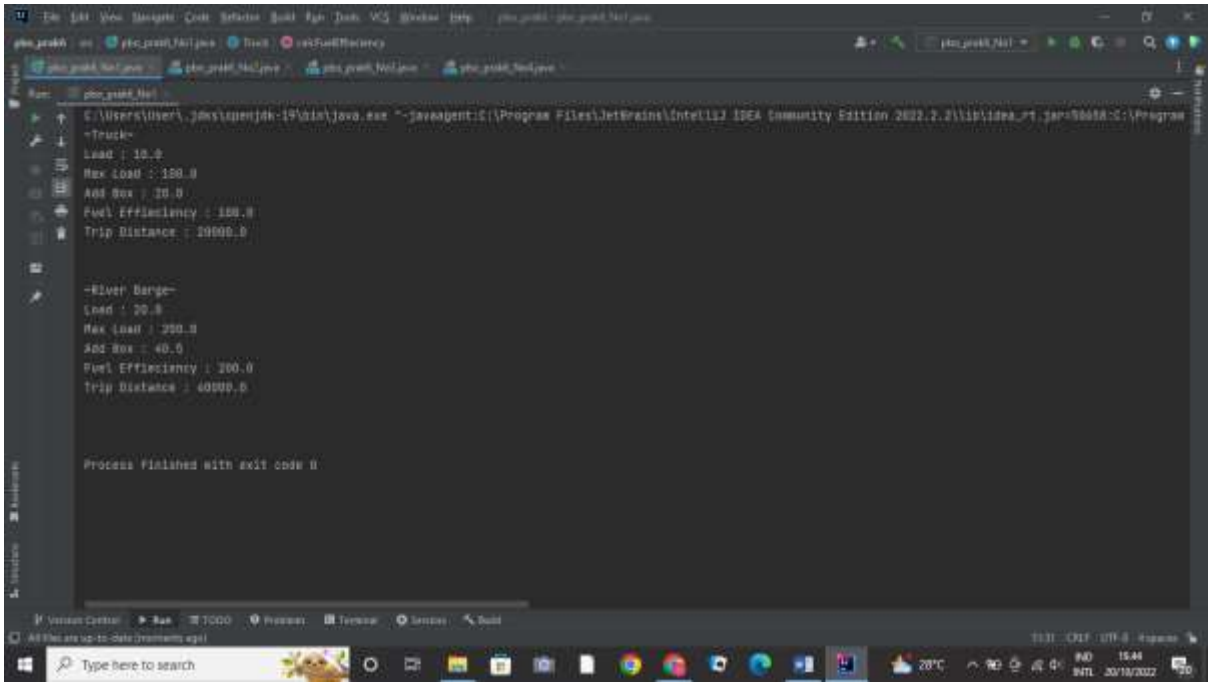
1.





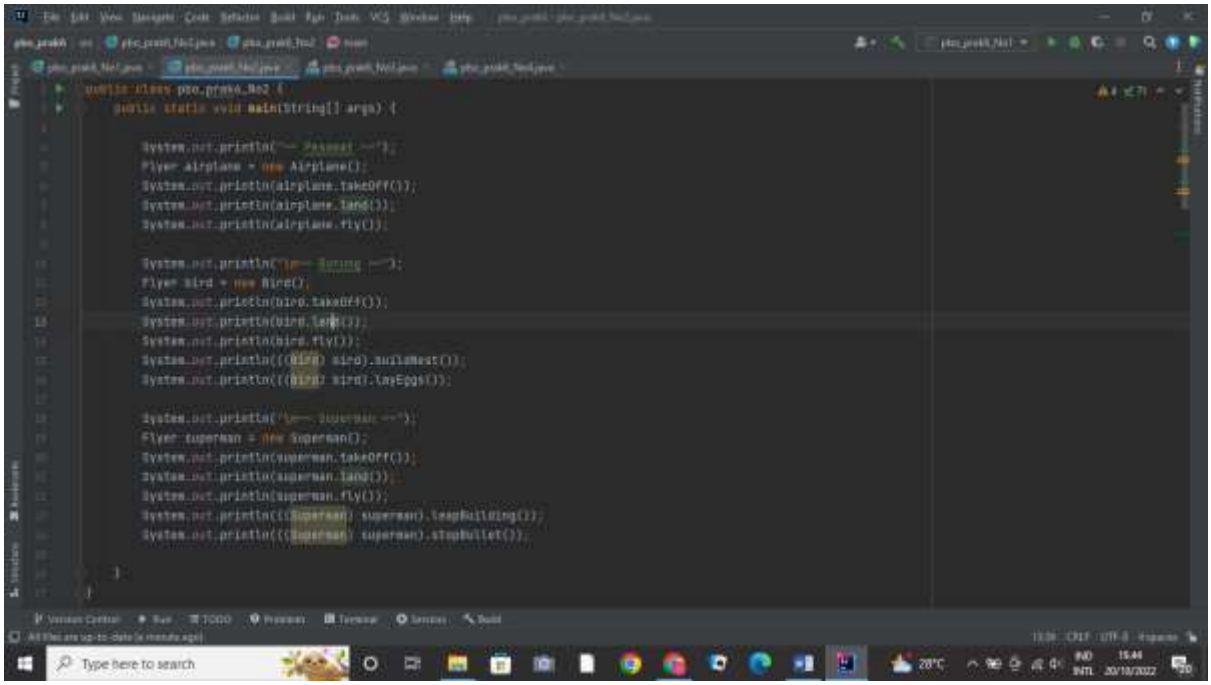


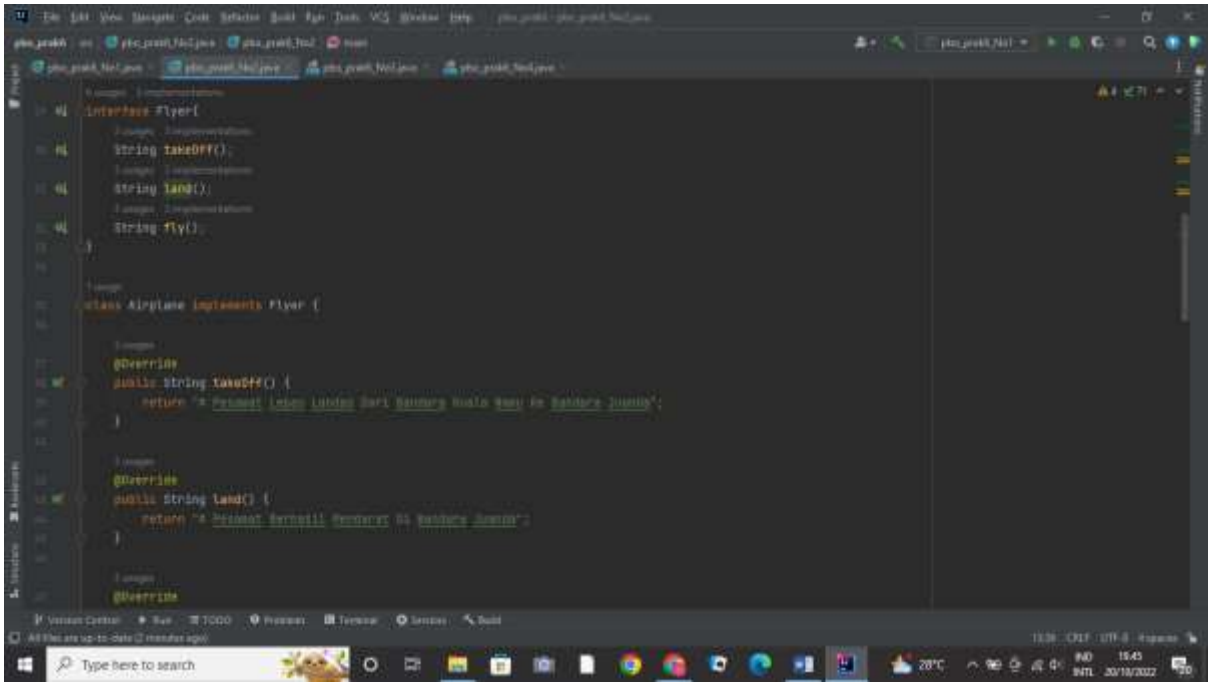
HASIL

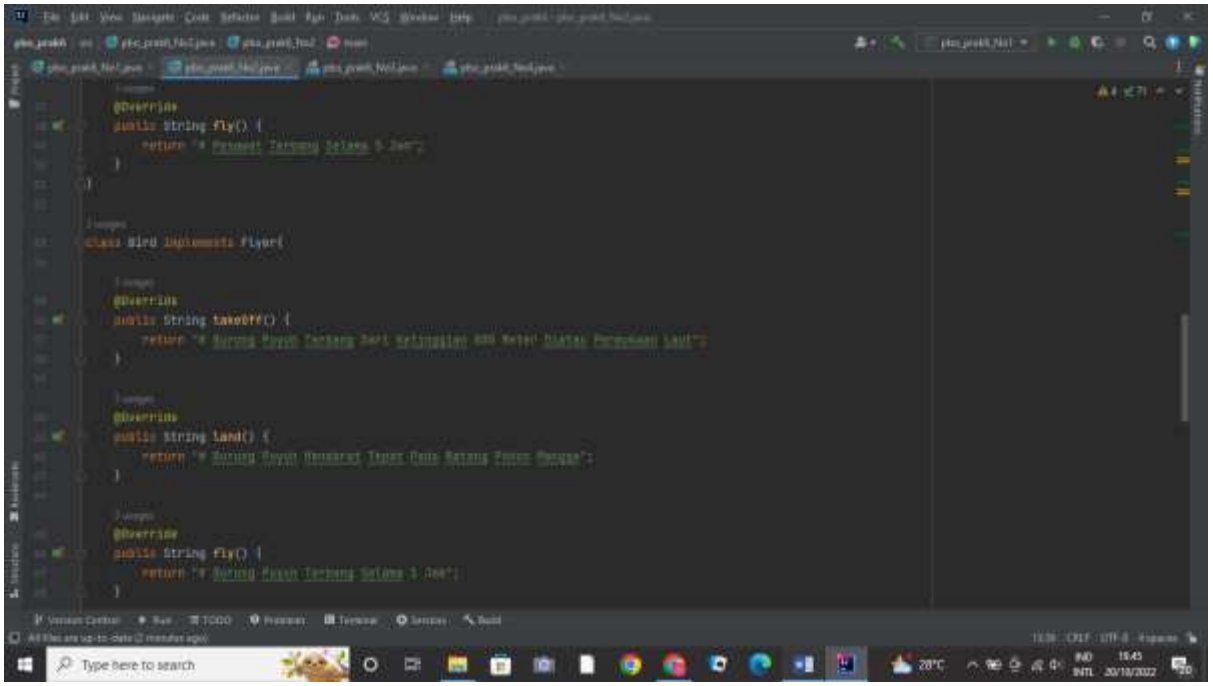


Dari kode program diatas terdapat 3 class yaitu vehicle, truck, riverbarge dimana class vehicle nantinya mewariskan methodnya ke class childnya. Didalam class truck dan riverbarge terdapat 2 method yang sama yaitu method bensin dan jarak tempuh. Kemudian, method truck dan riverbarge dipanggil difungsi main.

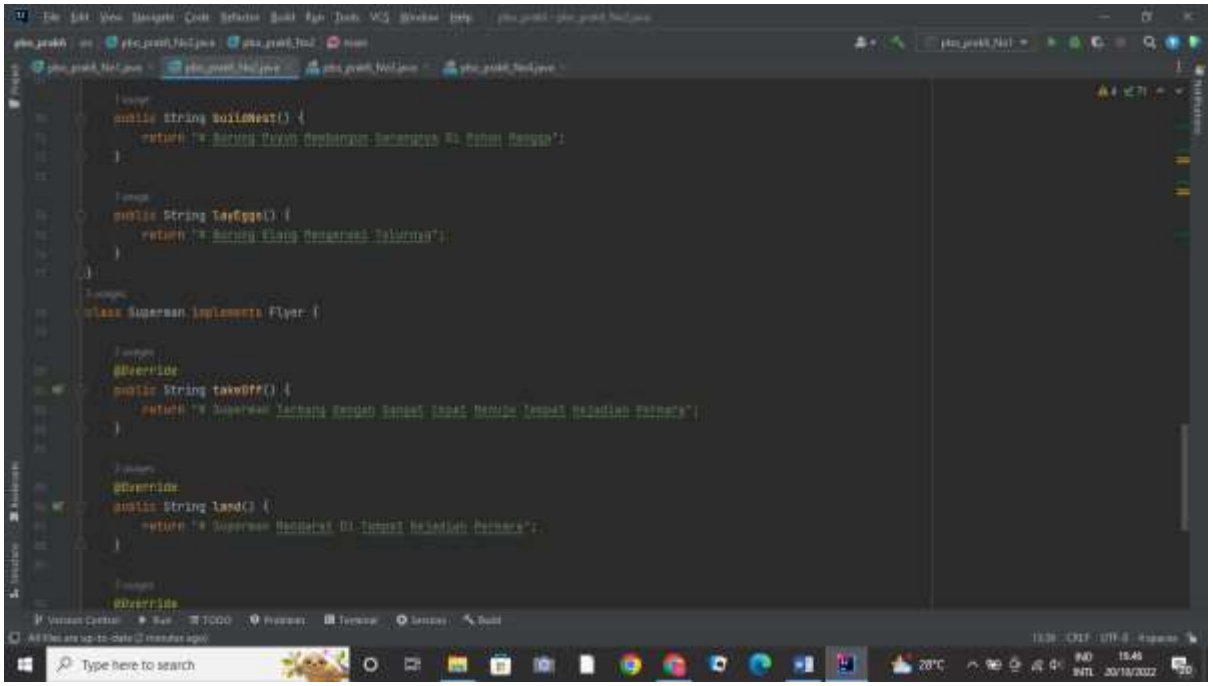
2.



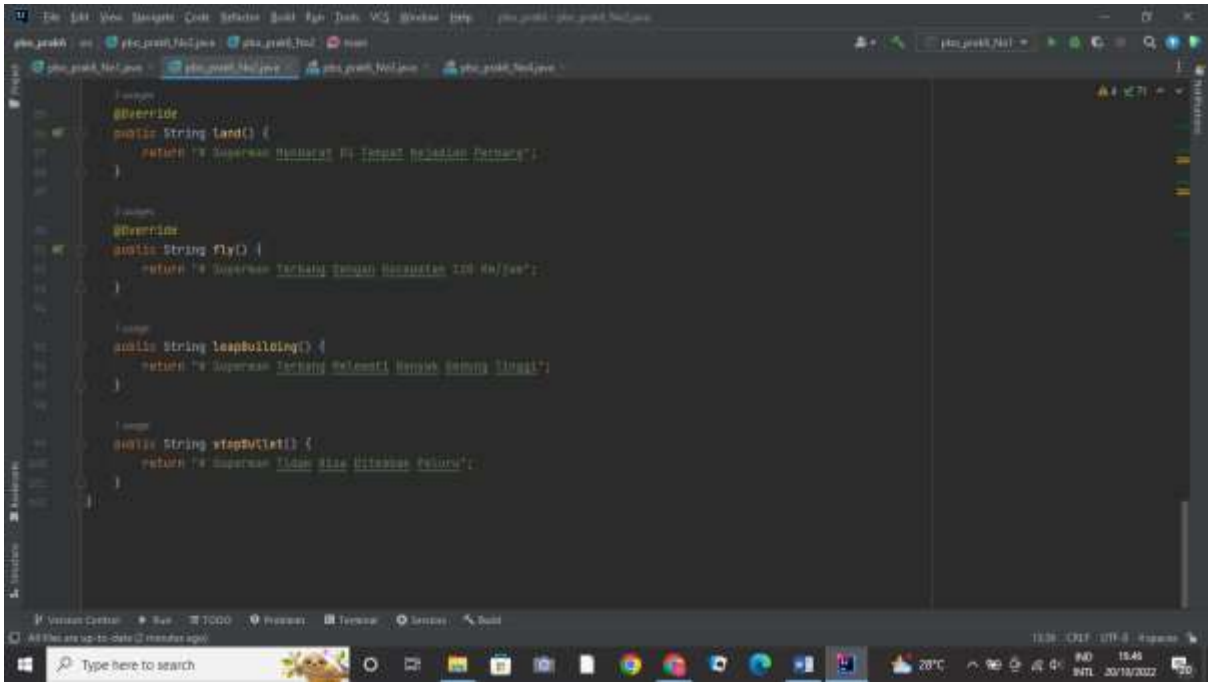




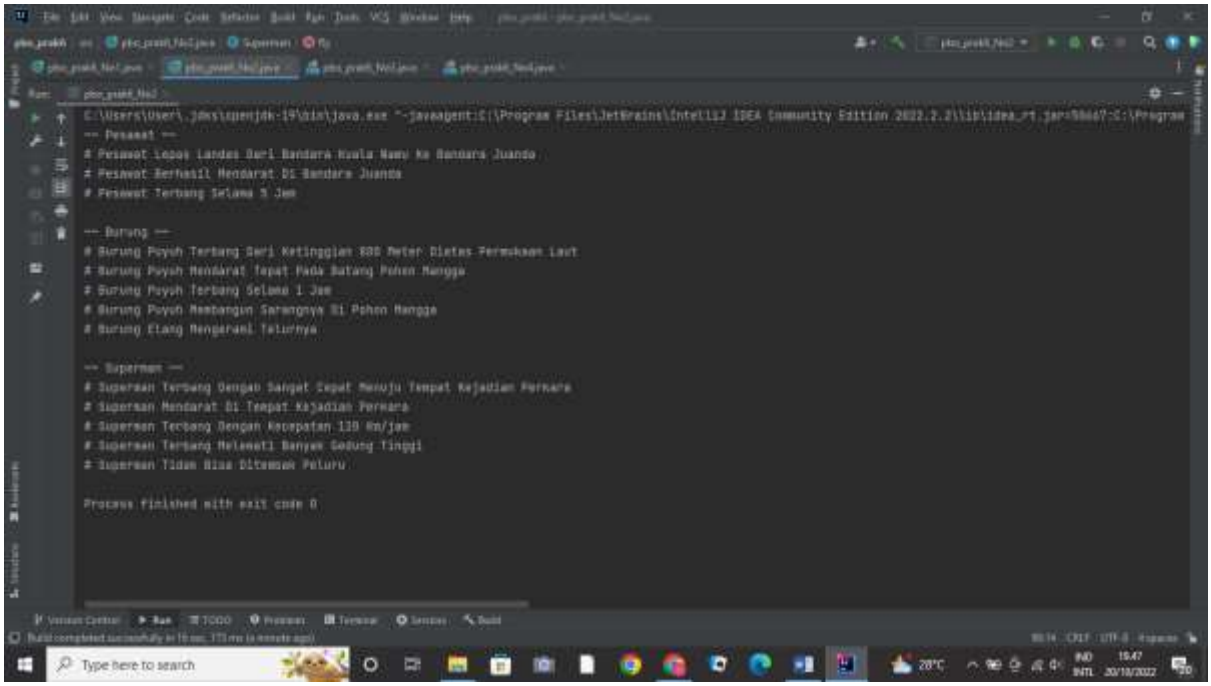
```
    @Override
    public String fly() {
        return "# Peswat Terbang Selama 5 Jam";
    }
}

class Bird implements Flyer{

    @Override
    public String takeOff() {
        return "# Burung Puyuh Terbang Dari Ketinggian 800 Meter Diatas Permukaan Laut";
    }

    @Override
    public String land() {
        return "# Burung Puyuh Mendarat Tepat Pada Batang Pohon Mangga";
    }

    @Override
    public String fly() {
        return "# Burung Puyuh Terbang Selama 3 Jam";
    }
```



```
    public String buildNest() {
        return "# Burung Puyuh Membangun Sarangnya Di Pohon Mangga";
    }

    public String layEggs() {
        return "# Burung Biang Mengerami Telurnya";
    }
}
class Superman implements Flyer {

    @Override
    public String takeOff() {
        return "# Superman Terbang Dengan Sangat Ingat Menuju Tempat Kejadian Perkara";
    }

    @Override
    public String land() {
        return "# Superman Mendarat Di Tempat Kejadian Perkara";
    }

    @Override
```

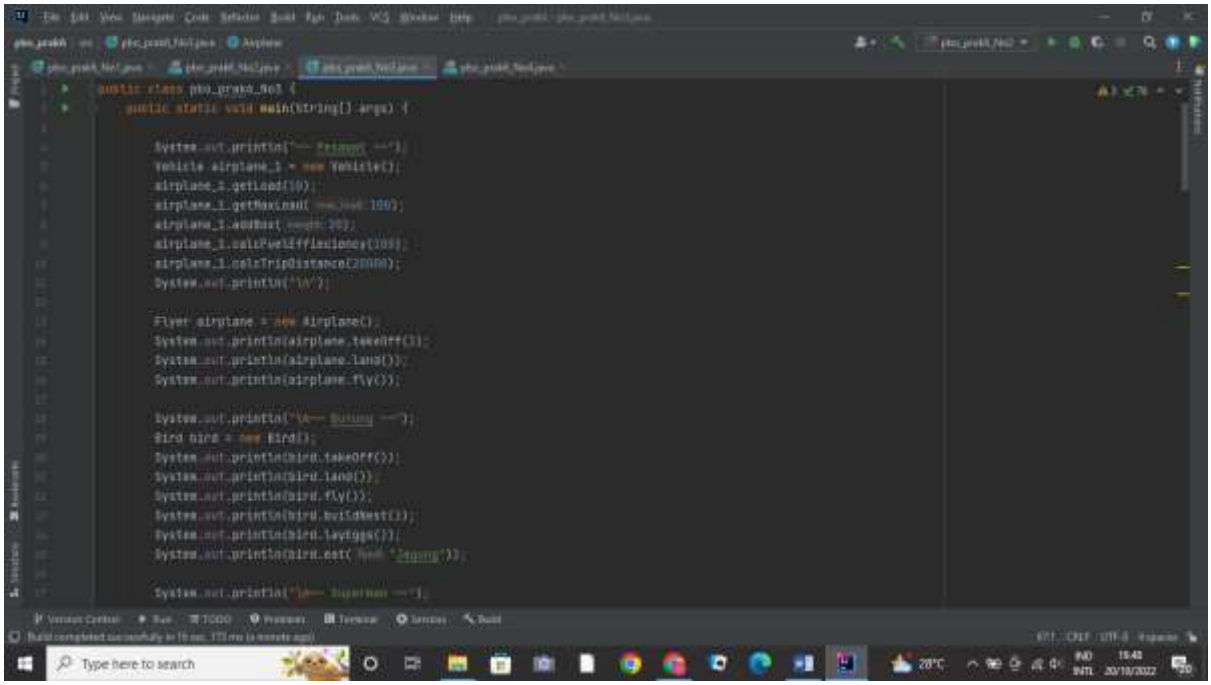


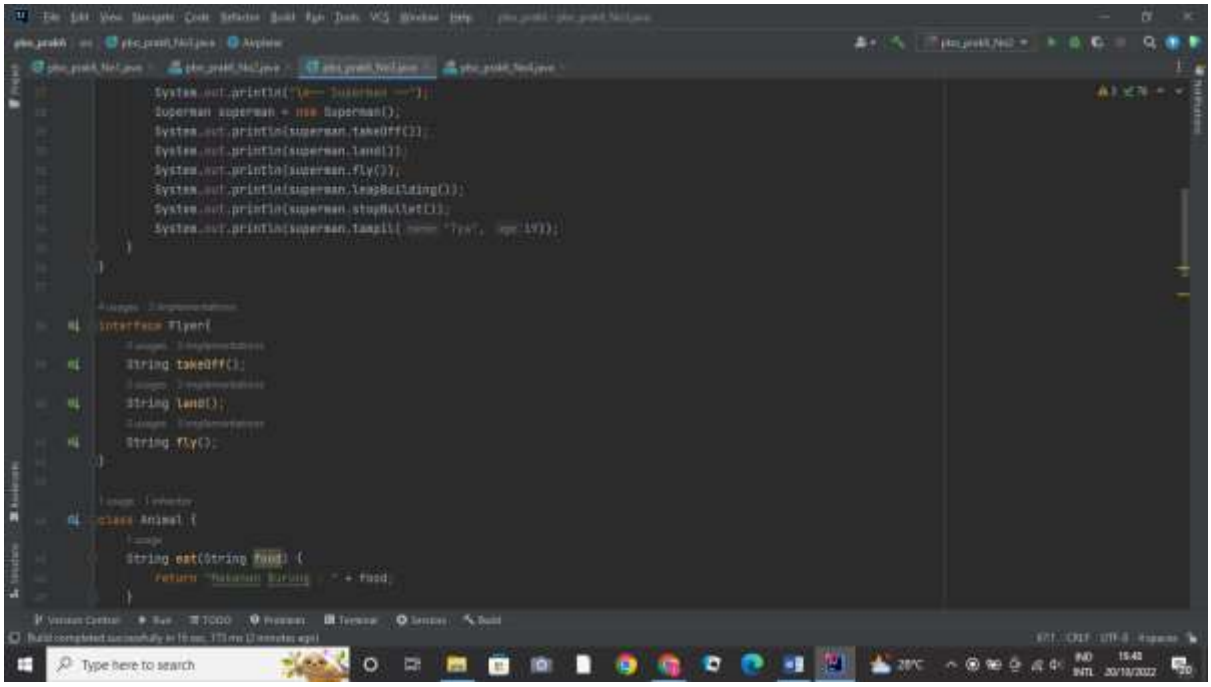
HASIL

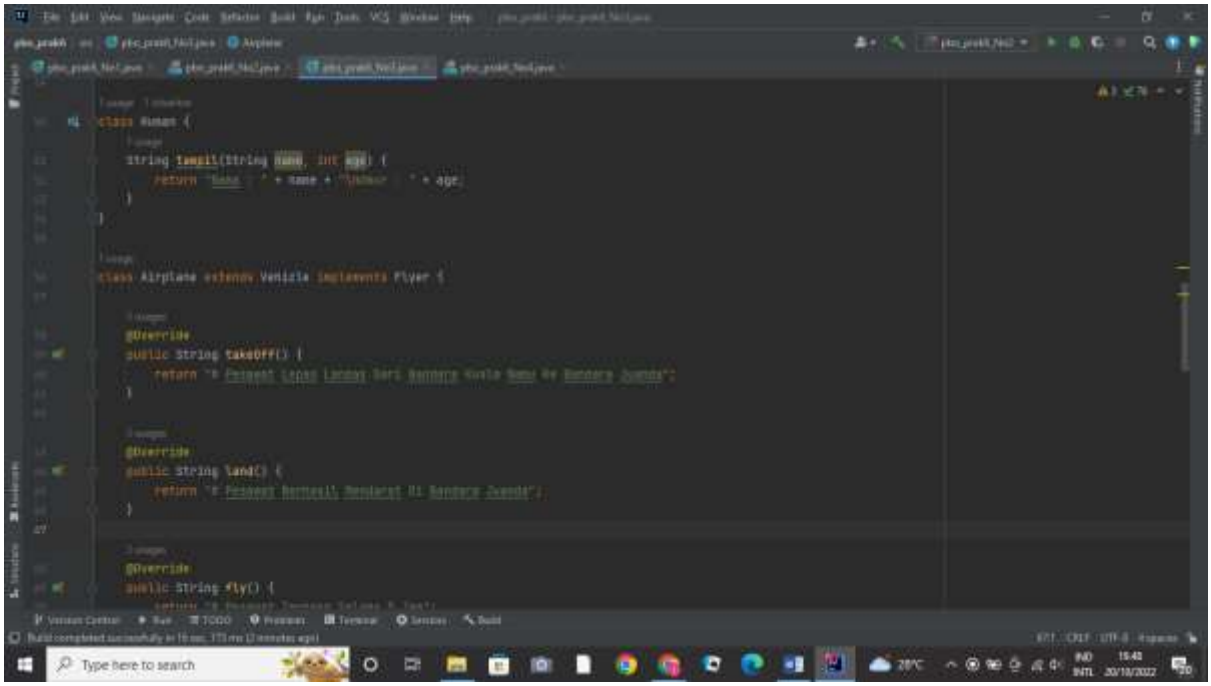


Pada kode program diatas terdapat 1 interface yang bernama flyer. Interface flyer memiliki 3 method yang nantinya dapat diimplementasikan pada kelas yang membutuhkannya. Terdapat class airplane, bird, superman yang mengimplementasikan interface flyer, masing2 classnya mengoverride method interfacenya. Fungsi main terdapat sintaks untuk menginisiasi class airplane, bird, supeman dan menampilkan hasilnya.

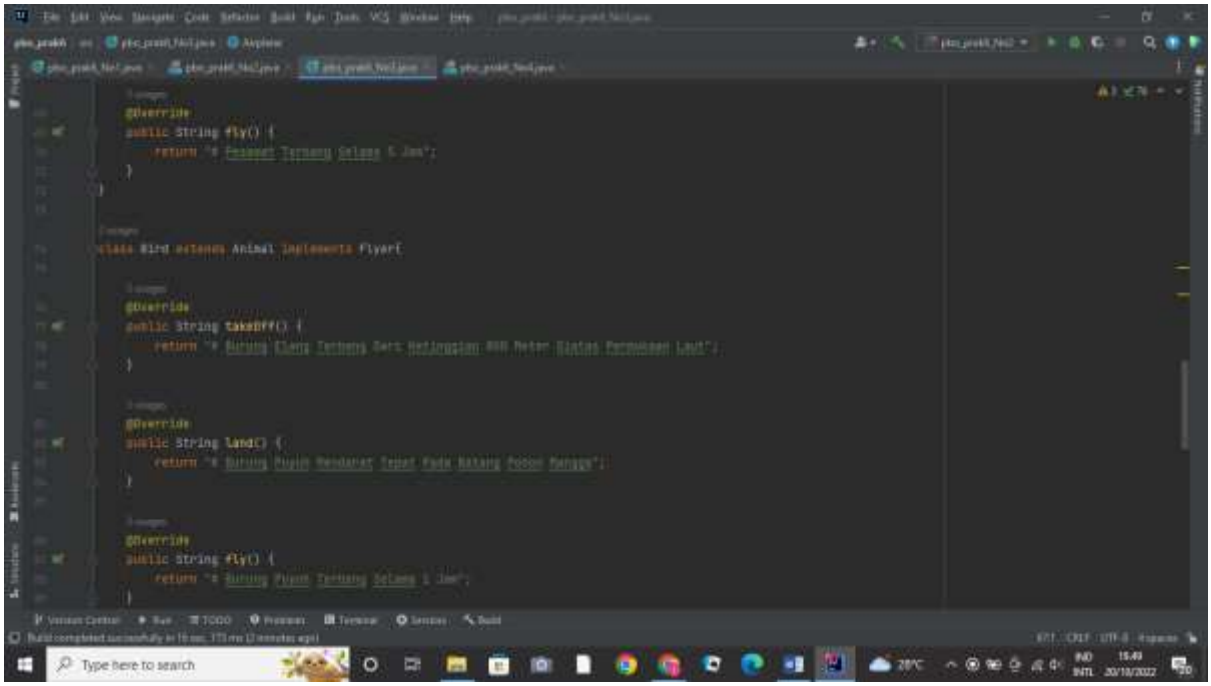
3.



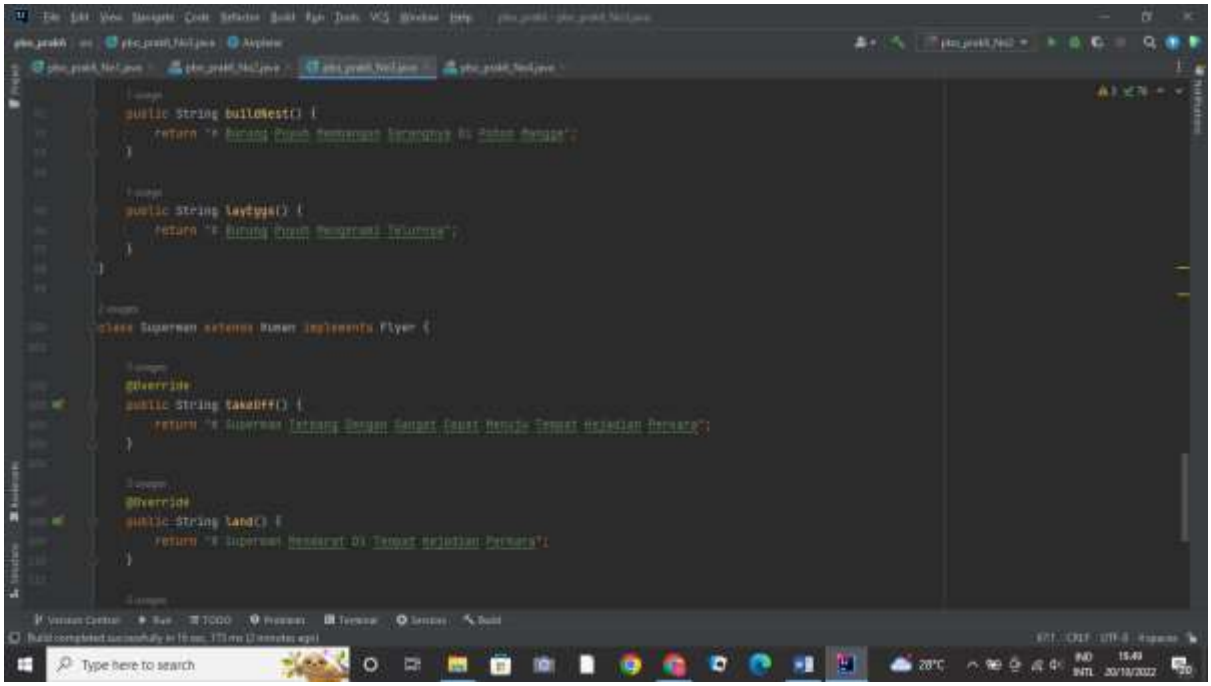


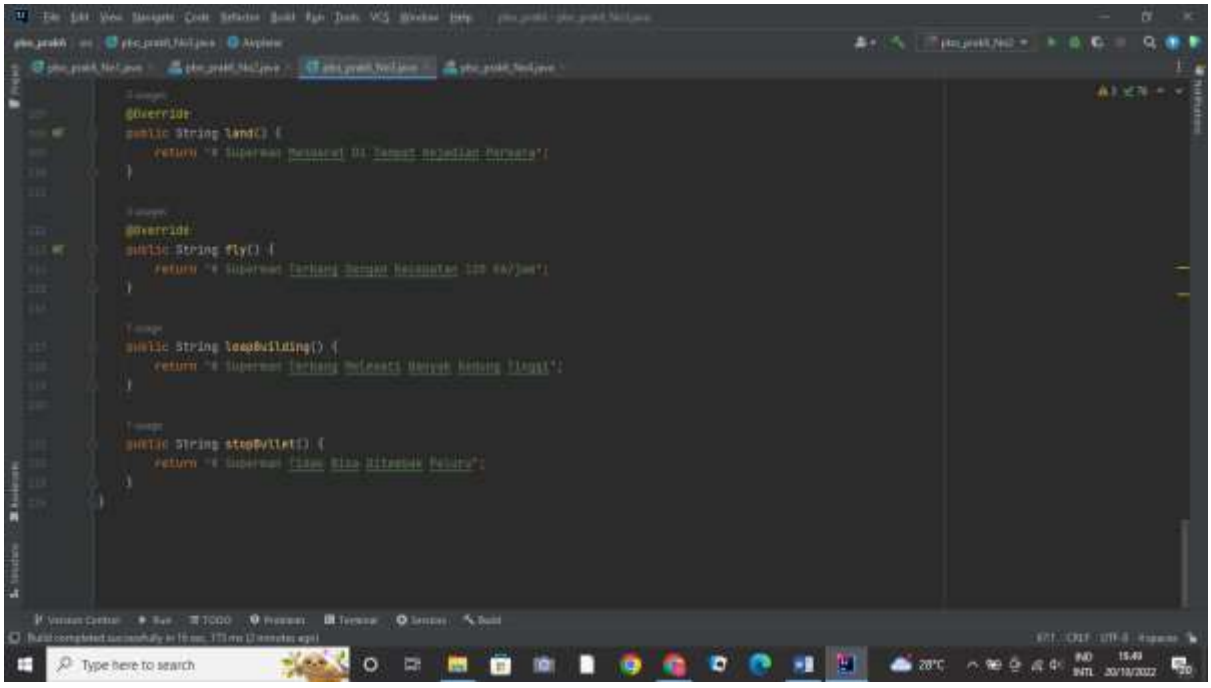


```
class Human {
    String tampil(String nama, int age) {
        return "Nama  " + nama + "\nUmur   " + age;
    }
}

class Airplane extends Vehicle implements Flyer {

    @Override
    public String takeOff() {
        return "= Pesawat Lepas Landas Dari Bandara Kuala Namu Ke Bandara Juanda";
    }

    @Override
    public String land() {
        return "= Pesawat Berhasil Mendarat Di Bandara Juanda";
    }

    @Override
    public String fly() {
```

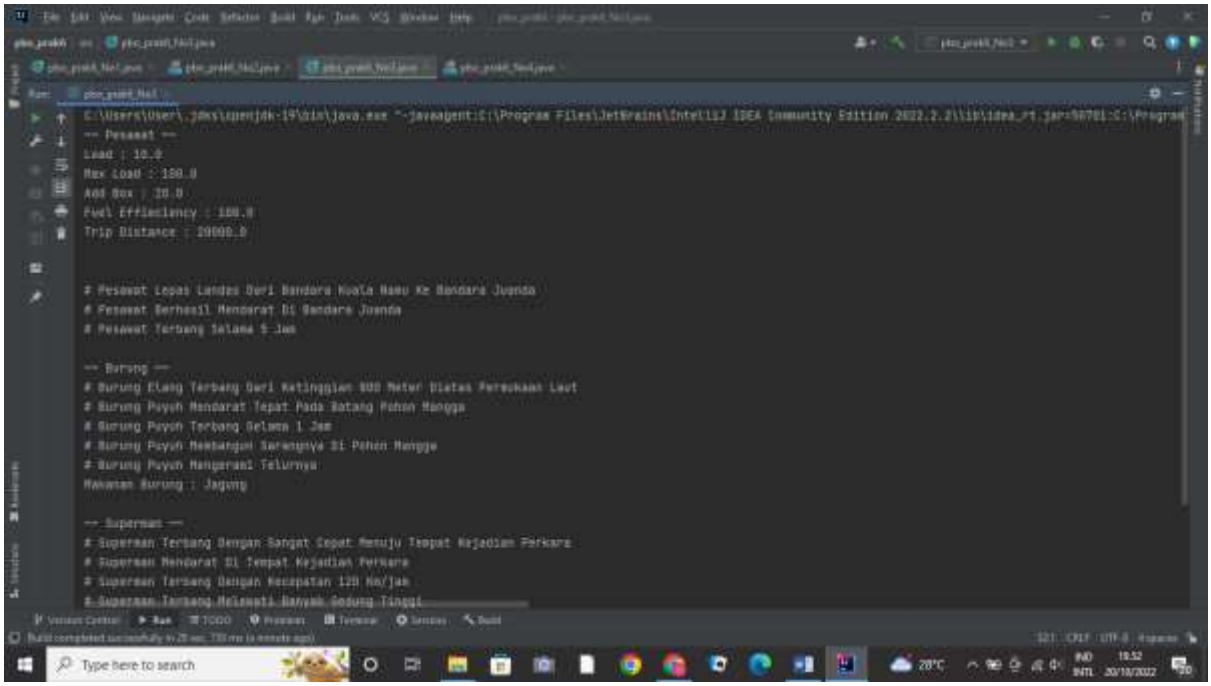


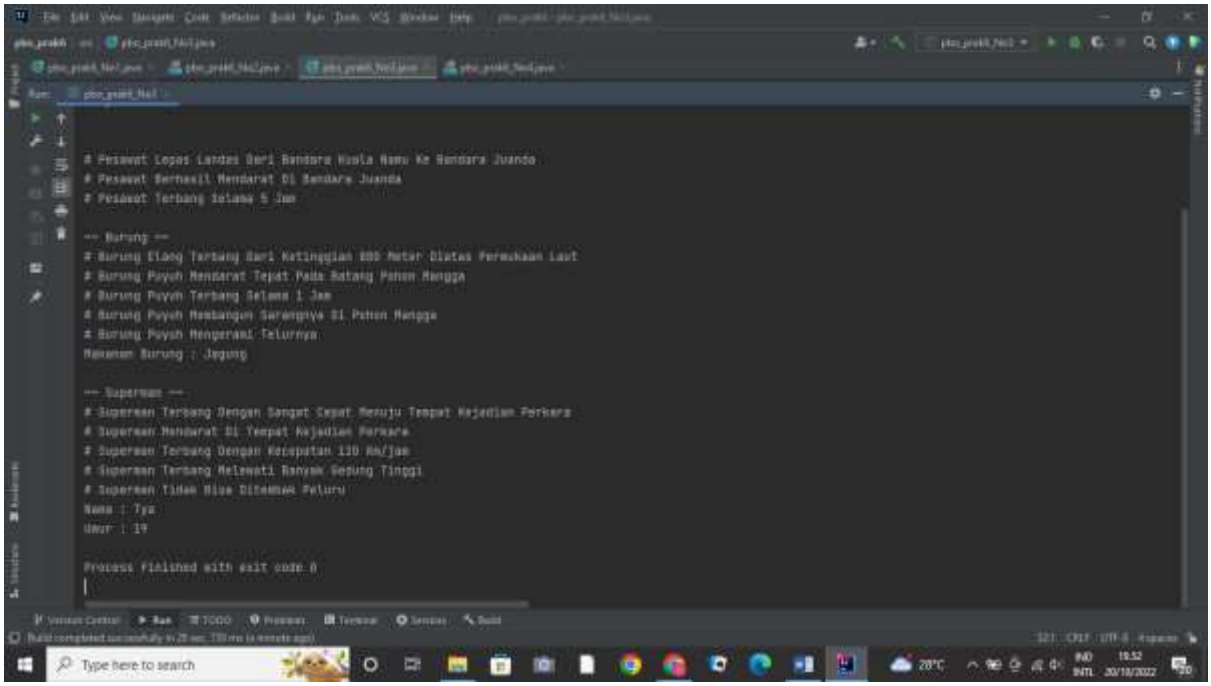
```
    @Override
    public String fly() {
        return "= Pesawat Terbang Selama 5 Jam";
    }
}

class Bird extends Animal implements Flyer{

    @Override
    public String takeOff() {
        return "= Burung Elang Terbang Dari Ketinggian 800 Meter Diatas Permukaan Laut";
    }

    @Override
    public String land() {
        return "= Burung Puyuh Mendarat Tepat Pada Batang Pohon Mangga";
    }

    @Override
    public String fly() {
        return "= Burung Puyuh Terbang Selama 1 Jam";
    }
```





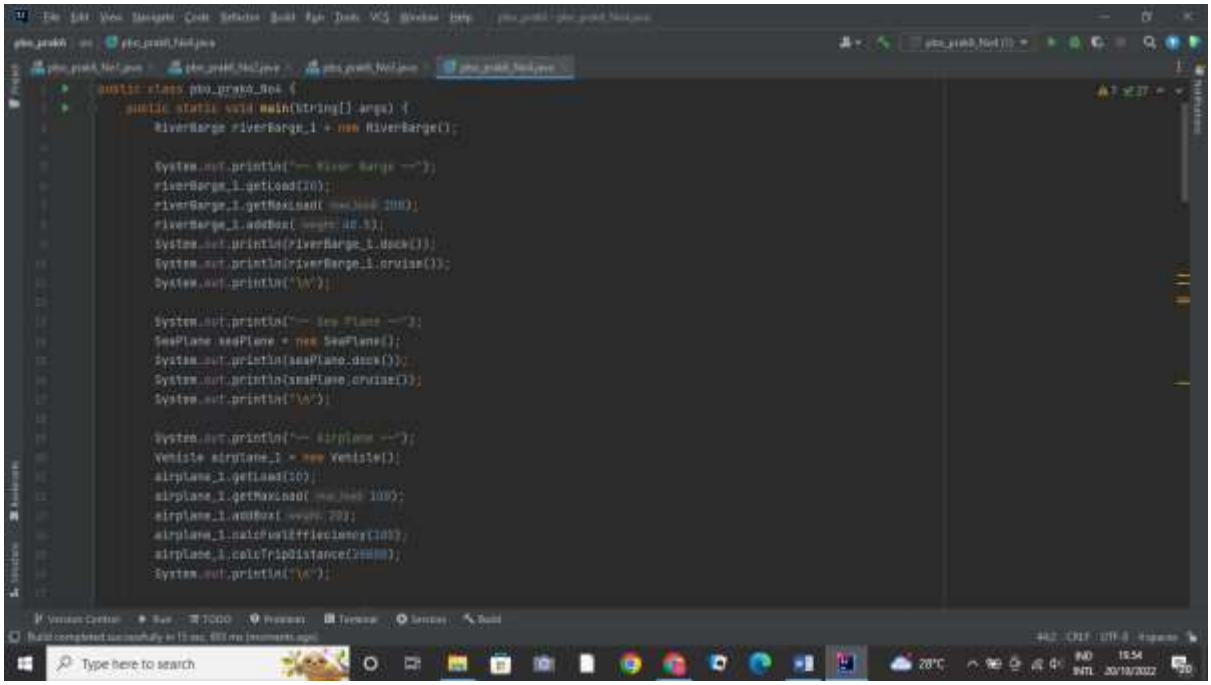
HASIL

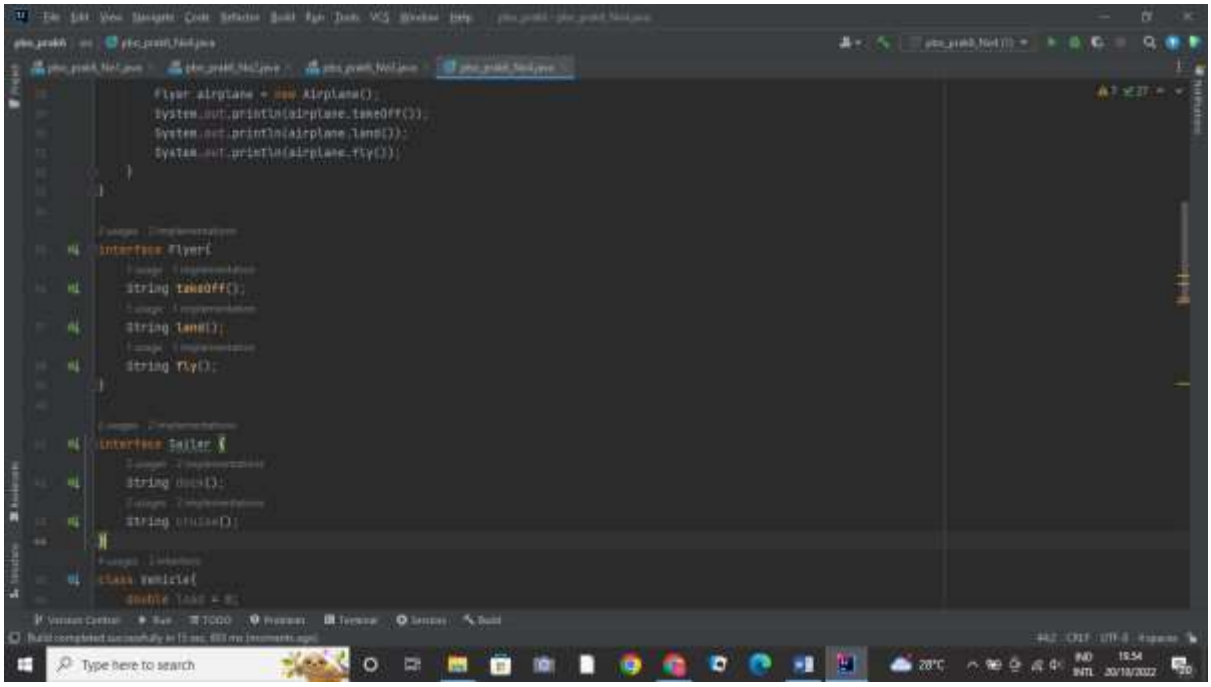


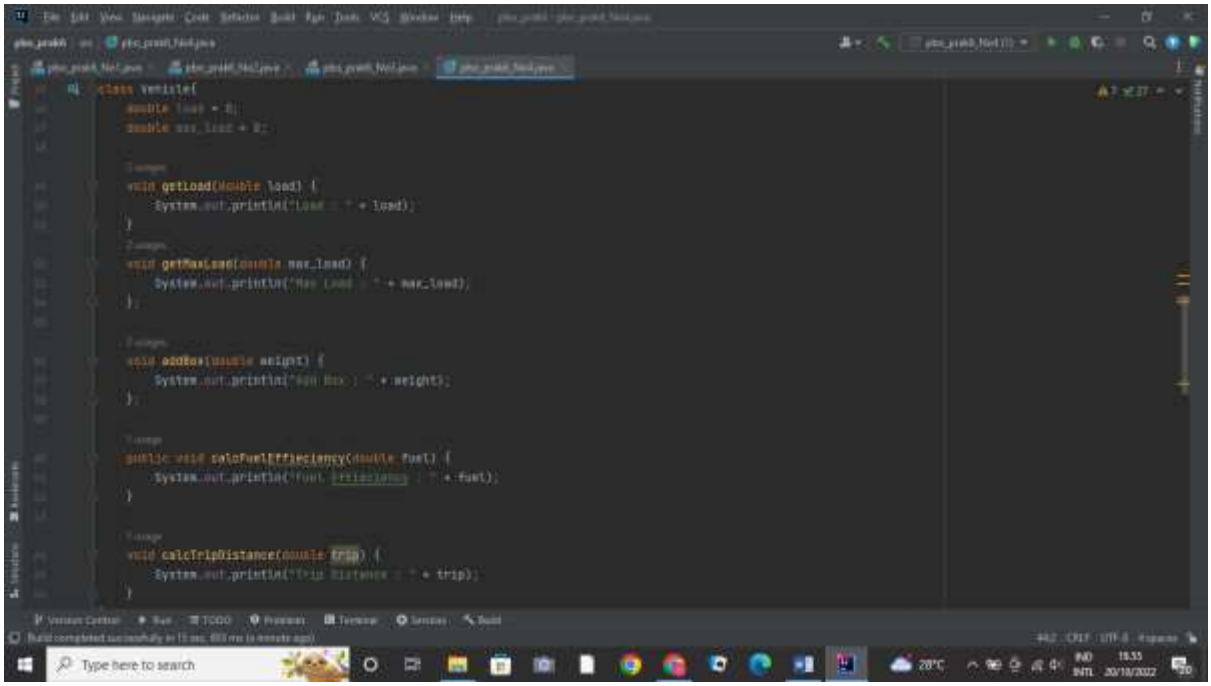


Sama seperti kode program sebelumnya, kode diatas memiliki interface flyer yang memiliki 3 method yang sama. Kode program diatas terdapat 2 class tambahan yaitu class animal dan human. Berbeda dengan kode sebelumnya, class airplane, bird, dan superman mewarisi dari class induknya. Class airplane mewarisi class vehicle yang terdapat pada soal no.1. class bird mewarisi class animal dan class superman mewarisi class human. Inisialisasi dan hasil ditampilkan pada fungsi main.

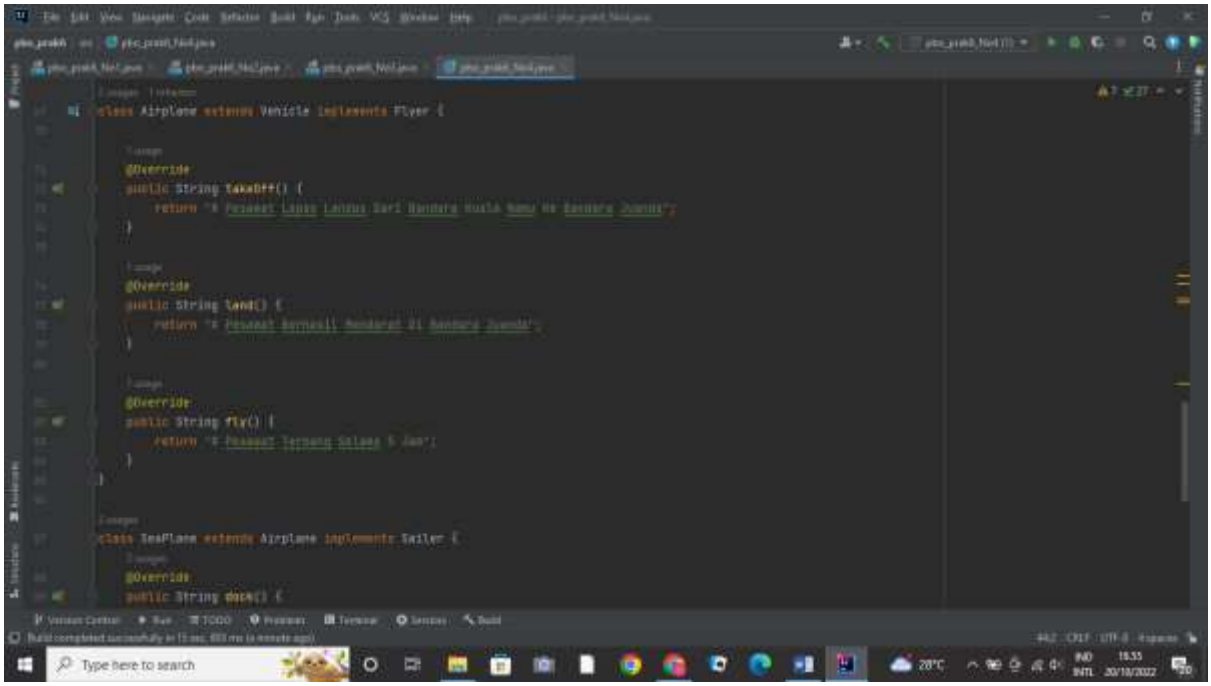
4.



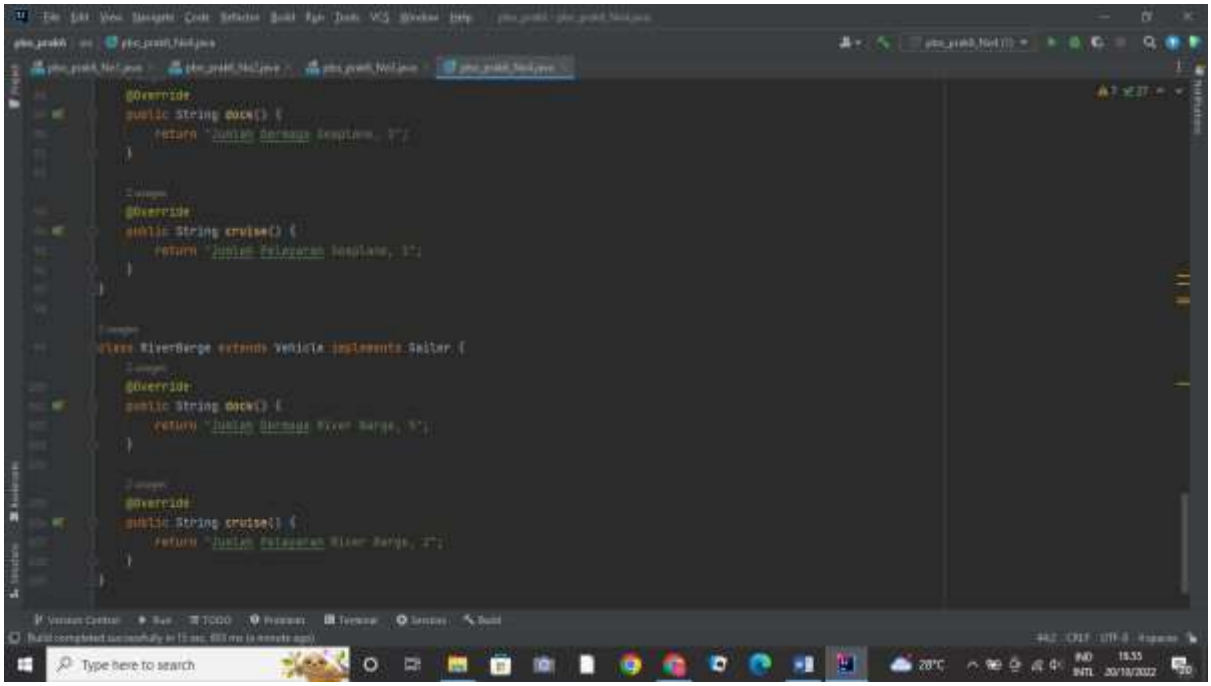




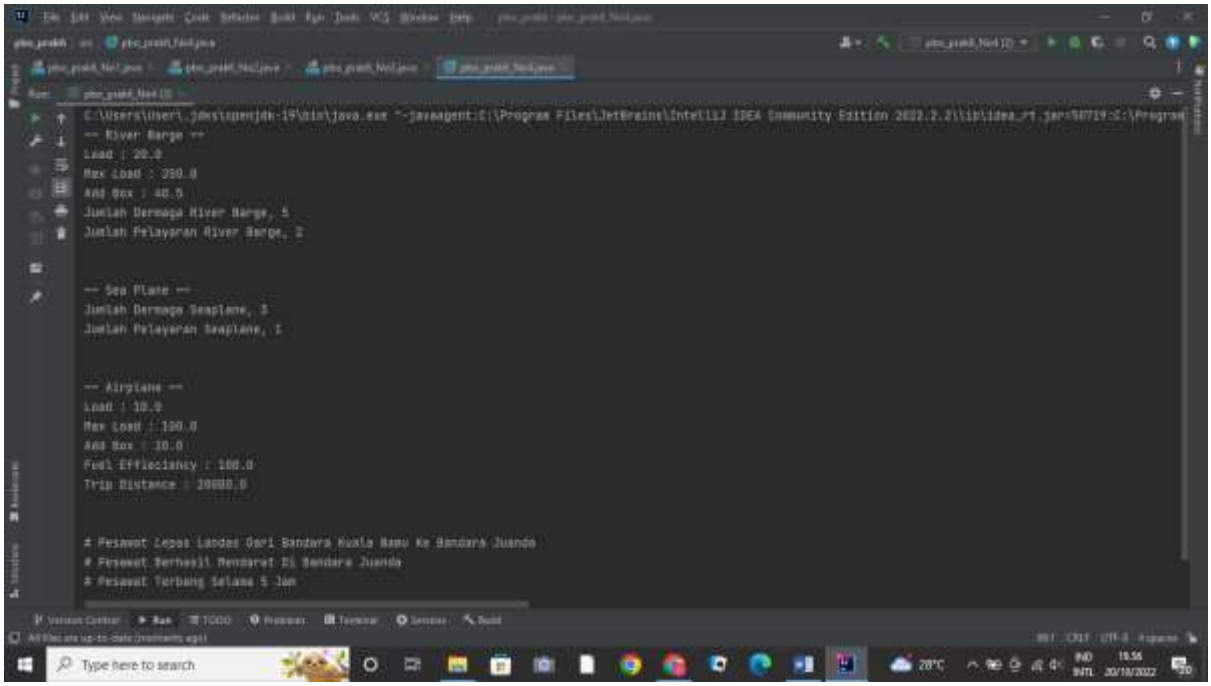
```
class Vehicle{
    double load = 0;
    double max_load = 0;

    void getLoad(double load) {
        System.out.println("Load : " + load);
    }
    void getMaxLoad(double max_load) {
        System.out.println("Max Load : " + max_load);
    }

    void addBox(double weight) {
        System.out.println("Add Box : " + weight);
    }

    public void calcFuelEfficiency(double fuel) {
        System.out.println("Fuel Efficiency : " + fuel);
    }

    void calcTripDistance(double trip) {
        System.out.println("Trip Distance : " + trip);
    }
```



```
class Airplane extends Vehicle implements Flyer {

    @Override
    public String takeOff() {
        return "A Pesawat Lepas Landas Dari Bandara Kuala Namu Ke Bandara Juanda";
    }

    @Override
    public String land() {
        return "A Pesawat Berhasil Mendarat Di Bandara Juanda";
    }

    @Override
    public String fly() {
        return "A Pesawat Terbang Selama 5 Jam";
    }
}

class SeaPlane extends Airplane implements Sailer {
    @Override
    public String dock() {
```



HASIL



Kode program diatas memiliki 2 interface yaitu flyer dan sailer. Class vehicle memiliki 2 variabel dan 5 method yang nantinya akan digunakan pada class turunannya. Class airplane, seaplane, riverbarge sama-sama mewarisi class induknya dan mengimplementasi class interface yang telah dibuat. Inisialisasi class dan pengiriman data pada method terdapat di fungsi main.