

Q1

A

- Aturan gaji total:

Pegawai

nama: String
masaKerjaThn: int
jamLembur: double
gajiPokok: int

hitungGajiLembur(): double
hitungTunjangan(): double
hitungGajiTotal(): double
printData(): void

- Hasil penambahan gaji pokok, gaji lembur dan tunjangan
- Gaji lembur ditentukan berdasar masa kerja:
- Masa kerja < 5 thn: gaji lembur per jam 5% dari gaji pokok
- Masa kerja >5 thn: gaji lembur per jam 10% dari gaji pokok
- Tunjangan ditentukan berdasar masa kerja:
- Masa kerja < 5 thn: tunjangan 25% dari gaji pokok
- Masa kerja >5 thn: tunjangan 50% dari gaji pokok
- Buat class **Pegawai** sesuai dengan

diagram class di atas!

- Buat class/method **main** yang di dalamnya terdapat koleksi data Pegawai sebanyak **n** data
- Selanjutnya buat program untuk menghitung gaji total rata-rata dari pegawai dengan masa kerja > 5 thn
- Selanjutnya buat program untuk mencari gaji lembur terbesar dari pegawai dengan msaa kerja < 5 thn. Dan tampilkan data pegawai dengan gaji lembur terkecil tersebut

B

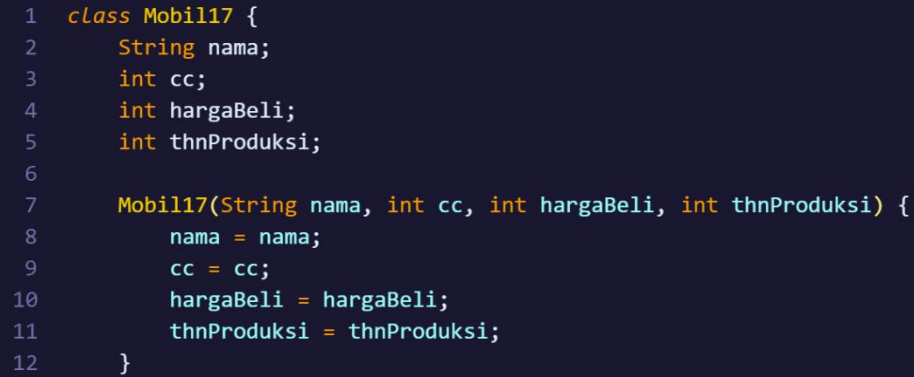
Mobil

```
nama:    String
cc:       int
hargaBeli: int
thnProduksi: int
```

```
hitungPajak(): double
hitungBiayaPerawatan():
double hitungBiayaTotal():
double printData(): void
```

- Biaya total merupakan hasil penambahan pajak dan biaya perawatan
 - Pajak ditentukan berdasar cc:
 - Cc > 1500 : pajak 10% dari harga beli
 - Selain itu dikenakan pajak 7.5% dari harga beli
 - Biaya perawatan ditentukan berdasar tahun produksi:
 - Tahun produksi < 2010 : biaya perawatan 8% dari harga beli
 - Tahun produksi > 2010 : biaya perawatan 5% dari harga beli
-
- Buat class **Mobil** sesuai dengan diagram class di atas!
 - Buat class/method **main** yang di dalamnya terdapat koleksi data Mobil sebanyak **n** data
 - Selanjutnya buat program untuk menghitung biaya perawatan rata-rata dari mobil dengan tahun produksi > 2010

- Selanjutnya buat program untuk mencari biaya total terbesar dari mobil dengan cc > 1500 dan tahun produksi < 2010 thn. Dan tampilkan data mobil dengan biaya total terbesar tersebut



```
1  class Mobil17 {  
2      String nama;  
3      int cc;  
4      int hargaBeli;  
5      int thnProduksi;  
6  
7      Mobil17(String nama, int cc, int hargaBeli, int thnProduksi) {  
8          nama = nama;  
9          cc = cc;  
10         hargaBeli = hargaBeli;  
11         thnProduksi = thnProduksi;  
12     }  
}
```

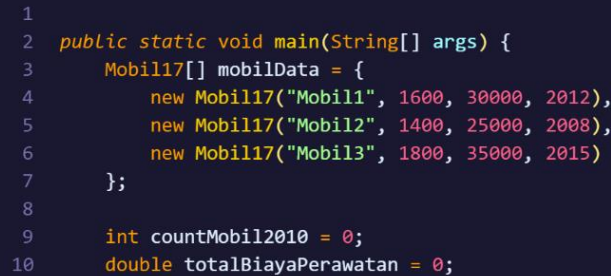
- **nama, cc, hargaBeli, dan thnProduksi** variabel ini adalah variabel anggota kelas **`Mobil17`**, digunakan untuk menyimpan data karakteristik mobil.
- **Mobil17 (String nama, int cc, int hargaBeli, int thnProduksi)** ini konstruktor kelas **`Mobil17`**. Konstruktor digunakan untuk menginisialisasi objek Mobil17 disaat objek dibuat.

```

1  double hitungPajak() {
2      if (cc > 1500) {
3          return 0.1 * hargaBeli;
4      } else {
5          return 0.075 * hargaBeli;
6      }
7  }
8
9  double hitungBiayaPerawatan() {
10     if (thnProduksi < 2010) {
11         return 0.08 * hargaBeli;
12     } else {
13         return 0.05 * hargaBeli;
14     }
15 }
16
17 double hitungBiayaTotal() {
18     return hitungPajak() + hitungBiayaPerawatan();
19 }
20
21 void printData() {
22     System.out.println("Nama: " + nama);
23     System.out.println("CC: " + cc);
24     System.out.println("Harga Beli: " + hargaBeli);
25     System.out.println("Tahun Produksi: " + thnProduksi);
26     System.out.println("Biaya Pajak: " + hitungPajak());
27     System.out.println("Biaya Perawatan: " + hitungBiayaPerawatan());
28     System.out.println("Biaya Total: " + hitungBiayaTotal());
29     System.out.println();
30 }

```

- Metode hitung pajak** ini menghitung nilai pajak berdasarkan aturan tertentu. Apabila kapasitas mesin (**cc**) mobil melebihi dari 1500, pajak dihitung sebagai **10%** dari harga beli. Jika tidak, maka pajak dihitung sebagai **7.5%** dari harga beli. **Metode hitungBiayaPerawatan** ini menghitung biaya perawatan berdasarkan tahun produksi mobil. Apabila tahun produksi mobil kurang dari 2010, biaya perawatan dihitung sebagai **8%** dari harga beli. Jika tidak, maka biaya perawatan akan dihitung sebagai **5%** dari harga beli. **Metode hitungBiayaTotal** ini menghitung total biaya mobil dengan menjumlahkan nilai pajak dan biaya perawatan tersebut. **Metode printData** ini digunakan untuk mencetak informasi lengkap tentang objek **`Mobil17`**. Setiap atribut seperti nama, kapasitas mesin (**`cc`**), harga beli, tahun produksi, biaya pajak, biaya perawatan, dan biaya total tersebut akan dicetak.



```
1
2  public static void main(String[] args) {
3      Mobil17[] mobilData = {
4          new Mobil17("Mobil1", 1600, 30000, 2012),
5          new Mobil17("Mobil2", 1400, 25000, 2008),
6          new Mobil17("Mobil3", 1800, 35000, 2015)
7      };
8
9      int countMobil2010 = 0;
10     double totalBiayaPerawatan = 0;
```

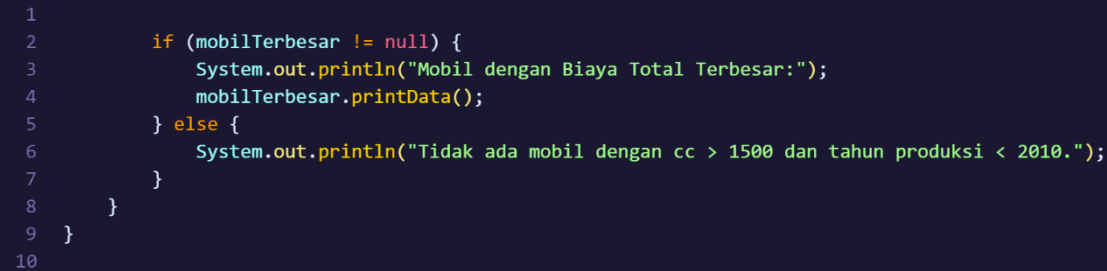
- Kode ``main`` ini adalah program utama yang membuat objek-objek dari kelas ``Mobil17``, menghitung beberapa statistik, dan mencetak hasilnya. Array dari objek ``Mobil17`` yaitu disebut ``mobilData``. Setiap elemen array adalah sebuah objek ``Mobil17`` yang diinisialisasi dengan parameter tertentu, seperti **nama mobil**, **kapasitas mesin (`cc`)**, **harga beli**, dan **tahun produksi**. Dalam loop ini, setiap objek ``Mobil17`` diambil dari array ``mobilData``. Jika tahun produksinya lebih dari 2010, maka biaya **perawatannya akan dihitung dan ditambahkan ke ``totalBiayaPerawatan``**. Setelah loop selesai memeriksa apakah ada mobil dengan tahun produksi lebih dari 2010. Apabila ada, maka rata-rata **biaya perawatan dihitung dan ditampilkan**. Jika tidak, pesan yang di cetak bahwa **tidak ada mobil dengan tahun produksi lebih dari 2010**.

```

1  int countMobil2010 = 0;
2  double totalBiayaPerawatan = 0;
3
4  for (Mobil17 mobil : mobilData) {
5      if (mobil.thnProduksi > 2010) {
6          totalBiayaPerawatan += mobil.hitungBiayaPerawatan();
7          countMobil2010++;
8      }
9  }
10
11 if (countMobil2010 > 0) {
12     double rataRataBiayaPerawatan = totalBiayaPerawatan / countMobil2010;
13     System.out.println("Rata-rata Biaya Perawatan Mobil > 2010: " + rataRataBiayaPerawatan);
14 } else {
15     System.out.println("Tidak ada mobil dengan tahun produksi > 2010.");
16 }
17
18 double maxBiayaTotal = 0;
19 Mobil17 mobilTerbesar = null;
20
21 for (Mobil17 mobil : mobilData) {
22     if (mobil.cc > 1500 && mobil.thnProduksi < 2010) {
23         double biayaTotal = mobil.hitungBiayaTotal();
24         if (biayaTotal > maxBiayaTotal) {
25             maxBiayaTotal = biayaTotal;
26             mobilTerbesar = mobil;
27         }
28     }
29 }
30

```

- Program menghitung **rata-rata biaya perawatan** dari mobil yang diproduksi setelah tahun 2010. Jumlah mobil yang memenuhi kriteria disimpan dalam **variabel `countMobil2010`**, dan total biaya perawatan disimpan dalam **totalBiayaPerawatan**. Setelah itu, program akan mencetak rata-rata biaya perawatan jika mobil ada yang memenuhi kriteria. Jika tidak, pesan **Tidak ada mobil dengan tahun produksi > 2010** dicetak. Setelah itu, program mencari mobil dengan biaya total terbesar dari mobil dengan kapasitas mesin lebih dari 1500 dan tahun produksi kurang dari 2010. Dalam proses pencarian, biaya total dihitung menggunakan metode **hitungBiayaTotal**. Jika biaya total mobil yang sedang diperiksa lebih besar dari **maxBiayaTotal**, nilai **maxBiayaTotal**, diperbarui, dan mobil tersebut dianggap sebagai mobil terbesar.



```
1
2     if (mobilTerbesar != null) {
3         System.out.println("Mobil dengan Biaya Total Terbesar:");
4         mobilTerbesar.printData();
5     } else {
6         System.out.println("Tidak ada mobil dengan cc > 1500 dan tahun produksi < 2010.");
7     }
8 }
9 }
10
```

- Setelah mencari mobil dengan biaya total tertinggi, akan diperiksa apakah variabel **mobilTerbesar** tidak bernilai null. Jika nilainya tidak null, program berhasil menemukan mobil yang memenuhi persyaratan. Selanjutnya, informasi tentang mobil terbesar akan dicetak menggunakan metode **printData** dari kelas **Mobil17**. Jika variabel **mobilTerbesar** bernilai null, ini berarti tidak ada mobil yang memenuhi persyaratan, dan pesan "Tidak ada mobil dengan cc > 1500 dan tahun produksi < 2010." dicetak.