

PACKAGES & INTERFACES

Package merupakan sebuah mekanisme yang dilakukan untuk mengorganisir kelas-kelas yang berkaitan, termasuk juga *interfaces*. Sebuah *package* tidak lebih dari sekedar *folder-folder* tempat kelas, sub-kelas dan *interfaces* berada. Di dalam pemrograman java, banyak sekali *class library* yang dapat digunakan, *library* inilah disebut dengan **API** - Application Programming Interface. Bila anda dapat mengakses API java, <https://docs.oracle.com/javase/8/docs/api/index.html>, disana terdapat berbagai *packages* yang ada, contohnya adalah “**java.io**”, “**java.lang**”, “**java.math**”, dan lain sebagainya. Berikut struktur *package* “java.io” termasuk kelas dan *interface* nya:

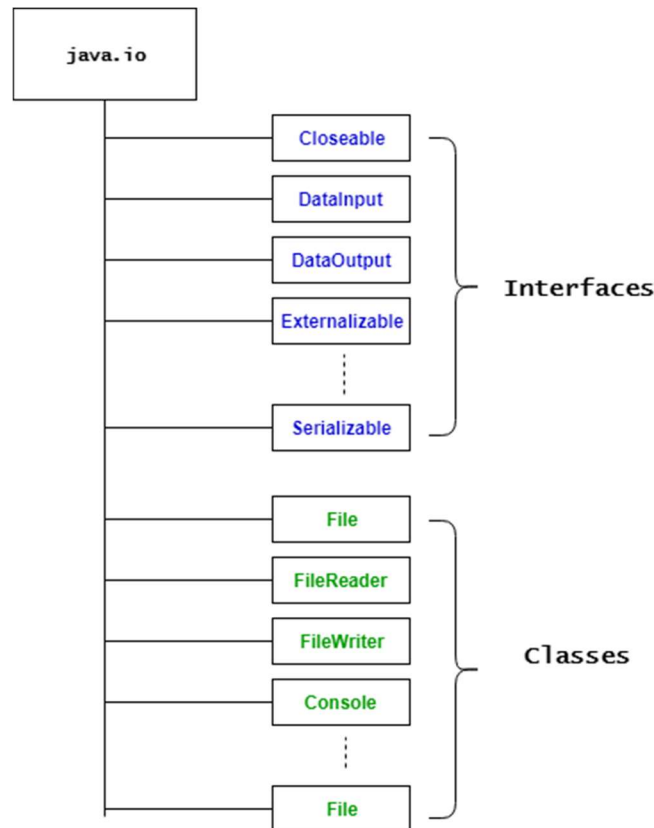


Figure 1. Salah satu contoh *package* di java, “java.io”, yang berisikan *interfaces* dan *classes* yang berkaitan.

Apabila anda menggunakan IDE terkini, seperti NetBeans dan juga Eclipse, pembuatan *package* sudah dimudahkan. Buatlah sebuah proyek baru dengan menggunakan NetBeans, misalnya, seperti langkah-langkah berikut:

File > New Project > (Categories) pilih Java > (Projects) pilih Java Application

Ketikkan **Project Name** dengan “PrjPackages”, lalu **Finish**. Rename *package* yang secara otomatis sudah dibuat oleh NetBeans menjadi “del.ac.id.main” (tekan tombol F2 tepat pada *package* yang di-highlight seperti contoh di bawah ini:

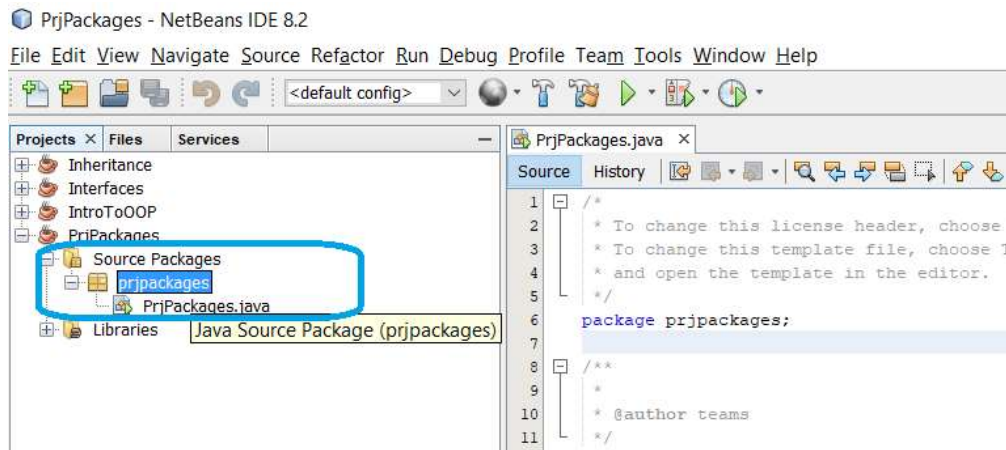


Figure 2. Rename nama package lewat IDE NetBeans

Jangan lupa untuk mencentang “Apply Rename on Comments” lalu tekan tombol “Refactor”, dan nama *package* pun berubah. Seperti yang telah disampaikan sebelumnya, bahwa sebuah *package* tidak lebih dari *folder*, tempat dimana kelas dan *interface* yang berkaitan disimpan. Anda dapat memeriksa *file* java yang telah anda buat sebelumnya (PrjPackages.java yang secara otomatis dibuat oleh NetBeans) dengan cara:

Sorot *file* PrjPackages.java > klik kanan > Properties > lalu sorot All Files

Di sana anda akan menemukan struktur *folder* sesuai dengan nama *package* yang telah dibuat sebelumnya, termasuk *files* yang ada.

Untuk membuat *package* yang baru dengan nama “del.ac.id.induks” dapat dilakukan melalui langkah-langkah berikut:

Pada panel **Projects** sebelah kanan, pilih **Source Packages** lalu klik kanan > Masukan “Package Name” dengan “del.ac.id. induks” > tekan tombol Finish.

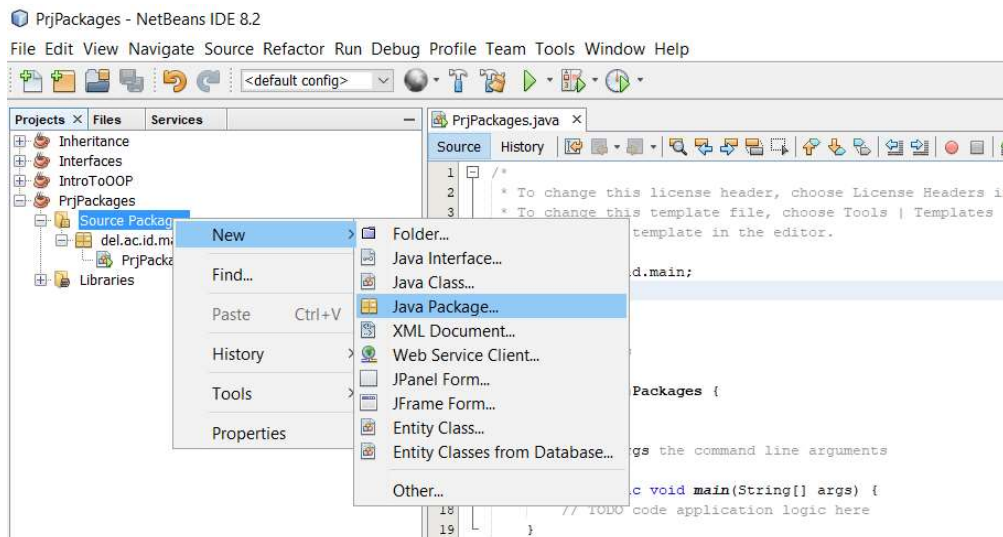


Figure 3. Membuat package baru pada project yang telah ada sebelumnya.

Di pemrograman java ada yang namanya *interface*. **Interface** berbeda dengan *class*. Apa sih maksud dan kegunaan *interface*? Contoh kasusnya seperti ini, anda di tingkat I telah mengerjakan proyek akhir (PA I). Kemudian, seorang dosen sebagai koordinator PA I mewajibkan anda semua untuk menggunakan sebuah *template* (kerangka kerja) dokumen yang telah disediakan. Akan tetapi, tiap-tiap *group pasti* mengganti konten dari dokumen tersebut **tanpa** mengubah *template* yang ada. *Template* inilah dapat dikatakan sebagai *interface*. Untuk lebih jelasnya penggunaan *interface*, ikutilah instruksi di bawah ini:

Buatlah *file interface* java pada *package* “del.ac.id.induks” dengan nama “IKendaraan.java” dengan cara:

Sorot “del.ac.id.induks” *package* lalu klik kanan > Pilih New > Pilih Java Interface > Masukkan nama *file* “IKendaraan” > Tekan tombol **Finish**.

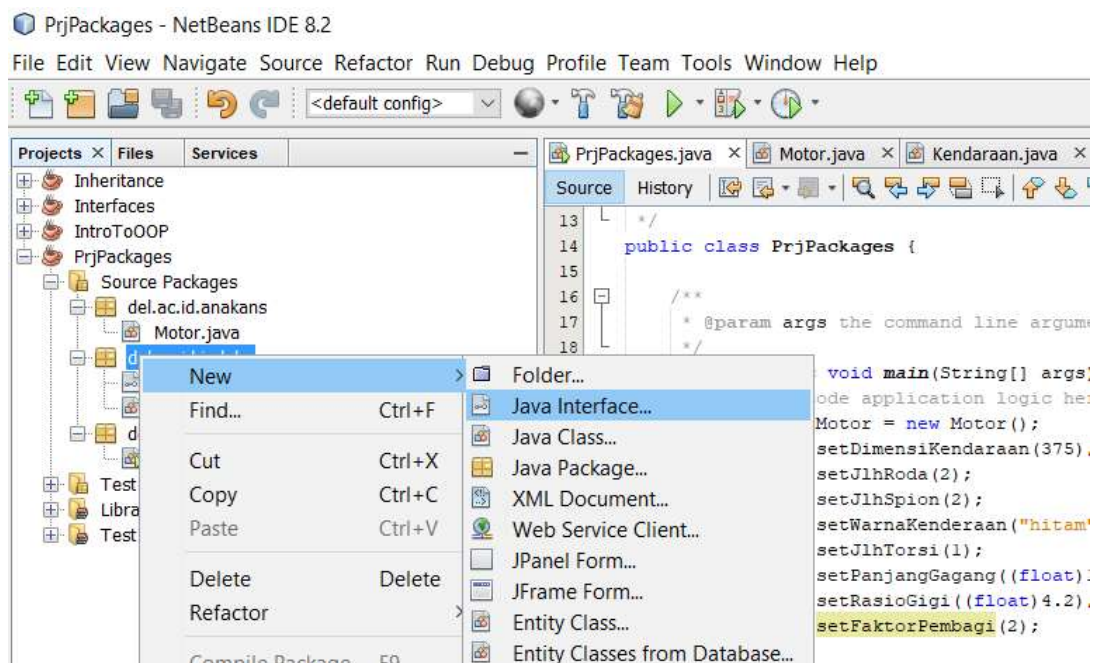


Figure 4. Cara membuat *interface* java melalui IDE NetBeans

Ketikan beberapa kode program pada kelas “Kendaraan.java” seperti contoh di bawah ini:

```
6 package del.ac.id.induks;
7
8 /**
9  *
10  * @author teams
11  */
12 public interface IKendaraan {
13     // prototype methods
14     void setJlhRoda(int value);
15     void setDimensiKendaraan(int value);
16     void setJlhSpion(int value);
17     void setWarnaKendaraan(String value);
18     void cetakInformasi(Object o);
19
20     int getJlhRoda();
21     int getDimensiKendaraan();
22     int getJlhSpion();
23
24     float kalkulasiPercepatan(int jlhTorsi, float panjangGagang, float rasioGigi);
25
26     String getWarnaKendaraan();
27 }
```

Kemudian buat sebuah kelas java tepat berada di dalam *package* “del.ac.id.induks” dan beri nama “Kendaraan.java”.

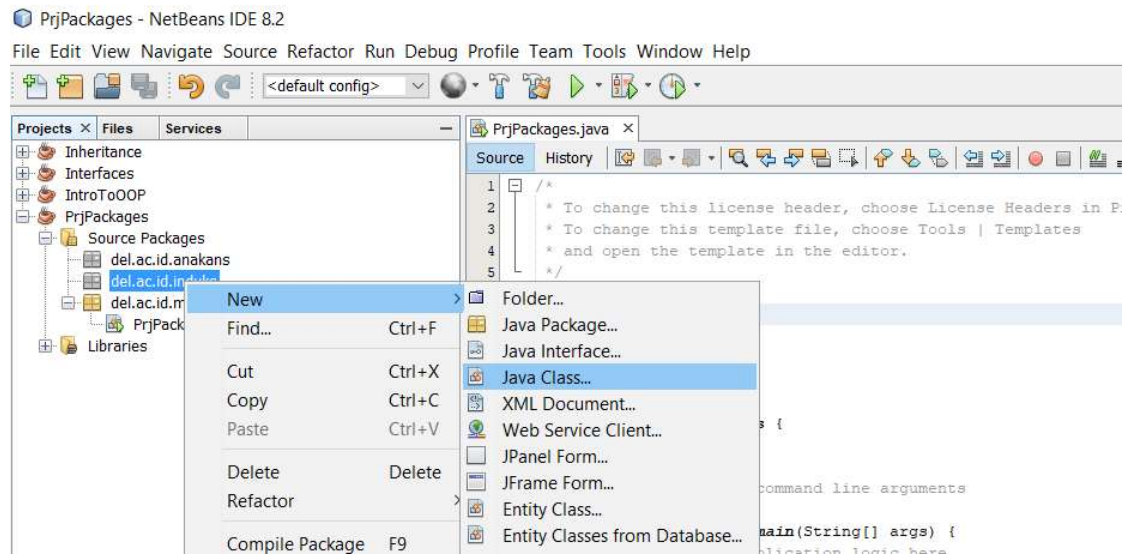


Figure 5. Cara pembuatan sebuah kelas di dalam sebuah *package* melalui IDE NetBeans

Ketikan beberapa kode program pada kelas “Kendaraan.java” seperti contoh di bawah ini:

```
6 package del.ac.id.induks;
7
8 /**
9  *
10  * @author teams
11  */
12 @
13 public class Kendaraan implements IKendaraan {
14     private int jlhRoda, dimensiKendaraan, jlhSpion;
15     private String warnaKendaraan;
16
17     public Kendaraan() {}
18
19     @Override
20     public void setJlhRoda(int value) {
21         jlhRoda = value;
22     }
23
24     @Override
25     public void setDimensiKendaraan(int value) {
26         dimensiKendaraan = value;
27     }
28
29     @Override
30     public void setJlhSpion(int value) {
31         jlhSpion = value;
32     }
33 }
```

```

33      @Override
34      public void setWarnaKendaraan(String value){
35          warnaKendaraan = value;
36      }
37
38      @Override
39      public void cetakInformasi(Object o){
40          if(o == null) System.out.println("Object is null");
41      }
42
43      @Override
44      public int getJlhRoda(){
45          return jlhRoda;
46      }
47
48      @Override
49      public int getDimensiKendaraan(){
50          return dimensiKendaraan;
51      }
52
53      @Override
54      public int getJlhSpion(){
55          return jlhSpion;
56      }
57
58      @Override
59      public String getWarnaKendaraan(){
60          return warnaKendaraan;
61      }
62
63      @Override
64      public float kalkulasiPercepatan(int jlhTorsi, float panjangGagang, float rasioGigi){
65          return jlhTorsi * panjangGagang * rasioGigi;
66      }
67  }

```

Setelah anda mengetikkan kode program Kendaraan.java, saatnya anda menurunkan segala sifat yang dimiliki kelas ini. Kelas anakan yang akan kita rancang, akan berada di dalam *package* "del.ac.id.anakans". Buatlah anakan dari kelas Kendaraan.java dan beri nama "Motor.java", lalu ketikkan kode program di bawah ini:

```

6      package del.ac.id.anakans;
7
8      import del.ac.id.induks.*;
9
10     /**
11      *
12      * @author teams
13      */
14     public class Motor extends Kendaraan {
15         private int jlhTorsi, faktorPembagi;
16         private float panjangGagang, rasioGigi;
17
18         public Motor(){}
19
20         public void setJlhTorsi(int value){
21             jlhTorsi = value;
22         }
23
24         public void setPanjangGagang(float value){
25             panjangGagang = value;
26         }
27
28         public void setRasioGigi(float value){
29             rasioGigi = value;
30         }
31
32         public void setFaktorPembagi(int value){
33             faktorPembagi = value;
34         }

```



```

36  @Override
37  public float kalkulasiPercepatan(int jlhTorsi, float panjangGagang, float rasioGigi) {
38      return super.kalkulasiPercepatan(jlhTorsi, panjangGagang, rasioGigi) * faktorPembagi;
39  }
40
41  @Override
42  public void cetakInformasi(Object o) {
43      System.out.println("=====");
44      System.out.println(String.format("Motor saya berwarna %s.",
45          ((Motor)o).getWarnaKendaraan()));
46      System.out.println(String.format("Dimensi motor tersebut kurang lebih %d meter persegi, ",
47          ((Motor)o).getDimensiKendaraan()));
48      System.out.println(String.format("dan dengan percepatan sekitar %.2f km/jam. ",
49          kalkulasiPercepatan(jlhTorsi, panjangGagang, rasioGigi)));
50      System.out.println(String.format("Motor tersebut memiliki %d kaca spion dan %d roda. ",
51          ((Motor)o).getJlhSpion(), ((Motor)o).getJlhRoda()));
52      System.out.println("=====");
53  }
54  }

```

Untuk melihat lebih jelas proses enkapsulasi dan *overriding method*, buatlah 2 kelas anakan tambahan, yaitu “Mobil.java” dan “Sepeda.java”.

```

6  package del.ac.id.anakans;
7
8  import del.ac.id.induks.Kendaraan;
9
10 /**
11  *
12  * @author teams
13  */
14 public class Mobil extends Kendaraan {
15     private int jlhTorsi, faktorPembagi;
16     private float panjangGagang, rasioGigi;
17
18     public Mobil() {}
19
20     public void setJlhTorsi(int value) {
21         jlhTorsi = value;
22     }
23
24     public void setPanjangGagang(float value) {
25         panjangGagang = value;
26     }
27
28     public void setRasioGigi(float value) {
29         rasioGigi = value;
30     }
31
32     public void setFaktorPembagi(int value) {
33         faktorPembagi = value;
34     }
35
36     @Override
37     public float kalkulasiPercepatan(int jlhTorsi, float panjangGagang, float rasioGigi) {
38         return (jlhTorsi * panjangGagang) / (faktorPembagi * rasioGigi);
39     }
40
41     @Override
42     public void cetakInformasi(Object o) {
43         System.out.println("=====");
44         System.out.println(String.format("Mobil saya berwarna %s.",
45             ((Mobil)o).getWarnaKendaraan()));
46         System.out.println(String.format("Dimensi mobil tersebut kurang lebih %d meter persegi, ",
47             ((Mobil)o).getDimensiKendaraan()));
48         System.out.println(String.format("dan dengan percepatan sekitar %.2f km/jam. ",
49             kalkulasiPercepatan(jlhTorsi, panjangGagang, rasioGigi)));
50         System.out.println(String.format("Mobil tersebut memiliki %d kaca spion dan %d roda. ",
51             ((Mobil)o).getJlhSpion(), ((Mobil)o).getJlhRoda()));
52         System.out.println("=====");
53     }
54 }

```

Figure 6. Kelas Mobil

```

6   package del.ac.id.anakans;
7
8   import del.ac.id.induks.Kendaraan;
9
10  /**
11   *
12   * @author teams
13   */
14  public class Sepeda extends Kendaraan {
15      public Sepeda() {}
16
17      @Override
18      public float kalkulasiPercepatan(int jlhTorsi, float panjangGagang, float rasioGigi) {
19          // return super.kalkulasiPercepatan(jlhTorsi, panjangGagang, rasioGigi);
20          // Karena sepeda tidak memiliki jumlah torsi, panjang gagang dan juga rasio gigi,
21          // disini akan diberi nilai default, yaitu 10 km/jam
22          return (jlhTorsi == 0 || panjangGagang == 0 || rasioGigi == 0) ?
23              (float)10.0 : (jlhTorsi * panjangGagang * rasioGigi);
24      }
25
26      @Override
27      public void cetakInformasi(Object o) {
28          System.out.println("=====");
29          System.out.println(String.format("Sepeda saya berwarna %s.",
30              ((Sepeda)o).getWarnaKendaraan()));
31          System.out.println(String.format("Dimensi sepeda tersebut kurang lebih %d meter persegi, ",
32              ((Sepeda)o).getDimensiKendaraan()));
33          System.out.println(String.format("dan dengan percepatan sekitar %.2f km/jam. ",
34              kalkulasiPercepatan(0, 0, 0)));
35          System.out.println(String.format("Sepeda tersebut memiliki %d kaca spion dan %d roda. ",
36              ((Sepeda)o).getJlhSpion(), ((Sepeda)o).getJlhRoda()));
37          System.out.println("=====");
38      }
39  }

```

Figure 7. Kelas Sepeda

Pada kelas main (PrjPackages.java) anda dapat mengetikkan beberapa potongan kode program berikut:

```

6   package del.ac.id.main;
7
8   import del.ac.id.anakans.*;
9
10  /**
11   *
12   * @author teams
13   */
14  public class PrjPackages {
15
16      /**
17       * @param args the command line arguments
18       */
19      public static void main(String[] args) {
20          // TODO code application logic here
21          Motor objMotor = new Motor();
22          objMotor.setDimensiKendaraan(274);
23          objMotor.setJlhRoda(2);
24          objMotor.setJlhSpion(2);
25          objMotor.setWarnaKendaraan("merah");
26          objMotor.setJlhTorsi(1);
27          objMotor.setPanjangGagang((float)15.5);
28          objMotor.setRasioGigi((float)4.2);
29          objMotor.setFaktorPembagi(2);

```

```
31 Mobil objMobil = new Mobil();
32 objMobil.setDimensiKendaraan(632);
33 objMobil.setJlhRoda(4);
34 objMobil.setJlhSpion(2);
35 objMobil.setWarnaKendaraan("hitam");
36 objMobil.setJlhTorsi(1);
37 objMobil.setPanjangGagang((float)20.5);
38 objMobil.setRasioGigi((float)50.4);
39 objMobil.setFaktorPembagi(3);
40
41 Sepeda objSepeda = new Sepeda();
42 objSepeda.setDimensiKendaraan(632);
43 objSepeda.setJlhRoda(2);
44 objSepeda.setJlhSpion(0);
45 objSepeda.setWarnaKendaraan("pink");
46
47 // invoking method cetakInformasi
48 objMotor.cetakInformasi(objMotor);
49 objMobil.cetakInformasi(objMobil);
50 objSepeda.cetakInformasi(objSepeda);
51
52 }
```

TUGAS

1. Lakukan analisa mengapa data member `jlhTorsi`, `faktorPembagi`, `panjangGagang`, dan `rasioGigi` termasuk setter-getter nya pada kelas `Motor` dan `Mobil` **tidak** dibuat di dalam `IKendaraan`? Buatlah laporan anda dalam bentuk *softcopy*.
2. Ubahlah seluruh *access modifier* metode yang ada pada kelas “Kendaraan” menjadi “protected”. Lakukan observasi kecil, mengapa perubahan ini mengakibatkan *error* pada program. Buatlah laporan anda dalam bentuk *softcopy*.
3. Sama seperti soal no.2, lakukan perubahan *access modifier* metode pada kelas “Mobil” menjadi “protected”. Lakukan observasi kecil, mengapa perubahan ini mengakibatkan *error* pada program. Buatlah laporan anda dalam bentuk *softcopy*.
4. Menyambung pertanyaan no.3 di atas, mengapa metode yang *access modifier*-nya “protected” yang telah diubah sebelumnya pada kelas “Mobil”, dapat di-*invoke* oleh kelas “Sepeda” melalui instansiasi kelas tersebut (Mobil)? Buatlah laporan anda dalam bentuk *softcopy*.
5. Ubahlah kode program di atas dimana *setting* value untuk data member `jlhRoda`, `dimensiKendaraan`, `jlhSpion`, dan `warnaKendaraan` tanpa menggunakan *setter*, namun dengan memanfaatkan *constructor*. (Note: *getter* tidak perlu diubah **apabila** masih dimungkinkan).