

# Tugas Praktikum 7

## 1. Buat interface dan class berikut

- Abstract class **LibraryItem**, dengan ketentuan
  - Atribut:
    - title
    - itemId : int
    - isBorrowed
  - Menggunakan constructor.
  - Abstract method:
    - getDescription() : string → Mengembalikan deskripsi item
    - borrowItem(int days) : String → Mengembalikan deskripsi peminjaman dengan format berikut, "Item [item.title] berhasil dipinjam selama [days] hari"
    - calculateFine(int daysLate): double → Mengembalikan jumlah denda berdasarkan besaran denda item dan jumlah keterlambatan
  - Method:
    - returnItem(): string → mengubah status peminjaman menjadi *false* dan mengembalikan string dengan format "[title] dikembalikan"
- Class **Book** dengan mengimplementasikan **LibraryItem**, dengan ketentuan:
  - Attribute tambahan:
    - author
  - Aturan peminjaman: Dapat dipinjam maksimal 14 hari. Kembalikan exception **IllegalArgumentException** jika melebihi 14 hari atau item telah dipinjam dan belum dikembalikan.
  - Denda: 10000 per hari keterlambatan
  - Deskripsi: "Buku: [title] oleh [author], ID: [itemId]"
- Class **DVD** dengan mengimplementasikan **LibraryItem**, dengan ketentuan:
  - Attribute tambahan:
    - duration (dalam menit)
  - Aturan peminjaman: Dapat dipinjam maksimal 7 hari. Kembalikan exception **IllegalArgumentException** jika melebihi 7 hari.
  - Denda: 25000 per hari keterlambatan
  - Deskripsi: "DVD: [title], durasi [duration] menit, ID: [itemId]"
- Class **Member** dengan ketentuan:
  - Attribute:
    - name
    - memberId
    - borrowedItems (List)
  - Method:
    - borrow(LibraryItem item, int days): String → Memanggil item.borrowItem(days) , mengatur status peminjaman item, dan

menambahkan ke daftar barang yang dipinjam. Apabila item tidak tersedia (sudah dipinjam), kembalikan exception `IllegalStateException`. Apabila berhasil, kembalikan string "Item [item.title] berhasil dipinjam selama [days] hari"

- `returnItem(LibraryItem item, int daysLate)` : String → Memanggil `item.returnItem()`, menghapus dari daftar peminjaman member, menghitung denda. Mengembalikan string "Item [item.title] berhasil dikembalikan dengan denda: Rp xxx.xxx"
- `getBorrowedItems()`: Print daftar item yang sedang dipinjam atau "Tidak ada item yang dipinjam". Format daftar item

```
+-----+-----+
| ID   | Judul   |
+-----+-----+
| 123  | Buku 1  |
| 124  | Buku 2  |
+-----+-----+
```

Silahkan tambahkan method lain apabila diperlukan.

2. Untuk manajemen perpustakaan yang lebih baik, buat 2 class berikut

- **LibraryLogger**, dengan ketentuan
  - Variabel: logs (List).
  - Metode:
    - `logActivity(String activity)`: String → Menambahkan aktivitas ke logs dengan timestamp (contoh: "2025-05-09 10:00:00 [Buku/DVD] dipinjam oleh [member name]").
    - `getLogs()`: String → Mengembalikan semua log sebagai String, satu log per baris. Contoh
      - `clearLogs()`: Mengosongkan logs.
  - Gunakan `java.time.LocalDateTime` dan `DateTimeFormatter` untuk timestamp.
  - Logger harus menyimpan data peminjaman dan pengembalian
- **Library**:
  - Attribute:
    - items (List)
    - members (List)
    - logger (LibraryLogger).
  - Method:
    - `addItem(LibraryItem item)`: String → Tambah item ke items. Mengembalikan "[title] berhasil ditambahkan"
    - `findItemById(int itemId)`: LibraryItem → Kembalikan LibraryItem atau exception `NoSuchElementException`.

- `getLibraryStatus()`: Kembalikan String dengan daftar semua item (deskripsi dan status: "Dipinjam" atau "Tersedia").

```
+-----+-----+-----+
| ID   | Judul   | Status   |
+-----+-----+-----+
| 123  | Buku 2  | Dipinjam |
| 124  | Buku 1  | Tersedia |
+-----+-----+-----+
```

- `getAllLogs()`: Kembalikan log dari logger.

Silahkan tambahkan method lain apabila diperlukan.

3. Buat CLI dari sistem di atas.

```
=== Sistem Manajemen Perpustakaan ===
1. Tambah Item
2. Tambah Anggota
3. Pinjam Item
4. Kembalikan Item
5. Lihat Status Perpustakaan
6. Lihat Log Aktivitas
7. Lihat Item yang Dipinjam Anggota
8. Keluar
```