PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN 10 : COLLECTION

Penyusun:

Rizky Satria Gunawan

241511089

2C-D3

PROGRAM STUDI D3 TEKNIK INFORMATIKA

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

POLITEKNIK NEGERI BANDUNG

2025

Github Link : Mata-Kuliah-PBO/FolderTugas10 at main · RizkySatria123/Mata-Kuliah-PBO

1. Pendahuluan

Praktikum ini bertujuan untuk menerapkan konsep-konsep inti Pemrograman Berorientasi Objek (PBO) dalam bahasa Java, dengan fokus pada penggunaan *Generics* dan *Collections Framework*. Laporan ini menganalisis dua tugas utama yang telah diimplementasikan dalam file StudentJTK.java dan GenericStackDemo.java.

Tujuan Praktikum:

1. Mendesain dan mengimplementasikan kelas Student dengan menerapkan prinsip **enkapsulasi** dan **generics** untuk atribut departemen.
2. Menggunakan *collection* ArrayList untuk menyimpan objek Student dan melakukan pengurutan (sorting).
3. Menggunakan *collection* Vector untuk menyimpan objek Student dan melakukan filtrasi data.
4. Menganalisis dan mengkonversi implementasi Stack berbasis array (dari JP_2_4_Practice.pdf) menjadi implementasi Stack generik berbasis ArrayList (sesuai GenericStackDemo.java).

2. Hasil dan Pembahasan

Analisis dibagi berdasarkan dua file implementasi utama.

2.1. Task 1: Kelas Student dan Collections (StudentJTK.java)

Tugas ini berfokus pada pembuatan kelas Student dan manipulasi objeknya menggunakan ArrayList dan Vector.

Temuan dan Analisis Kode:

- **Enkapsulasi dan Generics:** Kelas Student<T> berhasil dibuat sebagai kelas generik. Ini memungkinkan atribut department memiliki tipe data yang fleksibel (misalnya String atau Integer), sesuai dengan permintaan tugas. Semua atribut (studentID, name, address, department) dideklarasikan sebagai private dan hanya dapat diakses melalui *method* public (getter dan setter), yang secara efektif menerapkan prinsip **enkapsulasi**.
- **Sorting (ArrayList):** Kelas Student mengimplementasikan *interface* Comparable<Student<T>>. *Method* compareTo di-override untuk membandingkan objek Student berdasarkan atribut name.

Hal ini memungkinkan Collections.sort(studentsList) untuk mengurutkan ArrayList secara alfabetis berdasarkan nama mahasiswa.

- **Filtering (Vector):** Sebuah Vector digunakan untuk menyimpan 5 objek Student baru. Kode kemudian melakukan iterasi pada Vector dan menggunakan percabangan if untuk memfilter dan menampilkan hanya mahasiswa yang memiliki departemen "Computer Science".

Hasil Praktikum (Output StudentJTK.java):

Daftar mahasiswa (ArrayList) setelah diurutkan berdasarkan nama:

Student{id='S002', name='Andi Putra', address='Jakarta', department=Sistem Informasi}

Student{id='S005', name='Ikhsan Satriadi', address='Semarang', department=Teknik Informasi}

Student{id='S004', name='Muhamad Syahid', address='Padang', department=Manajemen Informatika}

Student{id='S001', name='Rizky Satria', address='Bandung', department=Teknik Informatika}

Student{id='S003', name='Yazid Alrasyid', address='Bangka Belitung', department=Teknik Komputer}

Mahasiswa di departemen Computer Science (Vector):

Student{id='V001', name='Arman', address='Bandung', department=Computer Science}

Student{id='V003', name='Gita', address='Depok', department=Computer Science}

```
PS C:\Users\lenovo\OneDrive - Politeknik Negeri Bandung\Documents\rizky satria\Kampus Polban\Mata-Kuliah-PBO\Folde
rTugas10>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users
\lenovo\AppData\Roaming\Code\User\workspaceStorage\3fb6caa2b1203d53463705e7e9ec61ee\redhat.java\jdt_ws\FolderTugas
10_66d02949\bin' 'StudentJTK'
Daftar mahasiswa (ArrayList) setelah diurutkan berdasarkan nama:
Student{id='S002', name='Andi Putra', address='Jakarta', department=Sistem Informasi}
Student{id='S005', name='Ikhsan Satriadi', address='Semarang', department=Teknik Informasi}
Student{id='S004', name='Muhamad Syahid', address='Padang', department=Manajemen Informatika}
Student{id='S001', name='Rizky Satria', address='Bandung', department=Teknik Informatika}
Student{id='S003', name='Yazid Alrasyid', address='Bangka Belitung', department=Teknik Komputer}

Mahasiswa di departemen Computer Science (Vector):
Student{id='V001', name='Arman', address='Bandung', department=Computer Science}
Student{id='V003', name='Gita', address='Depok', department=Computer Science}
PS C:\Users\lenovo\OneDrive - Politeknik Negeri Bandung\Documents\rizky satria\Kampus Polban\Mata-Kuliah-PBO\Folde
rTugas10>
```

Figure 1: Output Task 1

2.2. Task 2: Konversi ke GenericStack (GenericStackDemo.java)

Tugas ini, yang diinstruksikan dalam file JP_2_4_Practice.pdf , adalah mengkonversi kelas ArrayStack (yang menggunakan int[]) menjadi kelas generik menggunakan ArrayList.

Temuan dan Analisis Kode:

- **Konversi dari Array ke ArrayList:** File GenericStackDemo.java berhasil mengimplementasikan kelas GenericStack<E>. Perubahan mendasar dari ArrayStack (dalam PDF) adalah:
    - Penyimpanan internal diubah dari private int[] items menjadi private final ArrayList<E> items.
    - Konstruktor tidak lagi memerlukan maxsize, karena ArrayList memiliki ukuran yang dinamis.
- **Logika Stack (LIFO):**
    - *Method* push(E item) kini menggunakan items.add(item) untuk menambahkan elemen ke akhir *list*. Pengecekan *overflow* tidak lagi diperlukan.
    - *Method* pop() menggunakan items.remove(items.size() - 1) untuk menghapus dan mengembalikan elemen terakhir dari ArrayList, yang secara tepat mensimulasikan perilaku LIFO (Last-In, First-Out).
    - Pengecekan *underflow* tetap dipertahankan dengan memeriksa isEmpty() sebelum menghapus elemen.
- **Keuntungan Generics:** *Method* main dalam GenericStackDemo.java membuktikan fleksibilitas implementasi baru dengan membuat GenericStack<Integer> dan GenericStack<String>. Keduanya berfungsi dengan baik tanpa perlu *casting* dan memberikan keamanan tipe (*type safety*).

Vocabulary:

HashSet	Identify the vocabulary word for each definition below. A set similar to an ArrayList without any specific ordering.
List	An ordered Collection that may contain duplicates.
Collection	An interface used to define a group of objects. This includes lists and sets.

ArrayList	A list that is very similar to an array.
Set	A Collection of elements that does not contain any duplicates.

Hasil Praktikum (Output GenericStackDemo.java):

Pop dari GenericStack<Integer>:

30

20

10

Pop dari GenericStack<String>:

PBO

Generics

Stack

PHP

Java

```
PS C:\Users\lenovo\OneDrive - Politeknik Negeri Bandung
rTugas10>  & 'C:\Program Files\Java\jdk-21\bin\java.exe
\lenovo\AppData\Roaming\Code\User\workspaceStorage\3fb6
10_66d02949\bin' 'GenericStackDemo'
Pop dari GenericStack<Integer>:
30
20
10

Pop dari GenericStack<String>:
PBO
Generics
Stack
PHP
Java
PS C:\Users\lenovo\OneDrive - Politeknik Negeri Bandung
rTugas10>
```

3. Kesimpulan

Praktikum ini telah berhasil diselesaikan. Temuan utamanya adalah:

1. **Enkapsulasi** berhasil diterapkan pada kelas Student untuk melindungi data internal.
2. **Generics** (tipe <T> dan <E>) terbukti sangat efektif dalam menciptakan kode yang fleksibel dan tipe-aman, baik pada kelas Student (untuk departemen) maupun pada GenericStack (untuk tipe elemen).
3. **ArrayList** ideal untuk penyimpanan data dinamis dan mudah diintegrasikan dengan *method* Collections.sort (dengan implementasi Comparable) untuk pengurutan.
4. **Vector** berfungsi baik untuk penyimpanan data yang memerlukan filtrasi manual, meskipun dalam aplikasi modern ArrayList seringkali lebih disukai kecuali jika *thread-safety* diperlukan.
5. Konversi ArrayStack menjadi GenericStack berbasis ArrayList berhasil dilakukan, menghasilkan struktur data yang lebih modern, dinamis (tanpa maxsize), dan dapat digunakan kembali untuk tipe data apa pun.