

Pertemuan 9

Package Diagram

Package Diagram

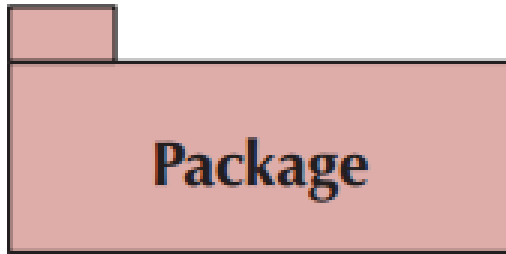
- Package

Package bisa dianggap sebagai folder. Ketika package digunakan dalam bahasa pemrograman seperti Java, package diimplementasikan sebagai folder.

- Package Diagram

Package diagram bisa diartikan sebagai sebuah class diagram yang hanya menunjukkan package.

Elemen-Elemen Package Diagram



- Package
 - Digunakan untuk menyederhanakan diagram UML dengan mengelompokkan elemen terkait menjadi satu elemen tingkat yang lebih tinggi.

Elemen-Elemen Package Diagram



- Dependency Relationship
 - Mewakili ketergantungan antara package: Jika package diubah, package dependen juga bisa dimodifikasi.
 - Memiliki panah yang diambil dari package tergantung pada package yang bergantung padanya

Dependency dan Stereotype

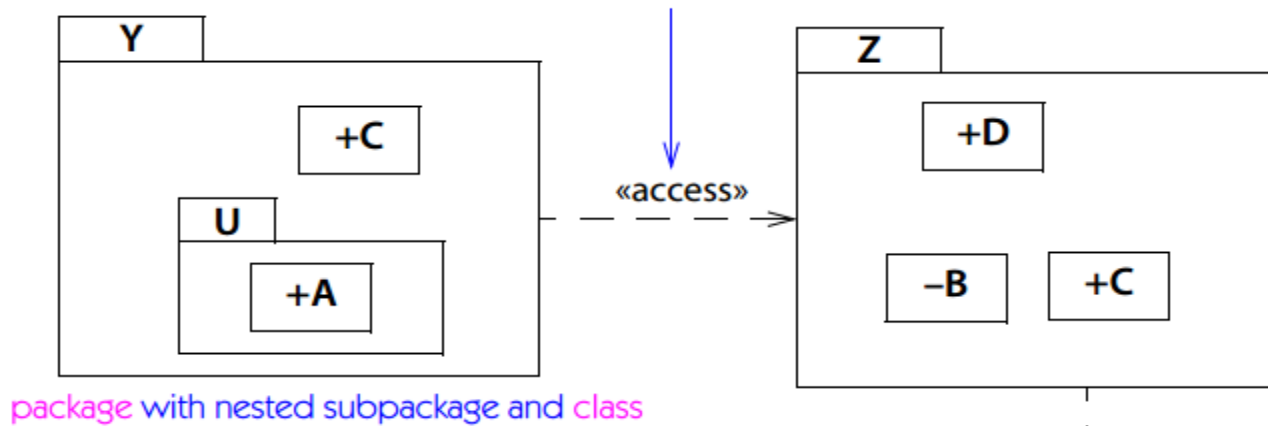
- Ada dua stereotype yang biasanya digunakan dalam menggambarkan package diagram, yaitu **<<import>>** dan **<<access>>**.

Table 14-3: UML Relationships

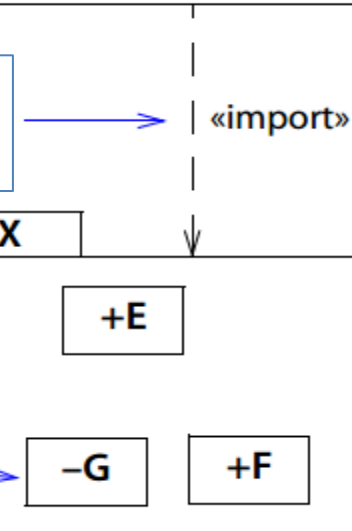
| Kind | Variety | Notation | ----> Keyword or Symbol |
|----------------------|------------------|----------------|--|
| abstraction | derivation | dependency | «derive» |
| | manifestation | dependency | «manifest» |
| | realization | realization | ----> |
| | refinement | dependency | «refine» |
| | trace dependency | dependency | «trace» |
| association | | association | ———— |
| binding | | dependency | «bind» (parameter _{list} ,) |
| deployment | | dependency | «deploy» or physical nesting |
| extend | | dependency | «extend» (extension point _{list} ,) |
| extension | | extension | ————> |
| generalization | | generalization | ————> |
| import | private | dependency | «access» |
| | public | dependency | «import» |
| include | | dependency | «include» |
| information flow | | dependency | «flow» |
| package merge | | dependency | «merge» |
| permission | | dependency | «permit» |
| protocol conformance | | | none specified |
| substitution | | dependency | «substitute» |
| usage | call | dependency | «call» |
| | creation | dependency | «create» |
| | instantiation | dependency | «instantiate» |
| | responsibility | dependency | «responsibility» |
| | send | dependency | «send» |

Stereotype Import dan Access

Package Y menambahkan konten public (yang bertanda +) dari package Z ke Y secara private.



Package Z menambahkan konten public (yang bertanda +) dari package X ke Z.



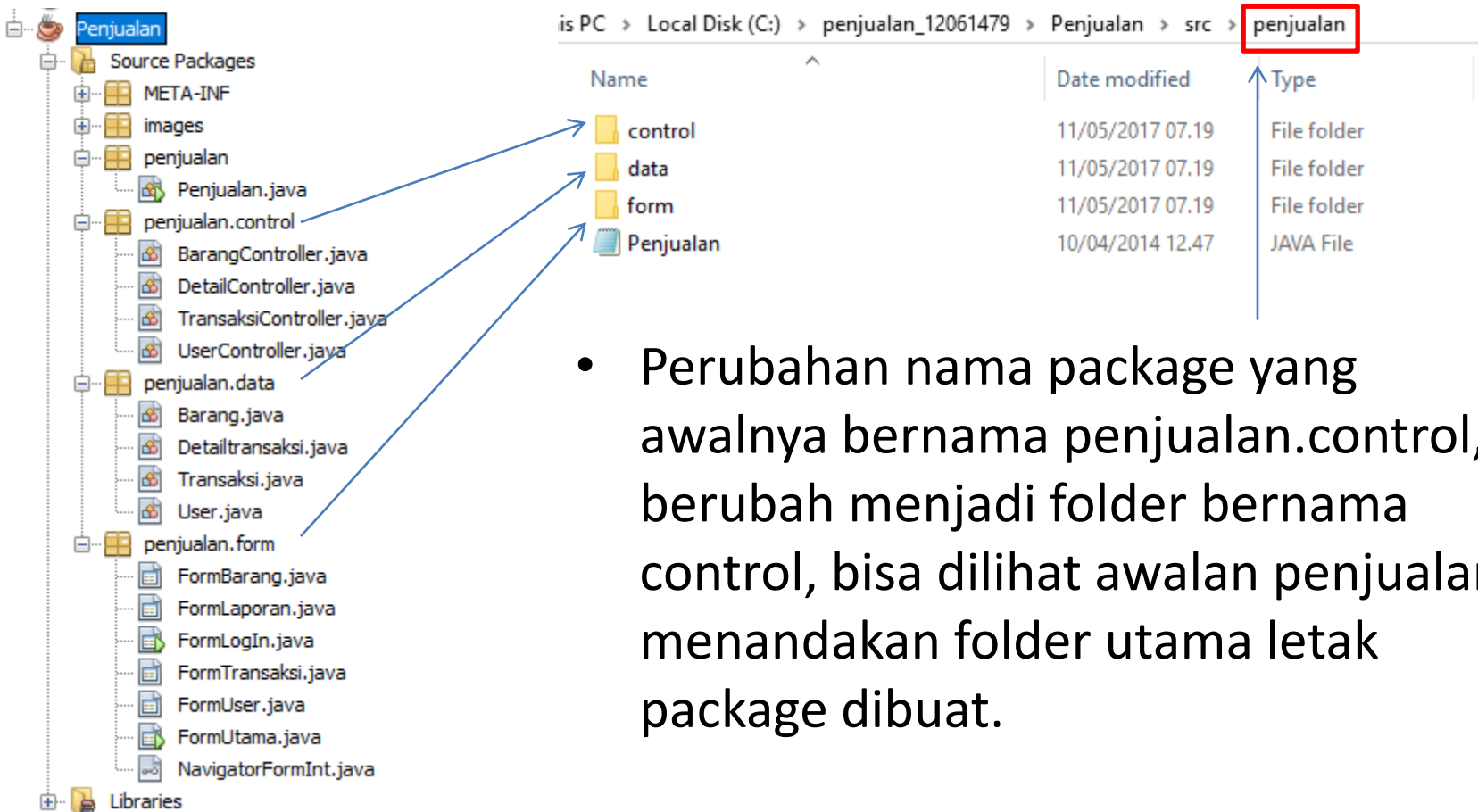
Pedoman Membuat Package Diagram

1. Gunakan package diagram untuk secara logis mengatur desain.
2. Jika ada hubungan inheritance, gambarlah secara vertikal, dengan package yang berisi superclass ditempatkan di atas package yang berisi subclass. Gambarlah secara horizontal untuk mendukung aggregation dan hubungan association, dengan package yang ditempatkan berdampingan.
3. Ketika ada hubungan ketergantungan (dependency) pada diagram, digambarkan dengan arah dari subclass ke superclass, dari klien ke server.

Pedoman Membuat Package Diagram

4. Saat menggunakan package untuk mengelompokkan use case, pastikan untuk menyertakan aktor dan asosiasi yang ada pada use case tersebut.
5. Berikan setiap package nama yang sederhana, tetapi deskriptif.
6. Buatlah package yang bersatu padu (kohesif).

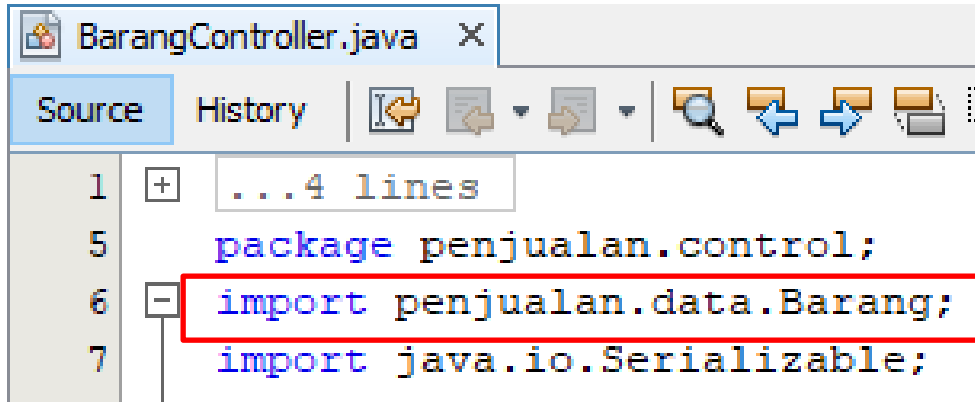
Implementasi Package dalam Java



| Name | Date modified | Type |
|-----------|------------------|-------------|
| control | 11/05/2017 07.19 | File folder |
| data | 11/05/2017 07.19 | File folder |
| form | 11/05/2017 07.19 | File folder |
| Penjualan | 10/04/2014 12.47 | JAVA File |

- Perubahan nama package yang awalnya bernama penjualan.control, berubah menjadi folder bernama control, bisa dilihat awalan penjualan. menandakan folder utama letak package dibuat.

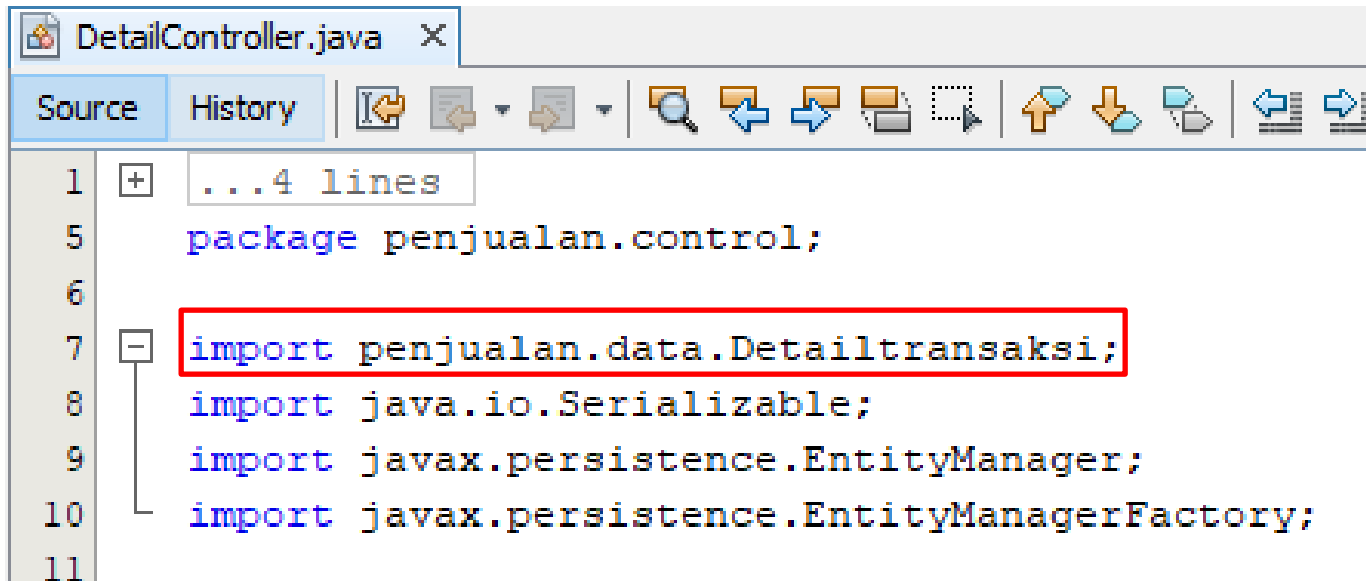
Import File Antar Package



```
BarangController.java X
Source History
1 ...4 lines
5 package penjualan.control;
6 import penjualan.data.Barang;
7 import java.io.Serializable;
```

- File BarangController mengimport file Barang.java yang ada di package penjualan.data

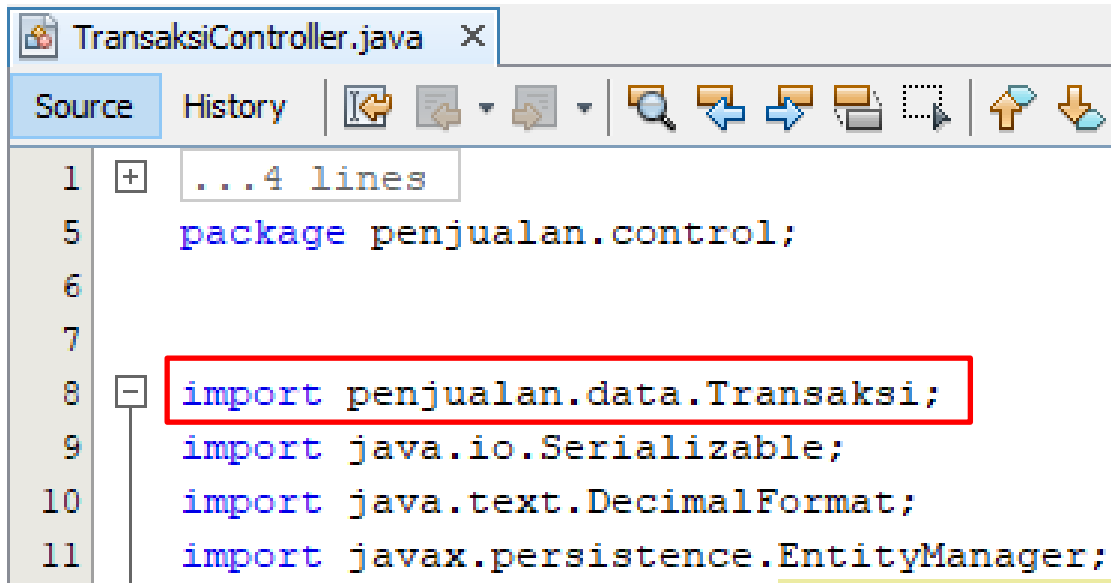
Import File Antar Package



```
DetailController.java X
Source History
1 ...4 lines
5 package penjualan.control;
6
7 import penjualan.data.Detailtransaksi;
8 import java.io.Serializable;
9 import javax.persistence.EntityManager;
10 import javax.persistence.EntityManagerFactory;
11
```

- File DetailController mengimport file DetailTransaksi.java yang ada di package penjualan.data

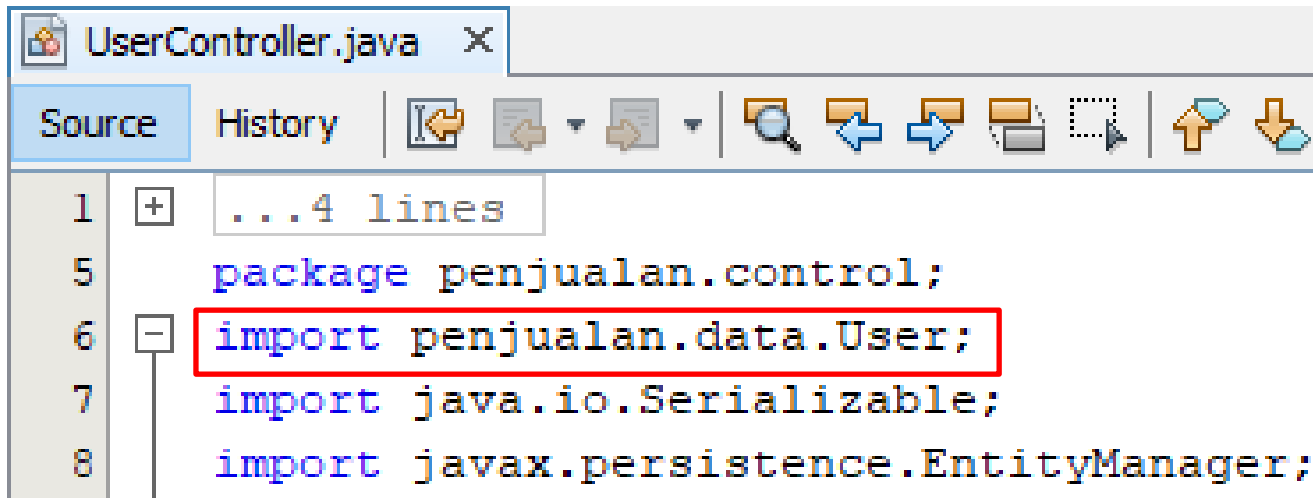
Import File Antar Package



```
TransaksiController.java X
Source History
1  ...4 lines
5  package penjualan.control;
6
7
8  import penjualan.data.Transaksi;
9  import java.io.Serializable;
10 import java.text.DecimalFormat;
11 import javax.persistence.EntityManager;
```

- File TransaksiController mengimport file Transaksi.java yang ada di package penjualan.data

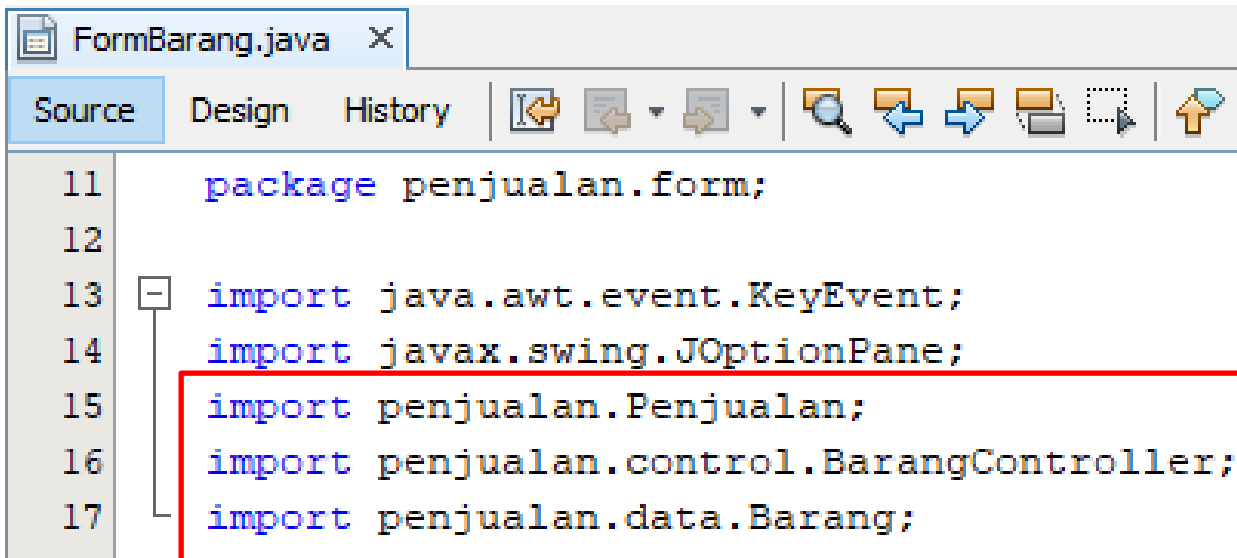
Import File Antar Package



```
UserController.java X
Source History
1 ...4 lines
5 package penjualan.control;
6 import penjualan.data.User;
7 import java.io.Serializable;
8 import javax.persistence.EntityManager;
```

- File UserController mengimport file User.java yang ada di package penjualan.data

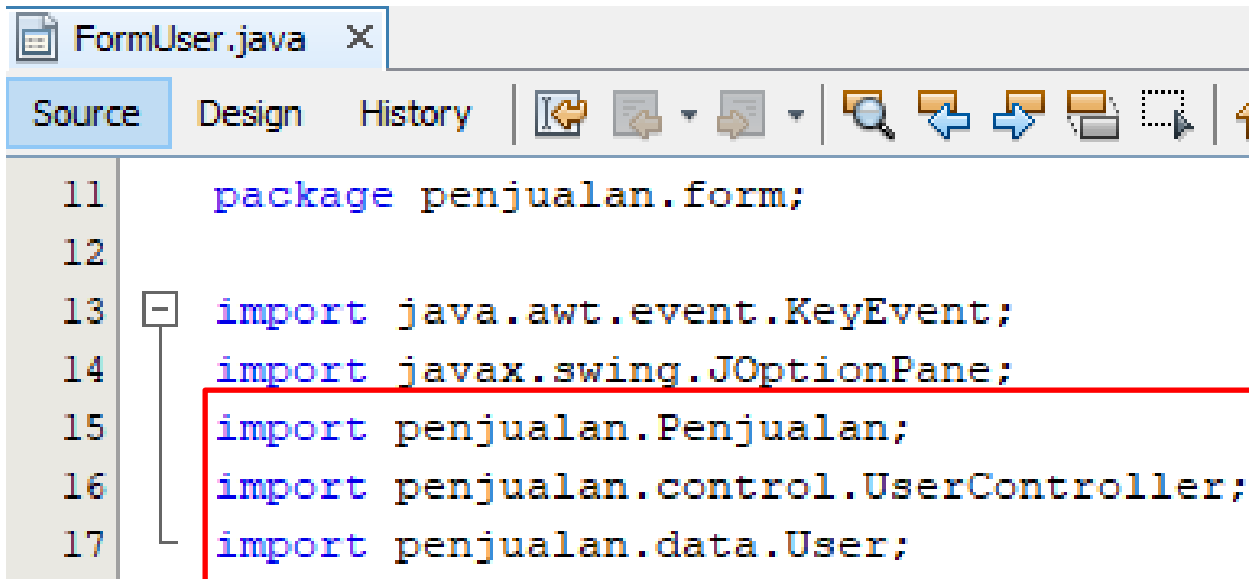
Import File Antar Package



```
FormBarang.java X
Source Design History
11 package penjualan.form;
12
13 import java.awt.event.KeyEvent;
14 import javax.swing.JOptionPane;
15 import penjualan.Penjualan;
16 import penjualan.control.BarangController;
17 import penjualan.data.Barang;
```

- File FormBarang mengimport file penjualan.java yang ada di package penjualan, file BarangController yang ada di package penjualan.control, dan file Barang yang ada di package penjualan.data

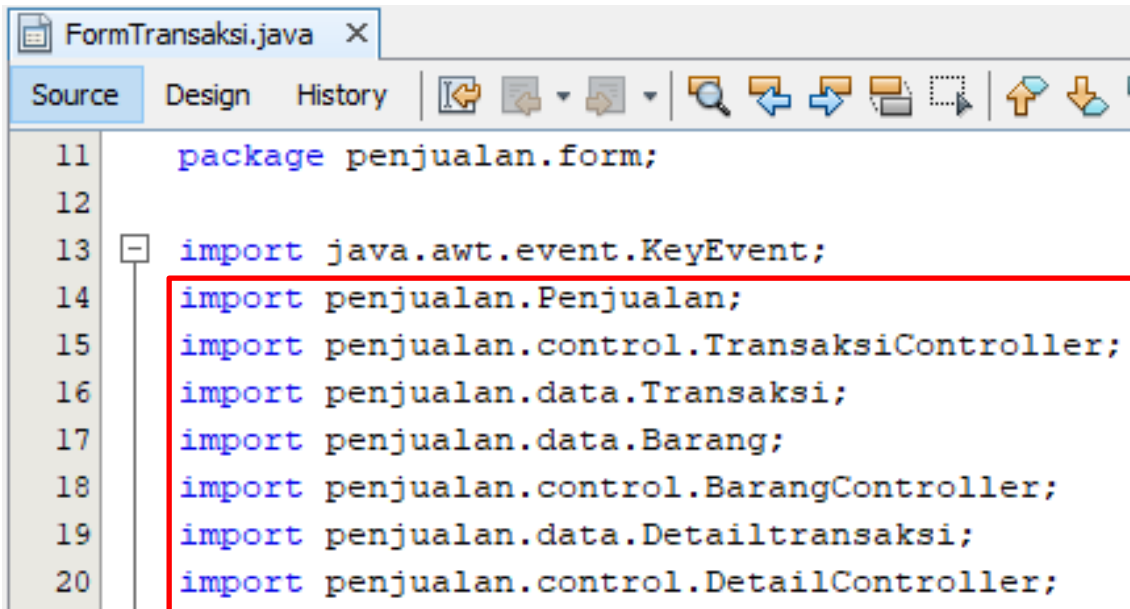
Import File Antar Package



```
FormUser.java X
Source Design History
11 package penjualan.form;
12
13 import java.awt.event.KeyEvent;
14 import javax.swing.JOptionPane;
15 import penjualan.Penjualan;
16 import penjualan.control.UserController;
17 import penjualan.data.User;
```

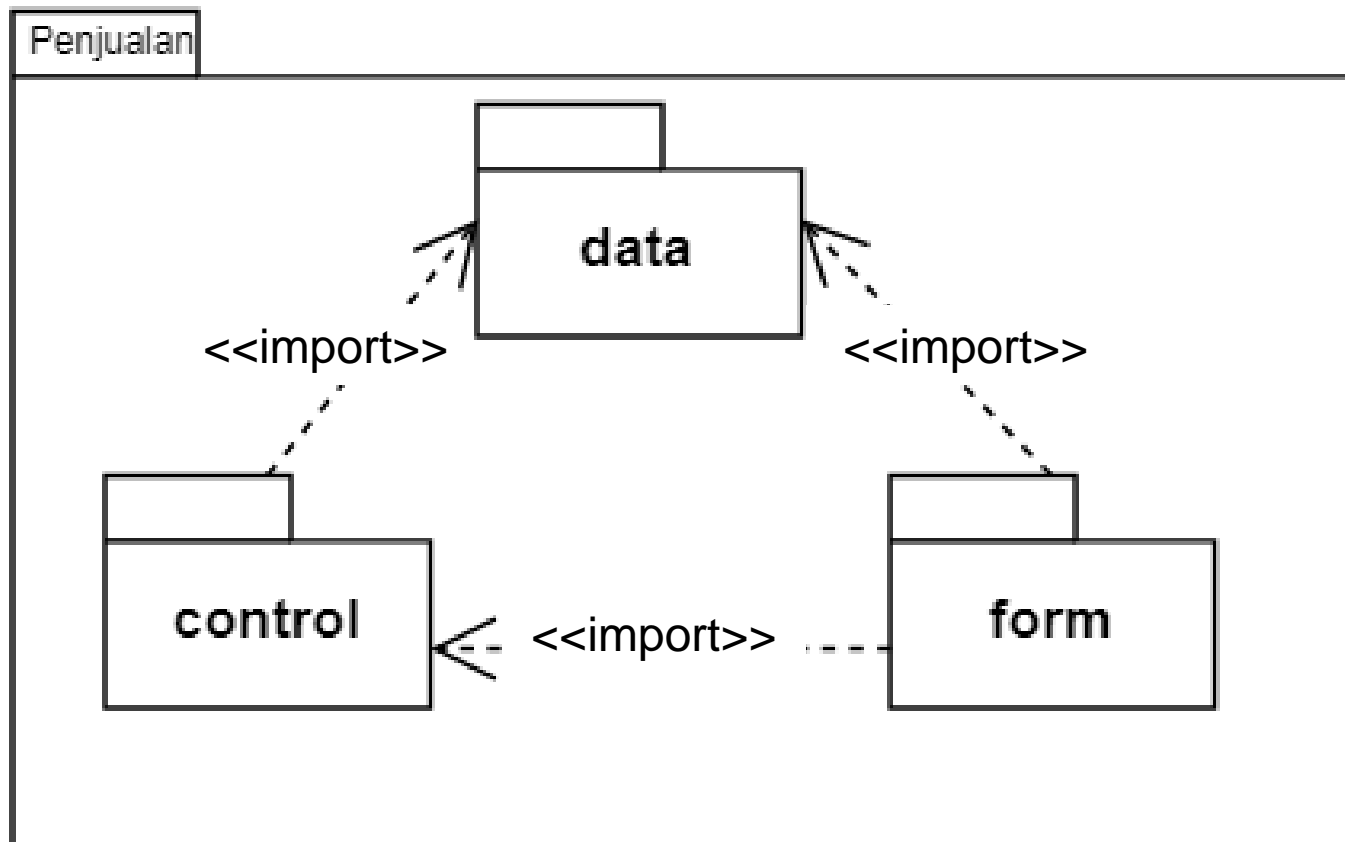
- File FormUser mengimport file penjualan.java yang ada di package penjualan, file UserController yang ada di package penjualan.control, dan file User yang ada di package penjualan.data

Import File Antar Package



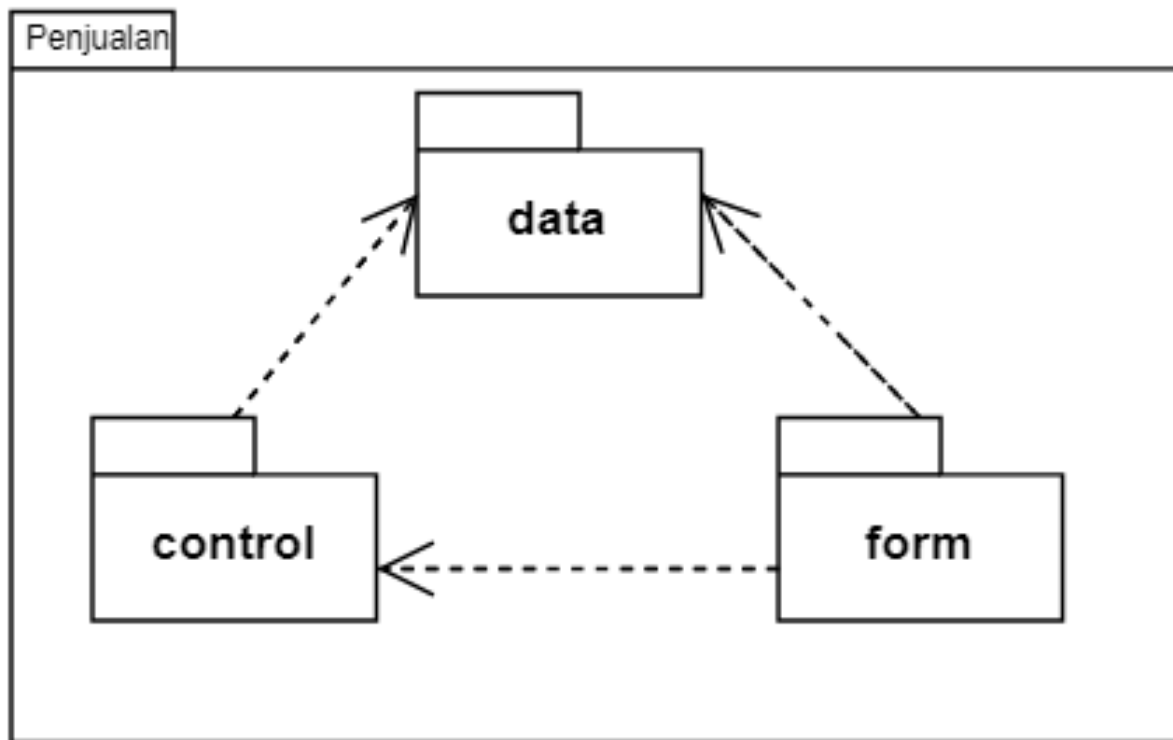
```
FormTransaksi.java x
Source Design History
11 package penjualan.form;
12
13 import java.awt.event.KeyEvent;
14 import penjualan.Penjualan;
15 import penjualan.control.TransaksiController;
16 import penjualan.data.Transaksi;
17 import penjualan.data.Barang;
18 import penjualan.control.BarangController;
19 import penjualan.data.Detailtransaksi;
20 import penjualan.control.DetailController;
```

Package Diagram



Package Diagram

- Beberapa referensi tidak menyantumkan stereotype/keyword Import di dalam diagram.



Tugas

- Buatlah kelompok terdiri dari maksimal 5 orang.
- Tugas dikerjakan di kelas.
- Waktu 1 jam.
- Buatlah diagram package (tanpa menyantumkan stereotype) dari soal cerita di slide berikut.

Tugas

- Diagram terdiri dari 4 package, yaitu: Customer Package, Order Package, Book Package, dan Shopping Package.
- Customer package menambahkan konten public dari Order package ke Customer package.
- Customer package menambahkan konten public dari Shopping package ke Customer package.
- Order package menambahkan konten public dari Book package ke Order package.
- Shopping package menambahkan konten public dari Book package ke Shopping package.