

Nama : Nurul Aulia Dewi
Kelas : QE

Soal Minggu Ke - 4 (17 Maret 2024)

Soal Eksplorasi

▼ Soal Eksplorasi (20)

1. Buatlah sebuah program yang dapat melakukan operasi CRUD (Create, Read, Update dan Delete) dengan kriteria sebagai berikut:
 - a. Program dapat menyimpan data buku dalam bentuk class. Nanti terdapat sebuah class untuk menyimpan data buku dengan nama Book.
 - b. Data buku disimpan dengan menggunakan penyimpanan lokal seperti ArrayList.
 - c. Menggunakan input dari user. Input dapat menggunakan class Scanner atau BufferedReader.
 - d. Data buku terdiri dari ID, judul, penulis dan kategori.
 - e. Data buku menggunakan UUID sebagai ID.
 - f. Terdapat operasi membuat data buku baru.
 - g. Terdapat operasi mendapatkan semua data buku.
 - h. Terdapat operasi mendapatkan data buku berdasarkan ID.
 - i. Terdapat operasi mengubah data buku berdasarkan ID.
 - j. Terdapat operasi menghapus data buku berdasarkan ID.
 - k. Contoh hasil program adalah sebagai berikut:

Menu program.

```
Welcome to book management app
Please choose your menu:
1. Create a new book
2. Get all books
3. Get book by ID
4. Update book
5. Delete book
6. Exit
```

1.

Membuat data baru.

```
ALL BOOKS===
=====
ID: 9c5665fd-1ae2-430d-bed0-ed1b1670c60a
Title: java basics
Author: enki gilbert
Category: tech
=====
ID: 02e51ab6-67cc-41c9-b2e4-cb7c23b95c4a
Title: docker basics
Author: ray krieger
Category: devops
=====
```

Mendapatkan data berdasarkan ID.

```
[...]
Enter book ID: 02e51ab6-67cc-41c9-b2e4-cb7c23b95c4a
===
ID: 02e51ab6-67cc-41c9-b2e4-cb7c23b95c4a
Title: docker basics
Author: ray krieger
Category: devops
===
[...]
```

Mengubah data.

```
[...]
Enter book ID:
02e51ab6-67cc-41c9-b2e4-cb7c23b95c4a
Enter title:
kubernetes basics
Enter author:
ryo watanabe
Enter category:
devops
book updated!
===
ID: 02e51ab6-67cc-41c9-b2e4-cb7c23b95c4a
Title: kubernetes basics
Author: ryo watanabe
Category: devops
===
[...]
```

Menghapus data.

```
[...]
Enter book ID: 02e51ab6-67cc-41c9-b2e4-cb7c23b95c4a
book deleted!
```

Keluar dari program.

```
[...]
Bye....
```

jawab

```
no usages
public String getTitle() {
    return title;
}

1 usage
public void setTitle(String title) {
    this.title = title;
}

no usages
public String getAuthor() {
    return author;
}

1 usage
public void setAuthor(String author) {
    this.author = author;
}

no usages
public String getCategory() {
    return category;
}

1 usage
public void setCategory(String category) {
    this.category = category;
}
1.

2 usages
public String getInfo() {
    return "ID: " + id.toString() + "\nTitle: " + title + "\nAuthor: " + author + "\nCategory: " + category;
}
}
```

```

package tugas;

import java.util.ArrayList;
import java.util.Scanner;
import java.util.UUID;

public class BookManager {
    private static ArrayList<Book> books = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.println("Welcome to book management app");
        System.out.println("Please choose your menu:");
        System.out.println("1. Create a new book");
        System.out.println("2. Get all books");
        System.out.println("3. Get book by ID");
        System.out.println("4. Update book");
        System.out.println("5. Delete book");
        System.out.println("6. Exit");
    }

    int choice;
    do {
        System.out.print("\nEnter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        switch (choice) {
            case 1:
                createBook();
                break;
            case 2:
                getAllBooks();
                break;
            case 3:
                getBookById();
                break;
            case 4:
                updateBook();
                break;
            case 5:
                deleteBook();
                break;
            case 6:
                System.out.println("Bye...");
                break;
            default:
                System.out.println("Invalid choice. Please choose again.");
        }
    } while (choice != 6);
}

private static void createBook() {
    System.out.println("\nCreating a new book");
    System.out.print("Enter title: ");
    String title = scanner.nextLine();
    System.out.print("Enter author: ");
    String author = scanner.nextLine();
    System.out.print("Enter category: ");
    String category = scanner.nextLine();

    Book newBook = new Book(title, author, category);
    books.add(newBook);

    System.out.println("Book created!");
}

private static void getAllBooks() {
    System.out.println("\n==== ALL BOOKS ====");
    for (Book book : books) {
        System.out.println("====\n" + book.getInfo() + "\n====");
    }
}

```

```

private static void getBookById() {
    System.out.println("\nGetting book by ID");
    System.out.print("Enter book ID: ");
    String idString = scanner.nextLine();
    UUID id = UUID.fromString(idString);

    Book foundBook = findBookById(id);
    if (foundBook != null) {
        System.out.println("==\n" + foundBook.getInfo() + "\n==");
    } else {
        System.out.println("Book not found.");
    }
}

3 usages
private static Book findBookById(UUID id) {
    for (Book book : books) {
        if (book.getId().equals(id)) {
            return book;
        }
    }
    return null;
}

private static void updateBook() {
    System.out.println("\nUpdating book");
    System.out.print("Enter book ID: ");
    String idString = scanner.nextLine();
    UUID id = UUID.fromString(idString);

    Book foundBook = findBookById(id);
    if (foundBook != null) {
        System.out.print("Enter title: ");
        String title = scanner.nextLine();
        System.out.print("Enter author: ");
        String author = scanner.nextLine();
        System.out.print("Enter category: ");
        String category = scanner.nextLine();

        foundBook.setTitle(title);
        foundBook.setAuthor(author);
        foundBook.setCategory(category);

        System.out.println("Book updated!");
    } else {
        System.out.println("Book not found.");
    }
}

1 usage
private static void deleteBook() {
    System.out.println("\nDeleting book");
    System.out.print("Enter book ID: ");
    String idString = scanner.nextLine();
    UUID id = UUID.fromString(idString);

    Book foundBook = findBookById(id);
    if (foundBook != null) {
        books.remove(foundBook);
        System.out.println("Book deleted!");
    } else {
        System.out.println("Book not found.");
    }
}

```

(Codingan soal eksplorasi)

```
/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=54592:/Applications/IntelliJ IDEA CE.app/Contents/bin
```

```
Welcome to book management app
Please choose your menu:
1. Create a new book
2. Get all books
3. Get book by ID
4. Update book
5. Delete book
6. Exit

Enter your choice: 1

Creating a new book
Enter title: Hujan
Enter author: Tere liye
Enter category: Fiction
Book created!

Enter your choice: 2

==== ALL BOOKS ====
===
ID: a23590ed-9847-4dd7-b33a-df6d96d0c9f2
Title: Hujan
Author: Tere liye
Category: Fiction
===

Enter your choice: 3

Getting book by ID
Enter book ID: a23590ed-9847-4dd7-b33a-df6d96d0c9f2
===
ID: a23590ed-9847-4dd7-b33a-df6d96d0c9f2
Title: Hujan
Author: Tere liye
Category: Fiction
===

Enter your choice: 4

Updating book
Enter book ID: a23590ed-9847-4dd7-b33a-df6d96d0c9f2
Enter title: hujan di bulan juni
Enter author: tere liye
Enter category: fiction
Book updated!

Enter your choice: 5

Deleting book
Enter book ID: a23590ed-9847-4dd7-b33a-df6d96d0c9f2
Book deleted!

Enter your choice: 6
Bye...
```

(output s