

Object Oriented Programming

Jobsheet 4

Class Relation



Name: Azahra Salsabila

NIM: 2241720073

Class: 2I

No: 05

Major: Information Technology

Study Program: Informatic Engineering

Experiment 1

Screenshot program code

```
1  package semester3.jobsheet4.zara.relasticlass.percobaan1;
2  public class Processor {
3      private String merk;
4      private double cache;
5
6      public Processor() {
7      }
8
9      public Processor(String merk, double cache) {
10         this.merk = merk;
11         this.cache = cache;
12     }
13
14     public void setMerk(String merk) {
15         this.merk = merk;
16     }
17
18     public void setCache(double cache) {
19         this.cache = cache;
20     }
21
22     public String getMerk() {
23         return merk;
24     }
25
26     public double getCache() {
27         return cache;
28     }
29
30     public void info() {
31         System.out.printf(format:"Merk Processor = %s\n", merk);
32         System.out.printf(format:"Cache Memory = %.2f\n", cache);
33     }
34 }
```

```
1  package semester3.jobsheet4.zara.relasticlass.percobaan1;
2  public class Laptop {
3      private String merk;
4      private Processor proc;
5
6      public Laptop() {
7      }
8
9      public Laptop(String merk, Processor proc) {
10         this.merk = merk;
11         this.proc = proc;
12     }
13
14     public void info() {
15         System.out.println("Merk Laptop = " + merk);
16         proc.info();
17     }
18
19     public void setMerk(String merk) {
20         this.merk = merk;
21     }
22
23     public void setProc(Processor proc) {
24         this.proc = proc;
25     }
26
27     public String getMerk() {
28         return merk;
29     }
30
31     public Processor getProc() {
32         return proc;
33     }
34 }
```

```

1  package semester3.jobsheet4.zara.relasticlass.percobaan1;
2  public class MainPercobaan1 {
    Run | Debug
3      public static void main(String[] args) {
4          Processor p = new Processor(merk:"Intel i5", cache:3);
5          Laptop l = new Laptop(merk:"Thinkpad", p);
6
7          l.info();
8
9          Processor p1 = new Processor();
10         p1.setMerk(merk:"Intel i5");
11         p1.setCache(cache:4);
12         Laptop L1 = new Laptop();
13         L1.setMerk(merk:"Thinkpad");
14         L1.setProc(p1);
15         L1.info();
16     }
17 }

```

Screenshot run

```

PS D:\terserah\terserah> &
in\java.exe' '-cp' 'D:\ters
4.zara.relasticlass.percobaan
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3.00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00

```

Question

1. Di dalam class Processor dan class Laptop , terdapat method setter dan getter untuk masing-masing atributnya. Apakah gunanya method setter dan getter tersebut?
 - Method setter (mutator) digunakan untuk mengatur (set) nilai atribut suatu objek, sedangkan method getter (accessor) digunakan untuk mendapatkan (get) nilai atribut dari suatu objek. Mereka memungkinkan kita untuk mengendalikan akses ke atribut objek dan menerapkan enkapsulasi. Dengan setter, Anda dapat mengatur nilai atribut, dan dengan getter, Anda dapat mendapatkan nilai atribut.
2. Di dalam class Processor dan class Laptop, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut?
 - Konstruktor Default: Konstruktor default adalah konstruktor yang tidak memiliki parameter. Ia digunakan untuk membuat objek dengan nilai awal default atau kosong untuk semua atribut objek.
 - Konstruktor Berparameter: Konstruktor berparameter adalah konstruktor yang memiliki parameter untuk menginisialisasi atribut objek sesuai dengan nilai yang diberikan saat objek dibuat. Ini memungkinkan kita untuk mengatur nilai awal objek sesuai dengan kebutuhan.

3. Perhatikan class Laptop, di antara 2 atribut yang dimiliki (merk dan proc), atribut manakah yang bertipe object?
 - Pada class Laptop, atribut 'proc' yang bertipe 'Processor' adalah yang bertipe objek. Atribut 'merk' adalah bertipe data String.
4. Perhatikan class Laptop, pada baris manakah yang menunjukkan bahwa class Laptop memiliki relasi dengan class Processor?
 - Relasi dengan class Processor terlihat pada baris 'private Processor proc;' di dalam class Laptop. Ini menunjukkan bahwa class Laptop memiliki atribut objek yang merupakan instance dari class Processor, sehingga ada hubungan (relasi) antara keduanya.
5. Perhatikan pada class Laptop, Apakah guna dari sintaks proc.info()?
 - Sintaks 'proc.info()' digunakan untuk memanggil 'method info()' dari objek 'proc (Processor)'. Ini memungkinkan untuk mencetak informasi tentang processor yang terpasang pada laptop.
6. Pada class MainPercobaan1, terdapat baris kode: Laptop l = new Laptop("Thinkpad", p);. Apakah p tersebut ? Dan apakah yang terjadi jika baris kode tersebut diubah menjadi: Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3)); Bagaimanakah hasil program saat dijalankan, apakah ada perubahan?
 - Variabel p adalah objek dari class Processor yang digunakan untuk menginisialisasi objek l dari class Laptop. Dalam baris Laptop l = new Laptop("Thinkpad", p);, laptop l akan memiliki processor yang sama dengan processor p.
 - Jika baris kode diubah menjadi Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));, maka objek l akan memiliki processor baru dengan merk "Intel i5" dan cache 3. Jadi, hasil program akan mencetak informasi tentang laptop l dengan processor baru yang diberikan dalam konstruktor, yang berbeda dari yang pertama.

Experiment 2

Screenshot program code

```
1 package semester3.jobsheet4.zara.relasticlass.percobaan2;
2 public class Mobil {
3     private String merk;
4     private int biaya;
5
6     public Mobil() {
7     }
8
9     public Mobil(String merk, int biaya) {
10         this.merk = merk;
11         this.biaya = biaya;
12     }
13
14     public void setMerk(String merk) {
15         this.merk = merk;
16     }
17
18     public void setBiaya(int biaya) {
19         this.biaya = biaya;
20     }
21
22     public String getMerk() {
23         return merk;
24     }
25
26     public int biaya() {
27         return biaya;
28     }
29
30     public int hitungBiayaMobil(int hari) {
31         return biaya * hari;
32     }
33 }
```

```
1  package semester3.jobsheet4.zara.relasticlass.percobaan2;
2  public class Sopir {
3      private String nama;
4      private int biaya;
5
6      public Sopir() {
7      }
8
9      public Sopir(String nama, int biaya) {
10         this.nama = nama;
11         this.biaya = biaya;
12     }
13
14     public void setNama(String nama) {
15         this.nama = nama;
16     }
17
18     public void setBiaya(int biaya) {
19         this.biaya = biaya;
20     }
21
22     public String getNama() {
23         return nama;
24     }
25
26     public int getBiaya() {
27         return biaya;
28     }
29
30     public int hitungBiayaSopir(int hari) {
31         return biaya * hari;
32     }
33 }
```



```
1 package semester3.jobsheet4.zara.relasiclass.percobaan2;
2 public class Pelanggan {
3     private String nama;
4     private Mobil mobil;
5     private Sopir sopir;
6     private int hari;
7
8     public Pelanggan() {
9     }
10
11     public Pelanggan(String nama, Mobil mobil, Sopir sopir, int hari) {
12         this.nama = nama;
13         this.mobil = mobil;
14         this.sopir = sopir;
15         this.hari = hari;
16     }
17
18     public void setNama(String nama) {
19         this.nama = nama;
20     }
21
22     public void setMobil(Mobil mobil) {
23         this.mobil = mobil;
24     }
25
26     public void setSopir(Sopir sopir) {
27         this.sopir = sopir;
28     }
29
30     public void setHari(int hari) {
31         this.hari = hari;
32     }
33
34     public String getNama() {
35         return nama;
36     }
37
38     public Mobil getMobil() {
39         return mobil;
40     }
41
42     public Sopir getSopir() {
43         return sopir;
44     }
45
46     public int getHari() {
47         return hari;
48     }
49
50     public int hitungBiayaTotal() {
51         return mobil.hitungBiayaMobil(hari) + sopir.hitungBiayaSopir(hari);
52     }
53 }
54
```



```

1  package semester3.jobsheet4.zara.relasiclass.percobaan2;
2  public class MainPercobaan2 {
    Run | Debug
3      public static void main(String[] args) {
4          Mobil m = new Mobil();
5          m.setMerk(merk:"Avanza");
6          m.setBiaya(biaya:350000);
7
8          Sopir s = new Sopir();
9          s.setNama(nama:"John Doe");
10         s.setBiaya(biaya:200000);
11
12         Pelanggan p = new Pelanggan();
13         p.setNama(nama:"Jane Doe");
14         p.setMobil(m);
15         p.setSopir(s);
16         p.setHari(hari:2);
17         System.out.println("Biaya Total = " + p.hitungBiayaTotal());
18     }
19 }

```

Screenshot run

```

PS D:\terserah\terserah>
in\java.exe' '-cp' 'D:\t
4.zara.relasiclass.percobaan2
Biaya Total = 1100000

```

Question

- Perhatikan class Pelanggan. Pada baris program manakah yang menunjukkan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir?
 - Relasi dengan class Mobil terdapat pada baris private Mobil mobil;, dan relasi dengan class Sopir terdapat pada baris private Sopir sopir;. Ini menunjukkan bahwa class Pelanggan memiliki atribut objek yang merupakan instance dari class Mobil dan class Sopir, sehingga ada relasi antara Pelanggan, Mobil, dan Sopir.
- Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari?
 - Argument "hari" digunakan dalam method hitungBiayaSopir pada class Sopir dan method hitungBiayaMobil pada class Mobil untuk menghitung biaya berdasarkan jumlah hari sewa. Ini memungkinkan perhitungan biaya yang berbeda tergantung pada berapa lama mobil atau sopir disewa.
- Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil (hari) dan sopir. hitungBiayaSopir (hari)?
 - Perintah-perintah ini digunakan untuk menghitung biaya total sewa mobil dan sopir oleh pelanggan berdasarkan jumlah hari yang diset dalam objek Pelanggan. Dengan kata lain, perintah ini menggabungkan perhitungan biaya

dari class Mobil dan class Sopir sesuai dengan jumlah hari yang diatur di objek Pelanggan.

4. Perhatikan class Main Percobaan2. Untuk apakah sintaks `p.setMobil (m)` dan `p.setSopir (s)`?
 - Sintaks ini digunakan untuk menghubungkan objek Mobil `m` dan objek Sopir `s` dengan objek Pelanggan `p`. Ini mengatur mobil dan sopir yang akan digunakan oleh pelanggan dalam sewaanannya.
5. Perhatikan class Main Percobaan2. Untuk apakah proses `p.hitungBiayaTotal ()` tersebut?
 - Proses ini menghitung biaya total sewa yang mencakup biaya mobil dan biaya sopir. Ini memanggil method `hitungBiayaTotal` pada objek Pelanggan `p`, yang menggabungkan perhitungan biaya mobil dan biaya sopir sesuai dengan jumlah hari yang telah ditentukan.
6. Perhatikan class Main Percobaan2, coba tambahkan pada baris terakhir dari method `main` dan amati perubahan saat di-run!
`System.out.println(p.getMobil().getMerk ());`
Jadi untuk apakah sintaks `p.getMobil().getMerk ()` yang ada di dalam method `main` tersebut?
 - Sintaks ini digunakan untuk mendapatkan merk mobil yang digunakan oleh pelanggan. Itu mengambil objek mobil yang telah diatur dalam objek pelanggan dan kemudian mengakses merk mobil dari objek tersebut. Hasilnya adalah merk mobil yang akan dicetak menggunakan `System.out.println`.

Experiment 3

Screenshot program code

```
1  package semester3.jobsheet4.zara.relasticlass.percobaan3;
2  public class Pegawai {
3      private String nip;
4      private String nama;
5
6      public Pegawai() {
7      }
8
9      public Pegawai(String nip, String nama) {
10         this.nip = nip;
11         this.nama = nama;
12     }
13
14     public void setNip(String nip) {
15         this.nip = nip;
16     }
17
18     public void setName(String nama) {
19         this.nama = nama;
20     }
21
22     public String getNip() {
23         return nip;
24     }
25
26     public String getName() {
27         return nama;
28     }
29
30     public String info() {
31         String info = "";
32         info += "Nip: " + this.nip + "\n";
33         info += "Nama: " + this.nama + "\n";
34         return info;
35     }
36 }
```

```
1 package semester3.jobsheet4.zara.relasticlass.percobaan3;
2 public class KeretaApi {
3     private String nama;
4     private String kelas;
5     private Pegawai masinis;
6     private Pegawai asisten;
7
8     public KeretaApi() {
9     }
10
11     public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten) {
12         this.nama = nama;
13         this.kelas = kelas;
14         this.masinis = masinis;
15         this.asisten = asisten;
16     }
17
18     public void setName(String nama) {
19         this.nama = nama;
20     }
21
22     public void setKelas(String kelas) {
23         this.kelas = kelas;
24     }
25
26     public void setMasinis(Pegawai masinis) {
27         this.masinis = masinis;
28     }
29
30     public void setAsisten(Pegawai asisten) {
31         this.asisten = asisten;
32     }
33
34     public String getName() {
35         return nama;
36     }
37
38     public String getKelas() {
39         return kelas;
40     }
41
42     public Pegawai getMasinis() {
43         return masinis;
44     }
45
46     public Pegawai getAsisten() {
47         return asisten;
48     }
49
50     public String info() {
51         String info = "";
52         info += "Nama: " + this.nama + "\n";
53         info += "Kelas: " + this.kelas + "\n";
54         info += "Masinis: \n" + this.masinis.info() + "\n";
55         info += "Asisten: \n" + this.asisten.info() + "\n";
56         return info;
57     }
58 }
59
```

```

1 package semester3.jobsheet4.zara.relasticlass.percobaan3;
2 public class MainPercobaan3 {
3     public static void main(String[] args) {
4         Pegawai masinis = new Pegawai(nip:"1234", nama:"Spongebob Squarepants");
5         Pegawai asisten = new Pegawai(nip:"4567", nama:"Patrick Star");
6         KeretaApi keretaApi = new KeretaApi(nama:"Gaya Baru", kelas:"Ekonomi", masinis, asisten);
7
8         System.out.println(keretaApi.info());
9     }
10 }

```

Screenshot run

```

Nama: Gaya Baru
Kelas: Ekonomi
Masinis:
Nip: 1234
Nama: Spongebob Squarepants

Asisten:
Nip: 4567
Nama: Patrick Star

```

Question

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa?
 - Baris this.masinis.info() dan this.asisten.info() dalam method info() pada class KeretaApi digunakan untuk memanggil method info() pada objek masinis dan asisten, yang merupakan objek dari class Pegawai. Ini memungkinkan untuk mencetak informasi tentang masinis dan asisten yang bekerja pada kereta api tersebut.
2. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main()!


```

Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis);
System.out.println(keretaApi.info());

```

```

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The constructor KeretaApi(String, String, Pegawai) is undefined

    at semester3.jobsheet4.zara.relasticlass.percobaan3.MainPertanyaan.main(MainPertanyaan.java:5)

```

3. Apa hasil output dari main program tersebut? Mengapa hal tersebut dapat terjadi?
 - Output dari main program tersebut akan menghasilkan kesalahan kompilasi. Hal ini terjadi karena constructor KeretaApi yang digunakan dalam main program hanya menerima dua parameter (nama dan kelas), sedangkan di main program hanya ada satu objek masinis. Namun, constructor KeretaApi seharusnya menerima tiga parameter (nama, kelas, masinis, dan asisten). Oleh karena itu, terjadi kesalahan saat mencoba membuat objek KeretaApi.
4. Perbaiki class KeretaApi sehingga program dapat berjalan!

Nama: Gaya Baru
Kelas: Ekonomi
Masinis:
Nip: 1234
Nama: Spongebob Squarepants

Asisten:
Nip: 4567
Nama: Patrick Star



```
public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten) {  
    this.nama = nama;  
    this.kelas = kelas;  
    this.masinis = masinis;  
    this.asisten = asisten;  
}
```

Experiment 4

Screenshot program code

```
1  package semester3.jobsheet4.zara.relasticlass.percobaan4;
2  public class Penumpang {
3      private String ktp;
4      private String nama;
5
6      public Penumpang() {
7      }
8
9      public Penumpang(String ktp, String nama) {
10         this.ktp = ktp;
11         this.nama = nama;
12     }
13
14     public void setKtp(String ktp) {
15         this.ktp = ktp;
16     }
17
18     public String getKtp() {
19         return ktp;
20     }
21
22     public void setName(String nama) {
23         this.nama = nama;
24     }
25
26     public String getName() {
27         return nama;
28     }
29
30     public String info() {
31         String info = "";
32         info += "Ktp: " + ktp + "\n";
33         info += "Nama: " + nama + "\n";
34         return info;
35     }
36 }
```

```
1 package semester3.jobsheet4.zara.relasticlass.percobaan4;
2 public class Kursi {
3     private String nomor;
4     private Penumpang penumpang;
5
6     public Kursi(String nomor) {
7         this.nomor = nomor;
8     }
9
10    public void setNomor(String nomor) {
11        this.nomor = nomor;
12    }
13
14    public void setPenumpang(Penumpang penumpang) {
15        this.penumpang = penumpang;
16    }
17
18    public String getNomor() {
19        return nomor;
20    }
21
22    public Penumpang getPenumpang() {
23        return penumpang;
24    }
25
26    public String info() {
27        String info = "";
28        info += "Nomor: " + nomor + "\n";
29        if (this.penumpang != null) {
30            info += "Penumpang: " + penumpang.info() + "\n";
31        }
32        return info;
33    }
34 }
```



```
1 package semester3.jobsheet4.zara.relasticlass.percobaan4;
2 public class Gerbong {
3     private String kode;
4     private Kursi[] arrayKursi;
5
6     public Gerbong(String kode, int jumlah) {
7         this.kode = kode;
8         this.arrayKursi = new Kursi[jumlah];
9         this.initKursi();
10    }
11
12    private void initKursi() {
13        for (int i = 0; i < arrayKursi.length; i++) {
14            this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));
15        }
16    }
17
18    public void setKode(String kode) {
19        this.kode = kode;
20    }
21
22    public void setPenumpang(Penumpang penumpang, int nomor) {
23        if (this.arrayKursi[nomor - 1].getPenumpang() != null) {
24            System.out.println("Kursi sudah terisi, silahkan pilih kursi lain");
25        } else {
26            this.arrayKursi[nomor - 1].setPenumpang(penumpang);
27        }
28    }
29
30    public String getKode() {
31        return kode;
32    }
33
34    public Kursi[] getArrayKursi() {
35        return arrayKursi;
36    }
37
38    public String info() {
39        String info = "";
40        info += "Kode: " + this.kode + "\n";
41        for (Kursi kursi : arrayKursi) {
42            info += kursi.info();
43        }
44        return info;
45    }
46 }
47
```

```

1  package semester3.jobsheet4.zara.relasiclass.percobaan4;
2  public class MainPercobaan4 {
    Run | Debug
3      public static void main(String[] args) {
4          Penumpang p = new Penumpang(ktp:"12345", nama:"Mr.Krab");
5          Gerbong gerbong = new Gerbong(kode:"A", jumlah:10);
6          gerbong.setPenumpang(p, nomor:1);
7          System.out.println(gerbong.info());
8      }
9  }

```

Screenshot run

```

Kode: A
Nomor: 1
Penumpang: Ktp: 12345
Nama: Mr.Krab

Nomor: 2
Nomor: 3
Nomor: 4
Nomor: 5
Nomor: 6
Nomor: 7
Nomor: 8
Nomor: 9
Nomor: 10

```

Question

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A?
 - Jumlah kursi dalam Gerbong A adalah 10, karena saat objek Gerbong gerbong dibuat dalam main program, ia diinisialisasi dengan 10 kursi.
2. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ? ... if (this.penumpang != null) { info += "Penumpang: " + penumpang.info() + "\n"; } ...
 - Kode dalam method info() pada class Kursi digunakan untuk memeriksa apakah kursi memiliki penumpang atau tidak. Jika penumpang (objek penumpang) dalam kursi tidak null (artinya ada penumpang), maka informasi tentang penumpang tersebut akan ditambahkan ke string info. Ini memungkinkan untuk mencetak informasi penumpang yang menduduki kursi, jika ada penumpang di kursi tersebut.
3. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1?
 - Nilai nomor dikurangi 1 pada method setPenumpang() dalam class Gerbong karena umumnya dalam pemrograman, indeks array dimulai dari 0. Jadi, kursi pertama memiliki nomor 1 dalam input dari pengguna, tetapi untuk mengakses indeks array, perlu dikurangi 1.
4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(budi, 1). Apakah yang terjadi?

- Jika Anda melakukan `gerbong.setPenumpang(budi, 1)` setelah mengatur penumpang "Mr.Krab" pada kursi 1 dalam main program, maka Anda akan mendapatkan output "Kursi sudah terisi, silahkan pilih kursi lain." Karena kursi 1 sudah terisi oleh "Mr.Krab", dan Anda mencoba untuk menempatkan "Budi" di kursi yang sudah terisi.
- 5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain!
 - Untuk tidak mengizinkan penumpang lain menduduki kursi yang sudah ada penumpang lain, Anda perlu memodifikasi method `setPenumpang()` dalam class `Gerbong`. Anda dapat menambahkan pengecekan apakah kursi sudah terisi sebelum menambahkan penumpang baru. Jika kursi sudah terisi, Anda dapat mengeluarkan pesan kesalahan atau melakukan tindakan lain sesuai dengan aturan yang Anda inginkan.

Assignment

1. Berdasarkan latihan di pertemuan teori, rancang dengan class diagram, kemudian implementasikan ke dalam program! Studi kasus harus mewakili relasi class dari percobaan-percobaan yang telah dilakukan pada materi ini, setidaknya melibatkan minimal 4 class (class yang berisi main tidak dihitung)
 - Screenshot program code

```
1  package semester3.jobsheet4.zara.relasiclass.assignment;
2  public class Dosen {
3      private String nip;
4      private String nama;
5
6      public Dosen(String nip, String nama) {
7          this.nip = nip;
8          this.nama = nama;
9      }
10
11     public String getNip() {
12         return nip;
13     }
14
15     public String getNama() {
16         return nama;
17     }
18 }
```

```
1  package semester3.jobsheet4.zara.relasticlass.assignment;
2  public class MataKuliah {
3      private String kode;
4      private String nama;
5      private int sks;
6
7      public MataKuliah(String kode, String nama, int sks) {
8          this.kode = kode;
9          this.nama = nama;
10         this.sks = sks;
11     }
12
13     public String getKode() {
14         return kode;
15     }
16
17     public String getNama() {
18         return nama;
19     }
20
21     public int getsks() {
22         return sks;
23     }
24 }
```

```
1 package semester3.jobsheet4.zara.relasicclass.assignment;
2 public class Mahasiswa {
3     private String nim;
4     private String nama;
5     private String alamat;
6     private Dosen dosenWali;
7
8     public Mahasiswa(String nim, String nama, String alamat, Dosen dosenWali) {
9         this.nim = nim;
10        this.nama = nama;
11        this.alamat = alamat;
12        this.dosenWali = dosenWali;
13    }
14
15    public String getNim() {
16        return nim;
17    }
18
19    public String getNama() {
20        return nama;
21    }
22
23    public String getAlamat() {
24        return alamat;
25    }
26
27    public Dosen getDosenWali() {
28        return dosenWali;
29    }
30 }
```

```

1  package semester3.jobsheet4.zara.relasticlass.assignment;
2  public class Nilai {
3      private Mahasiswa mahasiswa;
4      private MataKuliah mataKuliah;
5      private double nilai;
6
7      public Nilai(Mahasiswa mahasiswa, MataKuliah mataKuliah, double nilai) {
8          this.mahasiswa = mahasiswa;
9          this.mataKuliah = mataKuliah;
10         this.nilai = nilai;
11     }
12
13     public Mahasiswa getMahasiswa() {
14         return mahasiswa;
15     }
16
17     public MataKuliah getMataKuliah() {
18         return mataKuliah;
19     }
20
21     public double getNilai() {
22         return nilai;
23     }
24 }

```

```

1  package semester3.jobsheet4.zara.relasticlass.assignment;
2  public class MainProgram {
3      Run | Debug
4      public static void main(String[] args) {
5          Dosen dosenWali = new Dosen(nip:"1234", nama:"Dr. John Doe");
6          Mahasiswa mahasiswa = new Mahasiswa(nim:"20210001", nama:"Alice", alamat:"Jl. ABC No. 123", dosenWali);
7          MataKuliah mataKuliah = new MataKuliah(kode:"MATH101", nama:"Matematika Dasar", sks:3);
8          Nilai nilai = new Nilai(mahasiswa, mataKuliah, nilai:85.5);
9
10         System.out.println("Nama Mahasiswa: " + mahasiswa.getNama());
11         System.out.println("Dosen Wali: " + mahasiswa.getDosenWali().getNama());
12         System.out.println("Mata Kuliah: " + mataKuliah.getNama());
13         System.out.println("Nilai: " + nilai.getNilai());
14     }

```

➤ Screenshot run java

```

Nama Mahasiswa: Alice
Dosen Wali: Dr. John Doe
Mata Kuliah: Matematika Dasar
Nilai: 85.5

```