

JOBSHEET 8
ABSTRACT CLASS

Nama : Rizqi Rohmatul Huda
Kelas : 2G – TI
No.Absen : 26
NIM : 2141720264

3. PERCOBAAN

A. PERCOBAAN 1

Buatlah project baru dengan nama AbstractClass kemudian buat class baru dengan nama Hewan. Method bernapas dan bergerak tidak memiliki statement atau baris kode.

```
package pemrogramanberbasisobjek.pertemuan10.abstractclass;

public class Hewan {
    public double berat;
    public String makanan;
    public String habitat;

    public Hewan(double berat, String makanan, String habitat){
        this.berat = berat;
        this.makanan = makanan;
        this.habitat = habitat;
    }

    public void bergerak(){
        //
    }

    public void bernapas(){
        //
    }

    public void cetakInfo(){
        System.out.println("Berat\t: " + this.berat);
        System.out.println("Makanan\t: " + this.makanan);
        System.out.println("Habitat\t: " + this.habitat);
    }
}
```

1. Selanjutnya buat class Lebah sebagai subclass dari class Hewan sebagai berikut

```
package pemrogramanberbasisobjek.pertemuan10.abstractclass;

public class Lebah extends Hewan {
    public String kasta;

    public Lebah(String kasta, double berat, String makanan, String habitat){
        super(berat, makanan, habitat);
        this.kasta = kasta;
    }
}
```

2. Buat class main dengan nama AbstractClassDemo lalu instansiasi objek dari class Hewan dan class Lebah. Run program kemudian amati hasilnya.

```
package pemrogramanberbasisobjek.pertemuan10.abstractclass;

public class AbstractClassDemo {
    public static void main(String[] args) {
        Hewan hewan1 = new Hewan(10, "Rumput", "Savana");
        hewan1.cetakInfo();
        hewan1.bergerak();
        hewan1.bernapas();

        Lebah lebah1 = new Lebah("Ratu", 0.05, "Nektar", "Hutan");
        lebah1.cetakInfo();
        lebah1.bergerak();
        lebah1.bernapas();
    }
}
```

B. PERTANYAAN

1. Bagaimana hasil pada langkah 3? Apakah objek hewan1 dan lebah1 dapat diinstansiasi?

Jawab : Iya, masih dapat diinstansiasi karena belum dideklarasikan sebagai abstract class.

2. Class Lebah tidak memiliki method bergerak(), bernapas(), dan cetakInfo(), mengapa tidak terjadi error pada AbstractClassDemo?

Jawab : karena atribut dan method diwariskan dari parent class (class Hewan) kepada subclass (class Lebah) .

C. PERCOBAAN 2

1. Ubah method bergerak dan bernapas menjadi abstract method.

```
package pemrogramanberbasisobjek.pertemuan10.abstractclass;

public class Hewan {
    public double berat;
    public String makanan;
    public String habitat;

    public Hewan(double berat, String makanan, String habitat){
        this.berat = berat;
        this.makanan = makanan;
        this.habitat = habitat;
    }

    public abstract void bergerak();
    public abstract void bernapas();

    public void cetakInfo(){
        System.out.println("Berat\t: " + this.berat);
        System.out.println("Makanan\t: " + this.makanan);
        System.out.println("Habitat\t: " + this.habitat);
    }
}
```

2. Akan muncul error sebagai berikut

```
package abstractclass;

public class Hewan {
    public double berat;
    public String makanan;
    public String habitat;
}
```

Hewan is not abstract and does not override abstract method bernapas() in Hewan

(Alt-Enter shows hints)

3. Ubah class Hewan menjadi abstract Class. Jalankan program kemudian amati hasilnya.

```
1 package pemrogramanberbasisobjek.pertemuan10.abstractclass;
2
3 public abstract class Hewan {
4     public double berat;
5     public String makanan;
6     public String habitat;
7
8     public Hewan(double berat, String makanan, String habitat){
9         this.berat = berat;
10        this.makanan = makanan;
11        this.habitat = habitat;
12    }
13
14    public abstract void bergerak();
15    public abstract void bernapas();
16
17    public void cetakInfo(){
18        System.out.println("Berat\t: " + this.berat);
19        System.out.println("Makanan\t: " + this.makanan);
20        System.out.println("Habitat\t: " + this.habitat);
21    }
22 }
23
```

4. Ubah class demo sebagai berikut. Run program kemudian amati hasilnya

```
1 package pemrogramanberbasisobjek.pertemuan10.abstractclass;
2
3 public class AbstractClassDemo {
4     Run | Debug
5     public static void main(String[] args) {
6         Hewan hewan1 = new Hewan(10, "Rumput", "Savana");
7         hewan1.cetakInfo();
8         hewan1.bergerak();
9         hewan1.bernapas();
10    }
11 }
```

5. Ubah class demo sebagai berikut. Run program kemudian amati hasilnya

```
1 package pemrogramanberbasisobjek.pertemuan10.abstractclass;
2
3 public class AbstractClassDemo {
4     public static void main(String[] args) {
5
6         Lebah lebah1 = new Lebah(kasta: "Ratu", berat: 0.05, makanan: "Nektar", habitat: "Hutan");
7         lebah1.cetakInfo();
8         lebah1.bergerak();
9         lebah1.bernapas();
10    }
11 }
```

6. Klik icon lampu pada class Lebah, kemudian pilih option "Implement all abstract method"

```
1 package abstractclass;
2
3 public class Lebah extends Hewan{
4
5     public Lebah(String kasta, double berat, String makanan, String habitat) {
6         super(berat, makanan, habitat);
7         this.kasta = kasta;
8     }
9 }
10
```

7. Implementasi method bergerak dan bernapas pada class Lebah sebagai berikut. Run program kemudian amati hasilnya.

```
5
6 @Override
7 public void bernapas(){
8     System.out.println(x: "Otot perut mengundur, udara masuk melalui hidung di segmen tubuh");
9     System.out.println(x: "Trakea mengirimkan oksigen");
10    System.out.println(x: "Otot perut berkonstraksi, udara dikeluarkan");
11 }
12
13 @Override
14 public void bergerak(){
15     System.out.println(x: "Mendapatkan sayap ke depan");
16     System.out.println(x: "Memutar sayap hampir 90 derajat");
17     System.out.println(x: "Mengepakkan sayap ke belakang");
18 }
19
```

8. Tambahkan method cetakInfo() pada class Lebah. Run program kemudian amati hasilnya.

```
@Override
public void cetakInfo(){
    super.cetakInfo();
    System.out.println("Kasta\t: " + this.kasta);
}
```

9. Buat class Ular kemudian sebagai berikut. Instansiasi objek bertipe Ular pada class AbstractClassDemo. Eksekusi ketiga method untuk object tersebut.

```
1  package pemrogramanberbasisobjek.pertemuan10.abstractclass;
2
3  public class Ular extends Hewan {
4      public boolean isBerbisa;
5
6      public Ular(boolean isBerbisa, double berat, String makanan, String habitat){
7          super(berat, makanan, habitat);
8          this.isBerbisa = isBerbisa;
9      }
10
11     @Override
12     public void bergerak(){
13         System.out.println(x: "Mengerutkan Otot dari segala sisi hingga membentuk lengkungan");
14         System.out.println(x: "Menemukan titik penahanan seperti batu atau pohon");
15         System.out.println(x: "Menggunakan sisik untuk mendorong tubuh ke depan");
16     }
17
18     @Override
19     public void bernapas(){
20         System.out.println(x: "Otot tulang rusuk kontraksi, udara masuk lewat hidung ");
21         System.out.println(x: "Trakea mengirimkan udara ke paru-paru");
22         System.out.println(x: "Otot tulang rusuk relaksasi, udara dikeluarkan lewat hidung");
23     }
24
25     @Override
26     public void cetakInfo(){
27         super.cetakInfo();
28         System.out.println("Berbisa\t: " + ( this.isBerbisa ? "Ya" : "Tidak"));
29     }
30 }
31
```

D. PERTANYAAN

1. Pada langkah 1, mengapa sebaiknya method bergerak() dan bernapas() dideklarasikan sebagai abstract method?

Jawab : Karena belum diketahui secara pasti dan mungkin ada banyak perbedaan mengenai apa yang bisa dilakukan oleh sebuah class (tidak ada detail bagaimana cara melakukannya).

2. Mengapa pada langkah 2 muncul error?

Jawab : Karena kelas belum dideklarasikan sebagai abstract class, maka harus diberi keyword **abstract** untuk mendefinisikan abstract method.

3. Bagaimana hasil pada langkah 4? Mengapa demikian?

Jawab : Hasil akan error(**"Hewan is abstract; cannot be instantiated"**) karena abstract class merupakan class yang tidak dapat diinstansiasi (tidak dapat dibuat objectnya), umumnya abstract class digunakan sebagai generalisasi atau guideline dari subclass karena tidak ada detail bagaimana cara melakukannya.

4. Pada langkah 6-8, mengapa method bergerak() dan bernapas() **harus** di-override, namun method cetakInfo() **tidak harus** di-override?

Jawab : Karena method bergerak() dan bernapas() merupakan implementasi dari method abstract maka harus mengimplementasi semua abstract method dari parent class, jadi harus di-override, sedangkan untuk method cetakInfo() adalah method yang dapat didefinisikan pada superclass jadi tidak perlu diset sebagai abstract maka tidak harus mengimplementasi dari method abstract sehingga tidak harus di-override.

5. Dari percobaan tersebut, simpulkan kegunaan dari abstract class dan abstract method.

Jawab :

- **kegunaan dari abstract class :**

- Mencegah suatu class diinstansiasi atau dibuat objeknya
- Sebagai generalisasi/guideline pada class hierarki Method yang tidak dapat didefinisikan pada superclass
- Memaksa subclass untuk mengimplementasi seluruh abstract method tersebut.

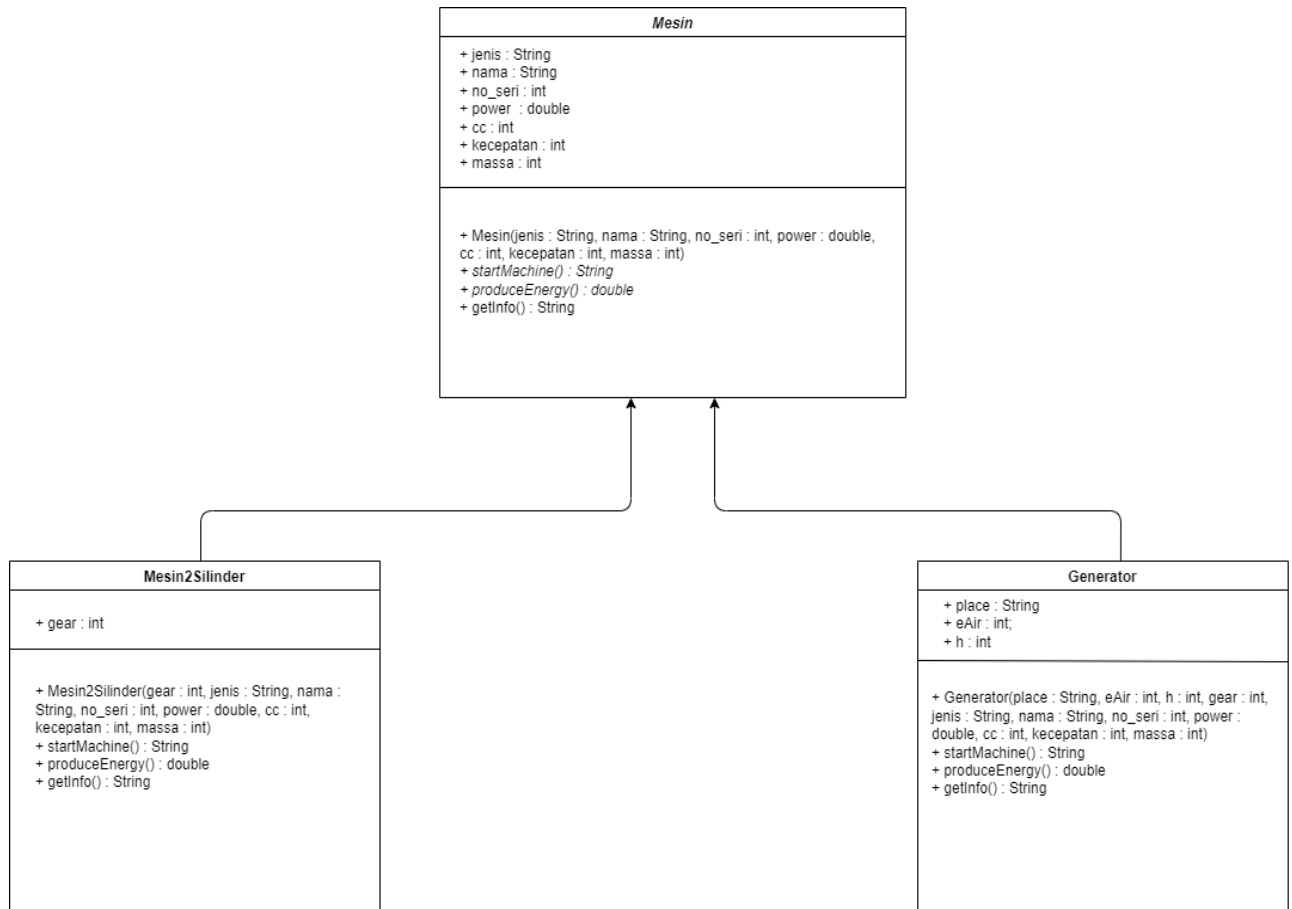
- **kegunaan dari abstract method :**

Memastikan class turunannya akan membuat kerangka yang sama dengan class parentnya.

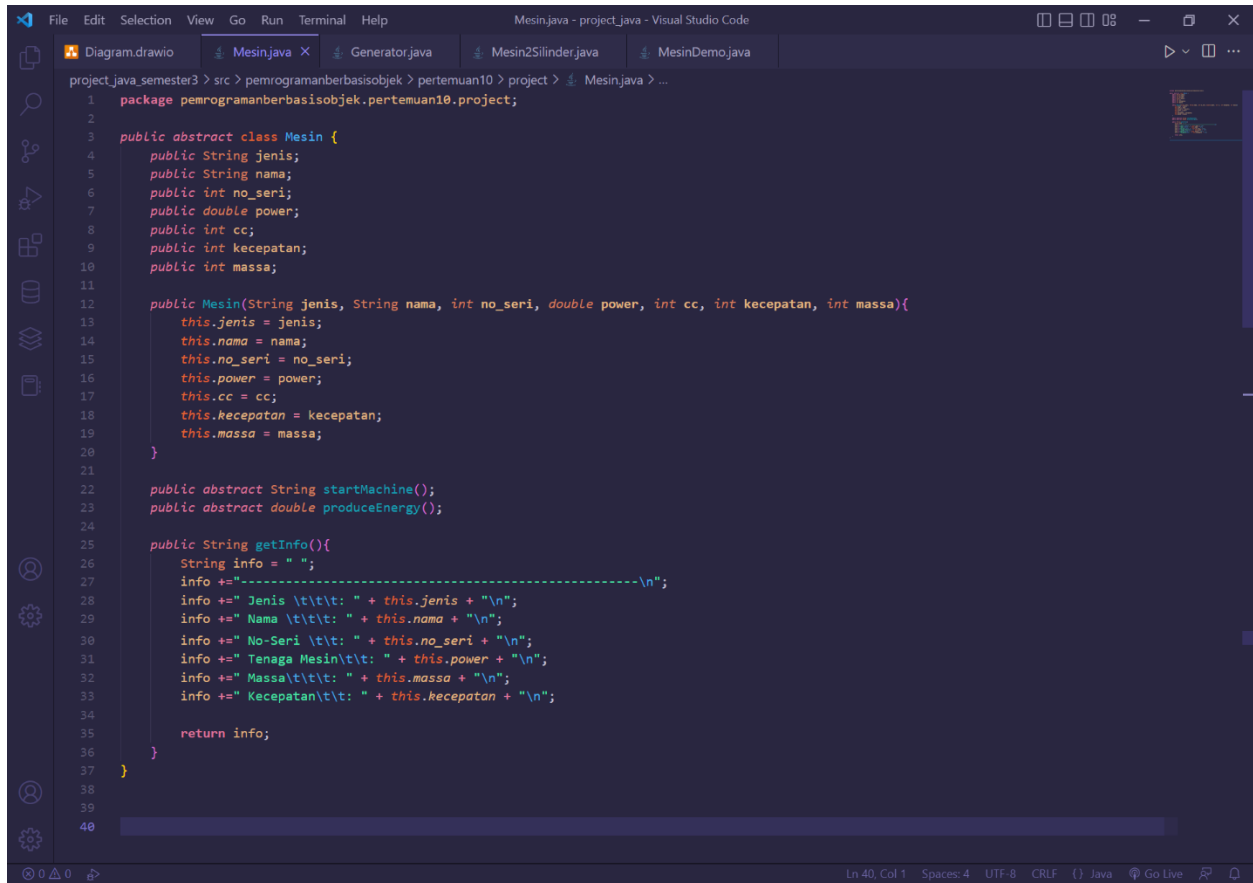
4. TUGAS

1. Implementasikan class diagram yang telah dirancang pada tugas PBO Teori ke dalam kode program.
Selanjutnya buatlah instansiasi objek dari masing-masing subclass kemudian coba eksekusi method-method yang dimiliki.

Revisi Diagram class



Code class Mesin



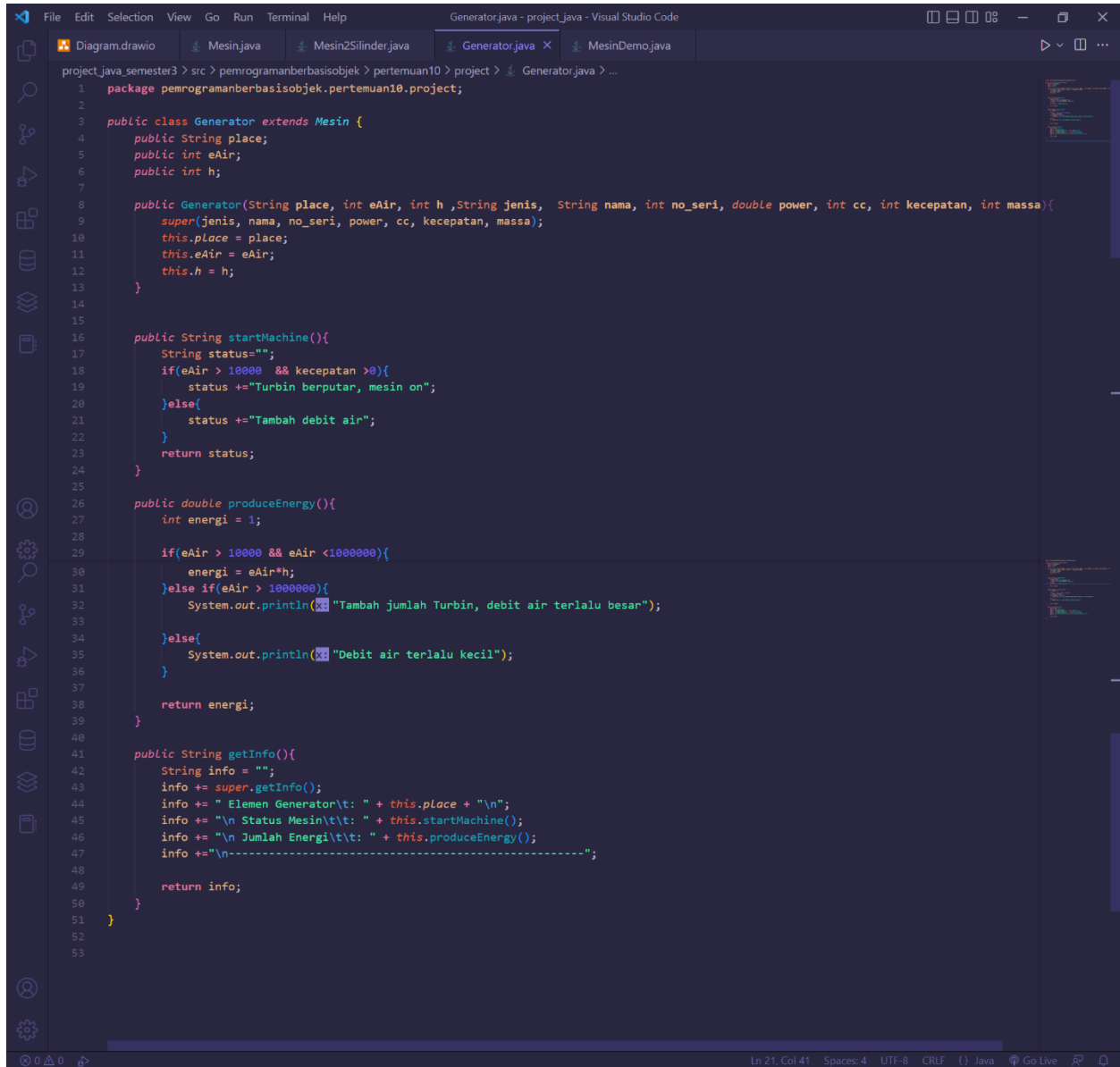
```
File Edit Selection View Go Run Terminal Help
Mesin.java - project_java - Visual Studio Code
Diagram.drawio Mesin.java Generator.java Mesin2Silinder.java MesinDemo.java
project_java_semester3 > src > pemrogramanberbasisobjek > pertemuan10 > project > Mesin.java > ...
1 package pemrogramanberbasisobjek.pertemuan10.project;
2
3 public abstract class Mesin {
4     public String jenis;
5     public String nama;
6     public int no_seri;
7     public double power;
8     public int cc;
9     public int kecepatan;
10    public int massa;
11
12    public Mesin(String jenis, String nama, int no_seri, double power, int cc, int kecepatan, int massa){
13        this.jenis = jenis;
14        this.nama = nama;
15        this.no_seri = no_seri;
16        this.power = power;
17        this.cc = cc;
18        this.kecepatan = kecepatan;
19        this.massa = massa;
20    }
21
22    public abstract String startMachine();
23    public abstract double produceEnergy();
24
25    public String getInfo(){
26        String info = " ";
27        info += "-----\n";
28        info += "Jenis \t\t\t: " + this.jenis + "\n";
29        info += "Nama \t\t\t: " + this.nama + "\n";
30        info += "No-Seri \t\t: " + this.no_seri + "\n";
31        info += "Tenaga Mesin\t\t: " + this.power + "\n";
32        info += "Massa\t\t\t: " + this.massa + "\n";
33        info += "Kecepatan\t\t: " + this.kecepatan + "\n";
34
35        return info;
36    }
37 }
38
39
40
```

Ln 40, Col 1 Spaces: 4 UTF-8 CRLF () Java Go Live

Code class Mesin2Silinder

```
File Edit Selection View Go Run Terminal Help Mesin2Silinder.java - project_Java - Visual Studio Code
Diagram.drawio Mesin.java Generator.java Mesin2Silinder.java X MesinDemo.java
project_Java_semester3 > src > pemrogramanberbasisobjek > pertemuan10 > project > Mesin2Silinder.java > Mesin2Silinder > getInfo()
1 package pemrogramanberbasisobjek.pertemuan10.project;
2
3 public class Mesin2Silinder extends Mesin {
4     public int gear;
5
6     public Mesin2Silinder(int gear, String jenis, String nama, int no_seri, double power, int cc, int kecepatan, int massa){
7         super(jenis, nama, no_seri, power, cc, kecepatan, massa);
8         this.gear = gear;
9     }
10
11     public String startMachine(){
12         String status="";
13         if(gear >= 0 && gear < 7){
14             if(gear == 0 && kecepatan == 0){
15                 status += "Mesin mati, Silahkan putar kunci pada posisi on!";
16             }
17             else if(gear > 0 && kecepatan ==0){
18                 status += "Mesin On";
19             }else if(gear > 0 && kecepatan !=0){
20                 status += "Mesin On, Bergerak";
21             }
22         }else{
23             status += "Periksa kembali kondisi motor";
24         }
25         return status;
26     }
27
28     public double produceEnergy(){
29         //Ek = 1/2*m*v*v
30         double energi = 1;
31         if(gear > 0 && gear < 7){
32             energi = (0.5)*kecepatan*kecepatan*massa;
33         }else{
34             return 0;
35         }
36         return energi;
37     }
38
39     public String getInfo(){
40         String info = "";
41         info += super.getInfo();
42         info += "\n Status Mesin\t\t: " + this.startMachine();
43         info += "\n Jumlah Energi\t\t: " + this.produceEnergy();
44         info += "\n Gear\t\t\t\t: " + this.gear + "\n";
45         info += "-----\n\n";
46         return info;
47     }
48 }
49
50
```

Code class Generator



```
1 package pemrogramanberbasisobjek.pertemuan10.project;
2
3 public class Generator extends Mesin {
4     public String place;
5     public int eAir;
6     public int h;
7
8     public Generator(String place, int eAir, int h, String jenis, String nama, int no_seri, double power, int cc, int kecepatan, int massa){
9         super(jenis, nama, no_seri, power, cc, kecepatan, massa);
10        this.place = place;
11        this.eAir = eAir;
12        this.h = h;
13    }
14
15
16    public String startMachine(){
17        String status="";
18        if(eAir > 10000 && kecepatan >0){
19            status += "Turbin berputar, mesin on";
20        }else{
21            status += "Tambah debit air";
22        }
23        return status;
24    }
25
26    public double produceEnergy(){
27        int energi = 1;
28
29        if(eAir > 10000 && eAir <1000000){
30            energi = eAir*h;
31        }else if(eAir > 1000000){
32            System.out.println("Tambah jumlah Turbin, debit air terlalu besar");
33        }else{
34            System.out.println("Debit air terlalu kecil");
35        }
36
37        return energi;
38    }
39
40
41    public String getInfo(){
42        String info = "";
43        info += super.getInfo();
44        info += " Elemen Generator\t: " + this.place + "\n";
45        info += "\n Status Mesin\t\t: " + this.startMachine();
46        info += "\n Jumlah Energi\t\t: " + this.produceEnergy();
47        info += "\n-----";
48
49        return info;
50    }
51 }
52
53
```

Class MesinDemo

```

File Edit Selection View Go Run Terminal Help
MesinDemo.java - project_java - Visual Studio Code
Diagram.drawio Mesin.java Mesin2Silinder.java Generator.java MesinDemo.java X
project_java.semester3 > src > pemrogramanberbasisobjek > pertemuan10 > project > MesinDemo.java > ...
1 package pemrogramanberbasisobjek.pertemuan10.project;
2
3 public class MesinDemo {
    Run | Debug
4     public static void main(String[] args) {
5         Mesin2Silinder mesin1 = new Mesin2Silinder(gear: 3, jenis: "Motor2Silinder", nama: "Rx-King", no_seri: 21109, power: 9.9, cc: 135,
6             kecepatan: 60, massa: 120);
7         System.out.println(mesin1.getInfo());
8
9         System.out.println("\t\t\t\t\t");
10
11        Generator generator1 = new Generator(place: "Air", aAir: 100000, h: 30, jenis: "Generator", nama: "Generator-DC", no_seri: 405978, power: 9999,
12            cc: 100000, kecepatan: 10000, massa: 1000);
13        System.out.println(generator1.getInfo());
14    }
15 }

```

Hasil running

```
File Edit Selection View Go Run Terminal Help
MesinDemo.java - project.java - Visual Studio Code

PROBLEMS DEBUG CONSOLE TERMINAL JUPYTER
Run: MesinDemo +v [icon] [icon] [icon] [icon] [icon]

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\project_java> & 'C:\Program Files\Java\jdk-18.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zonal\AppData\Roaming\Code\User\workspacesStorage\0ae86845b6229fb21af10af863b2965a\redhat.java\jdt_ws\project_java_c923881\bin' 'pemrogramanberbasisobjek.pertemuan10.project.MesinDemo'

-----
Jenis          : Motor2Sylinder
Nama           : Rx-King
No-Seri        : 21109
Tenaga Mesin   : 9.9
Massa          : 120
Kecepatan      : 60

Status Mesin   : Mesin On, Bergerak
Jumlah Energi  : 216000.0
Gear           : 3
-----

-----
Jenis          : Generator
Nama           : Generator-DC
No-Seri        : 405978
Tenaga Mesin   : 9999.0
Massa          : 1000
Kecepatan      : 10000
Elemen Generator : Air

Status Mesin   : Turbin berputar, mesin on
Jumlah Energi  : 3000000.0
-----

PS D:\project_java>
```