

Nama : Ahmad Franklyn Bima Aquilla  
Kelas : SIB 1F – 01  
NIM : 2341760027  
Matkul : Praktikum Algoritma dan Struktur Data  
Github : <https://github.com/JustQuill25/Kuis2ASDTeori>

## KUIS 2 ASD

### Deskripsi Kuis 2 Algoritma dan Struktur Data

#### 1. Spesifikasi program:

Buat program untuk melakukan **penyimpanan dan operasi data (tambah, delete, dan baca/ akses data)** dengan memanfaatkan struktur data yang sudah diajarkan. Anda diminta mengimplementasikan **dua macam struktur data** dengan ketentuan yang **satu wajib mengimplementasikan Java Collection**. Sedangkan **program yang satunya bebas memakai struktur data yang mana (bisa dipilih diantara linked list, double linked list, stack, queue, tree, dsb)**. Beberapa ketentuan atau spesifikasi program:

- Data yang disimpan dan diolah adalah object (**bukan tipe data primitif**) dengan **minimal 3 atribut**. Misal **object Mahasiswa dengan atribut: nim, nama, tglLahir**.
- Object apa yang disimpan dan diolah, dibebaskan untuk setiap mahasiswa
- Wajib ada konstruktor berparameter selain konstruktor default
- Operasi yang wajib diimplementasikan: **menambahkan elemen, menghapus elemen, membaca/ mengakses elemen tertentu, mencetak semua elemen**
- Operasi tambahan (opsional): sorting, searching.
- Fungsi utama berisi menu yang berisi pilihan untuk mengoperasikan masing-masing method/ fungsi/ operasi pada data
- Program **sudah diisikan minimal 4 data/ object di awal**, agar memudahkan saat mengetes program
- Fungsi utama (main) harus dalam file java terpisah
- **Jumlah class bebas**, sesuai kebutuhan, akan tetapi harus jelas pemisahannya. Satu class diimplementasikan dalam satu class java
- Setiap orang membuat dua macam implementasi:
  - o **Satu program dengan struktur data bebas (bisa terdiri dari bbrp class java)**
  - o **Satu program mengimplementasikan Java Collection (bisa terdiri dari bbrp class java)**

#### 2. Buat laporan berisi:

- a. Deskripsi program: penjelasan program yang Anda buat. Tuliskan se jelas mungkin
- b. Screen shoot hasil running program dan penjelasannya, tidak hanya gambar
- c. Tuliskan link github repository

#### 3. Penamaan dan pengumpulan file

- Program java:
  - o Penamaan class bebas, akan tetapi menunjukkan isi dari program. Jangan lupa tambahkan no absen di belakangnya. Misal: nasabah\_01.java, mhsLinkedList\_01.java, mainBuku\_01.java
  - o Semua file dijadikan satu folder dan dikirim dalam bentuk \*.rar/ \*.zip
  - o Penamaan file \*.rar/ \*.zip: <no\_absen>\_<nama\_pendek>\_kuis2
- Laporan:
  - o Dikumpulkan dalam bentuk file pdf
  - o Penamaan: <no\_absen>\_<nama\_pendek>\_kuis2.pdf
- Dikumpulkan dengan cara upload pada LMS pada course: Praktikum Algoritma dan Struktur Data pada topic: Quiz 2

## [ Tidak Menggunakan Java Collection ]

### 1. Class Buku01

- Fungsi dan Tugas Class Buku01. Kelas Buku01 digunakan untuk merepresentasikan sebuah buku dengan atribut isbn, judul, dan pengarang. Konstruktor digunakan untuk inialisasi nilai atribut, sedangkan getter dan setter digunakan untuk mengakses dan mengubah nilai atribut tersebut. Metode toString dioverride untuk memberikan representasi String dari objek Buku01.

#### 1. Konstruktor Untuk Inialisasi Objek

```
public Buku01(String isbn, String judul, String pengarang) {  
    this.isbn = isbn;  
    this.judul = judul;  
    this.pengarang = pengarang;  
}
```

- **Deskripsi** : Konstruktor ini digunakan untuk membuat objek Buku01 dengan nilai awal isbn, judul, dan pengarang sesuai dengan parameter yang diberikan.

#### 2. Getter dan Setter untuk Atribut

```
public String getIsbn() {  
    return isbn;  
}  
  
public void setIsbn(String isbn) {  
    this.isbn = isbn;  
}  
  
public String getJudul() {  
    return judul;  
}  
  
public void setJudul(String judul) {  
    this.judul = judul;  
}  
  
public String getPengarang() {  
    return pengarang;  
}  
  
public void setPengarang(String pengarang) {  
    this.pengarang = pengarang;  
}
```

- **Deskripsi** : Getter dan setter digunakan untuk mengakses dan mengubah nilai dari atribut isbn, judul, dan pengarang pada objek Buku01. Hal ini memungkinkan manipulasi data buku dari luar kelas Buku01 tanpa melanggar prinsip enkapsulasi.

#### 3. Menghasilkan Representasi String dari Objek Buku01

```
@Override  
public String toString() {  
    return "Buku01 {" +  
        " isbn = '" + isbn + '\'' +  
        ", judul = '" + judul + '\'' +  
        ", pengarang = '" + pengarang + '\'' +  
        '}';  
}
```

- **Deskripsi** : Metode toString() dioverride untuk menghasilkan representasi String dari objek Buku01. Ini memungkinkan objek Buku01 dicetak atau ditampilkan dalam format yang telah ditentukan.

## 2. Class Node01

- Class Node01 merepresentasikan sebuah node dalam struktur data Binary Search Tree (BST). Setiap node memiliki objek Buku01 sebagai datanya, serta dua referensi ke node anak (kiri dan kanan). Konstruktor digunakan untuk inisialisasi node dengan data buku dan mengatur kedua referensi anak (left dan right) menjadi null.

### 1. Konstruktor

```
public Node01(Buku01 data) {  
    this.data = data;  
    left = right = null;  
}
```

- **Deskripsi** : Konstruktor ini digunakan untuk membuat objek Node01 yang merupakan bagian dari struktur data Binary Search Tree (BST). data adalah objek Buku01 yang disimpan di dalam node, sedangkan left dan right adalah referensi ke node anak kiri dan kanan yang awalnya diatur menjadi null.

## 3. Class BinarySearchTree01

- Class BinarySearchTree01 mengimplementasikan struktur data Binary Search Tree (BST) untuk menyimpan dan mengelola objek Buku01. root adalah referensi ke node akar BST. Konstruktor menginisialisasi root menjadi null, menandakan BST awalnya kosong. Metode tambahBuku digunakan untuk menambahkan buku ke dalam BST secara rekursif berdasarkan nilai ISBN. Metode hapusBuku menghapus buku dari BST berdasarkan ISBN, juga menggunakan rekursi. Metode tampilkanBuku menampilkan seluruh buku dalam urutan inorder, yaitu dari node terkiri ke node paling kanan.

### 1. Konstruktor

```
public BinarySearchTree01() {  
    root = null;  
}
```

- **Deskripsi** : Konstruktor ini digunakan untuk membuat objek BinarySearchTree01, yang merepresentasikan Binary Search Tree (BST). root awalnya diatur menjadi null, menunjukkan bahwa BST awalnya kosong.

### 2. Menambahkan Method ( tambahBuku )

```
public void tambahBuku(Buku01 data) {  
    root = tambahRekursif(root, data);  
}
```

- **Deskripsi** : Metode ini digunakan untuk menambahkan objek Buku01 ke dalam BST. Jika BST masih kosong, objek data akan menjadi akar (root). Jika tidak, metode rekursif tambahRekursif dipanggil untuk menemukan tempat yang tepat untuk menambahkan data.

### 3. Menambahkan Method ( tambahRekursif )

```
private Node01 tambahRekursif(Node01 root, Buku01 data) {
    if (root == null) {
        root = new Node01(data);
        return root;
    }

    if (data.getIsbn().compareTo(root.data.getIsbn()) < 0)
        root.left = tambahRekursif(root.left, data);
    else if (data.getIsbn().compareTo(root.data.getIsbn()) > 0)
        root.right = tambahRekursif(root.right, data);

    return root;
}
```

- **Deskripsi:** Metode rekursif ini menambahkan objek Buku01 ke dalam BST. Jika root adalah null, maka root baru dibuat dengan data sebagai nilai. Jika tidak, metode ini secara rekursif mencari posisi yang tepat berdasarkan perbandingan nilai ISBN data dengan root.data, dan menempatkan data ke kiri atau kanan sesuai dengan aturan BST.

### 4. Menambahkan Method ( hapusBuku )

```
public void hapusBuku(String isbn) {
    root = hapusRekursif(root, isbn);
}
```

- **Deskripsi :** Metode ini digunakan untuk menghapus buku dari BST berdasarkan ISBN. Jika buku ditemukan, metode rekursif hapusRekursif dipanggil untuk melakukan penghapusan.

### 5. Menambahkan Method ( hapusRekursif )

```
private Node01 hapusRekursif(Node01 root, String isbn) {
    if (root == null) return root;

    if (isbn.compareTo(root.data.getIsbn()) < 0)
        root.left = hapusRekursif(root.left, isbn);
    else if (isbn.compareTo(root.data.getIsbn()) > 0)
        root.right = hapusRekursif(root.right, isbn);
    else {
        if (root.left == null)
            return root.right;
        else if (root.right == null)
            return root.left;

        root.data = minValue(root.right);
        root.right = hapusRekursif(root.right, root.data.getIsbn());
    }

    return root;
}
```

- **Deskripsi :** Metode rekursif ini menghapus buku dari BST berdasarkan ISBN. Jika root adalah null, artinya buku tidak ditemukan. Jika buku ditemukan, kondisi diproses berdasarkan hubungan nilai ISBN, dengan menggeser node yang tepat ke kiri atau kanan setelah penghapusan.

#### 6. Menambahkan Method ( minValue )

```
private Buku01 minValue(Node01 root) {  
    Buku01 minv = root.data;  
    while (root.left != null) {  
        minv = root.left.data;  
        root = root.left;  
    }  
    return minv;  
}
```

- **Deskripsi** : Metode ini mencari dan mengembalikan nilai minimum dalam subtree yang diberikan, dimulai dari root. Ini digunakan dalam proses penghapusan node yang memiliki dua anak.

#### 7. Menambahkan Method ( tampilkanBuku )

```
public void tampilkanBuku() {  
    tampilkanRekursif(root);  
}
```

- **Deskripsi** : Metode ini menampilkan semua buku dalam BST dalam urutan inorder. Ini dilakukan dengan memanggil metode rekursif tampilkanRekursif yang dimulai dari root.

#### 8. Menambahkan Method ( tampilkanRekursif )

```
private void tampilkanRekursif(Node01 root) {  
    if (root != null) {  
        tampilkanRekursif(root.left);  
        System.out.println(root.data);  
        tampilkanRekursif(root.right);  
    }  
}
```

- **Deskripsi** : Metode rekursif ini digunakan untuk menampilkan semua buku dalam BST dalam urutan inorder. Prosesnya dimulai dari node ter kiri (root.left), kemudian mencetak node saat ini (root.data), dan melanjutkan ke node kanan (root.right).

### 4. Class MainBST01

- Class MainBST01 bertanggung jawab untuk mengatur interaksi dengan pengguna melalui menu interaktif. Ini memanfaatkan struktur data BST (BinarySearchTree01) untuk menyimpan dan mengelola data buku (Buku01) dengan operasi dasar seperti penambahan, penghapusan, dan pencetakan. Program ini menunjukkan implementasi praktis dari BST untuk penyimpanan dan manajemen data dalam konteks perpustakaan atau sistem informasi buku.

#### 1. Deklarasi Kelas dan Objek

```
public class MainBST01 {  
    private static BinarySearchTree01 bst = new BinarySearchTree01();
```

- **Deskripsi** : Berfungsi untuk mendeklarasikan kelas 'MainBST01'. Serta membuat objek 'bst' dari kelas 'BinarySearchTree' untuk menyimpan dan mengelola data buku

## 2. Menambah Method Main Serta 4 Data Awal

```
Run | Debug
public static void main(String[] args) {
    // Tambahkan 4 data di awal
    bst.tambahBuku(new Buku01(isbn:"01", judul:"Algoritma Dan Struktur Data", pengarang:"Bu Triana"));
    bst.tambahBuku(new Buku01(isbn:"02", judul:"Basis Data", pengarang:"Pak Farid"));
    bst.tambahBuku(new Buku01(isbn:"03", judul:"Sistem Operasi", pengarang:"Bu Meyti"));
    bst.tambahBuku(new Buku01(isbn:"04", judul:"Analisis Proses Bisnis", pengarang:"Pak Hendra"));
}
```

- **Deskripsi** : Berfungsi sebagai metode utama yang akan dieksekusi saat program berjalan serta menambahkan 4 data di awal program

## 3. Inisialisasi Scanner

```
import java.util.Scanner;

Scanner scanner = new Scanner(System.in);
```

- **Deskripsi** : Berfungsi untuk membuat objek 'scanner'. Untuk membaca input dari pengguna

## 4. Loop 'do-while'

```
int pilihan;
do {
    System.out.println(x:"Menu:");
    System.out.println(x:"1. Tambah Buku");
    System.out.println(x:"2. Hapus Buku");
    System.out.println(x:"3. Tampilkan Buku");
    System.out.println(x:"4. Keluar");
    System.out.print(s:"Pilih : ");
    pilihan = scanner.nextInt();
    scanner.nextLine();
}
```

- **Deskripsi** : Berfungsi memulai Loop untuk menampilkan menu interaktif kepada pengguna dan menjalankan aksi sesuai pilihan yang dibuat pengguna. Serta menampilkan daftar opsi kepada pengguna dan membaca pilihan pengguna dari input

## 5. Menambah Statement 'switch' untuk Memproses Pilihan

```
switch (pilihan) {
    case 1:
        tambahBuku(scanner);
        break;
    case 2:
        hapusBuku(scanner);
        break;
    case 3:
        tampilkanBuku();
        break;
    case 4:
        System.out.println(x:"Program Selesai. Semoga Harimu Menyenangkan :");
        break;
    default:
        System.out.println(x:"Pilihan tidak valid.");
}
} while (pilihan != 4);
scanner.close();
```

- **Deskripsi** : Untuk memproses pilihan pengguna menggunakan pernyataan switch untuk menjalankan aksi sesuai dengan pilihan yang dibuat pengguna:
- case 1: Memanggil tambahBuku(scanner) untuk menambahkan buku baru.
- case 2: Memanggil hapusBuku(scanner) untuk menghapus buku berdasarkan ISBN.
- case 3: Memanggil tampilkanBuku() untuk menampilkan semua buku yang tersimpan.
- case 4: Mengakhiri program dengan menampilkan pesan dan keluar dari loop.
- **Fungsi**: Menutup objek Scanner setelah selesai menggunakan input dari pengguna untuk menghindari kebocoran sumber daya.



## 6. Menambahkan Methode 'tambahBuku'

```
private static void tambahBuku(Scanner scanner) {
    System.out.print(s:"ISBN : ");
    String isbn = scanner.nextLine();
    System.out.print(s:"Judul : ");
    String judul = scanner.nextLine();
    System.out.print(s:"Pengarang : ");
    String pengarang = scanner.nextLine();
    bst.tambahBuku(new Buku01(isbn, judul, pengarang));
}
```

- **Deskripsi** : Berfungsi menambahkan buku baru ke dalam BST dengan memanggil tambahBuku pada objek bst.

## 7. Menambahkan Methode 'hapusBuku'

```
private static void hapusBuku(Scanner scanner) {
    System.out.print(s:"ISBN yang akan dihapus : ");
    String isbn = scanner.nextLine();
    bst.hapusBuku(isbn);
}
```

- **Deskripsi** : Berfungsi menghapus buku dari BST berdasarkan ISBN dengan memanggil hapusBuku pada objek bst.

## 8. Menambahkan Methode 'tampilBuku'

```
private static void tampilkanBuku() {
    bst.tampilkanBuku();
}
```

- **Deskripsi** : Berfungsi Menampilkan semua buku yang disimpan dalam BST dengan memanggil tampilkanBuku pada objek bst.

## 5. Output Program Pertama

### ➤ Tampilan Awal

```
Menu:
1. Tambah Buku
2. Hapus Buku
3. Tampilkan Buku
4. Keluar
```

### > Input '3'

```
Pilih : 3
Buku01 { isbn = '02', judul = 'Basis Data', pengarang = 'Pak Farid'}
Buku01 { isbn = '03', judul = 'Sistem Operasi', pengarang = 'Bu Meyti'}
Buku01 { isbn = '04', judul = 'Analisis Proses Bisnis', pengarang = 'Pak Hendra'}
Buku01 { isbn = '05', judul = 'Matematika Lanjut', pengarang = 'Bu Deasy'}
```

### ➤ Input '1'

```
Pilih : 1
ISBN : 05
Judul : Matematika Lanjut
Pengarang : Bu Deasy
Menu:
1. Tambah Buku
2. Hapus Buku
3. Tampilkan Buku
4. Keluar
```

### > Input '4'

```
Pilih : 4
Program Selesai. Semoga Harimu Menyenangkan :)
PS D:\Algoritma dan Struktur Data\Kuis2>
```

### ➤ Input '2'

```
Pilih : 2
ISBN yang akan dihapus : 01
```

## [ Menggunakan Java Collection ]

### 1. Class Buku01

- Class Buku01 merepresentasikan entitas buku dengan atribut isbn, judul, dan pengarang. Atribut-atribut ini diinisialisasi melalui konstruktor dan dapat diakses serta dimodifikasi melalui metode getter dan setter. Selain itu, kelas ini mengoverride metode toString untuk menyediakan representasi string yang bermakna dari objek buku, memudahkan pencetakan informasi buku dengan format yang mudah dibaca. Class ini berfungsi untuk menyediakan struktur data yang rapi dan mudah diakses untuk menyimpan informasi buku.

#### 1. Deklarasi Kelas dan Atribut

```
public class Buku01 {  
    private String isbn;  
    private String judul;  
    private String pengarang;
```

- **Deskripsi** : Berfungsi untuk Mendeklarasikan kelas Buku01 dengan tiga atribut: isbn, judul, dan pengarang.

#### 2. Konstruktor Objek 'Buku01'

```
public Buku01(String isbn, String judul, String pengarang) {  
    this.isbn = isbn;  
    this.judul = judul;  
    this.pengarang = pengarang;  
}
```

- **Deskripsi** : Berfungsi sebagai Konstruktor untuk menginisialisasi objek Buku01 dengan nilai ISBN, judul, dan pengarang yang diberikan

#### 3. Konstruktor Objek 'Buku01'

```
public Buku01(String isbn, String judul, String pengarang) {  
    this.isbn = isbn;  
    this.judul = judul;  
    this.pengarang = pengarang;  
}
```

- **Deskripsi** : Berfungsi sebagai Konstruktor untuk menginisialisasi objek Buku01 dengan nilai ISBN, judul, dan pengarang yang diberikan

#### 4. Menambah Method 'getIsbn'

```
public String getIsbn() {  
    return isbn;  
}
```

- **Deskripsi** : Berfungsi menambahkan Metode getter untuk mendapatkan nilai ISBN dari objek 'Buku01'



#### 5. Menambah Method 'setIsbn'

```
public void setIsbn(String isbn) {  
    this.isbn = isbn;  
}
```

- **Deskripsi** : Berfungsi menambahkan Metode setter untuk mengubah nilai ISBN dari objek 'Buku01'

#### 6. Menambah Method 'getJudul'

```
public String getJudul() {  
    return judul;  
}
```

- **Deskripsi** : Berfungsi menambahkan Metode getter untuk mendapatkan nilai judul dari objek 'Buku01'

#### 7. Menambah Method 'setJudul'

```
public void setJudul(String judul) {  
    this.judul = judul;  
}
```

- **Deskripsi** : Berfungsi menambahkan Metode setter untuk mengubah nilai dari judul objek 'Buku01'

#### 8. Menambah Method 'getPengarang'

```
public String getPengarang() {  
    return pengarang;  
}
```

- **Deskripsi** : Berfungsi menambahkan Metode getter untuk mendapatkan nilai dari pengarang dari objek 'Buku01'

#### 9. Menambah Method 'toString'

```
@Override  
public String toString() {  
    return "Buku {" +  
        "isbn =" + isbn + '\'' +  
        ", judul =" + judul + '\'' +  
        ", pengarang =" + pengarang + '\'' +  
        '}';  
}
```

- **Deskripsi** : Berfungsi menambahkan Override metode 'toString' untuk memperbaiki representasi string dari objek 'Buku01', yang mencakup informasi ISBN, Judul, dan pengarang

## 2. Class MainArrayList01

- Class MainArrayList01 berfungsi sebagai aplikasi utama yang mengelola daftar buku menggunakan struktur data ArrayList. Program ini menawarkan fitur untuk menambah, menghapus, dan menampilkan buku melalui menu interaktif yang dioperasikan oleh pengguna. Dengan menyediakan antarmuka menu yang sederhana, pengguna dapat dengan mudah berinteraksi dengan program untuk menambah buku baru, menghapus buku berdasarkan ISBN, dan menampilkan semua buku yang ada dalam daftar. Program ini menunjukkan penggunaan koleksi Java (ArrayList) untuk menyimpan dan mengelola data buku secara dinamis dan efisien.

### 1. Mengimport Class

```
import java.util.ArrayList;  
import java.util.Scanner;
```

- **Deskripsi** : Berfungsi untuk Mengimport kelas ArrayList dan Scanner dari pustaka Java untuk digunakan dalam program.

### 2. Mendeklarasikan class 'MainArrayList'

```
public class MainArrayList01 {  
    private static ArrayList<Buku01> daftarBuku = new ArrayList<>();
```

- **Deskripsi** : Berfungsi untuk Mendeklarasikan kelas MainArrayList01 dan membuat objek ArrayList untuk menyimpan daftar buku.

### 3. Menambah Method 'main'

```
Run | Debug  
public static void main(String[] args) {  
    // Tambahkan 4 data di awal  
    daftarBuku.add(new Buku01(isbn:"01", judul:"Algoritma Dan Struktur Data", pengarang:"Bu Triana"));  
    daftarBuku.add(new Buku01(isbn:"02", judul:"Basis Data", pengarang:"Pak Farid"));  
    daftarBuku.add(new Buku01(isbn:"03", judul:"Sistem Operasi", pengarang:"Bu Meyti"));  
    daftarBuku.add(new Buku01(isbn:"04", judul:"Analisis Proses Bisnis", pengarang:"Pak Hendra"));
```

- **Deskripsi** : Berfungsi sebagai Metode main adalah titik masuk untuk menjalankan program. Bagian ini menambahkan empat buku awal ke dalam daftar.

### 4. Membuat Scanner

```
import java.util.Scanner;  
  
Scanner scanner = new Scanner(System.in);
```

- **Deskripsi** : Berfungsi untuk Membuat objek Scanner untuk membaca input dari pengguna.

## 5. Mendeklarasikan variabel 'pilihan'

```
int pilihan;
do {
    System.out.println(x:"Menu:");
    System.out.println(x:"1. Tambah Buku");
    System.out.println(x:"2. Hapus Buku");
    System.out.println(x:"3. Tampilkan Buku");
    System.out.println(x:"4. Keluar");
    System.out.print(s:"Pilih : ");
    pilihan = scanner.nextInt();
    scanner.nextLine();
}
```

- **Deskripsi** : Berfungsi untuk Mendeklarasikan variabel pilihan untuk menyimpan pilihan menu pengguna dan menampilkan menu kepada pengguna. Setelah pengguna memasukkan pilihan, nilai tersebut dibaca dan disimpan dalam variabel pilihan.

## 6. Membuat Struktur 'switch'

```
switch (pilihan) {
    case 1:
        tambahBuku(scanner);
        break;
    case 2:
        hapusBuku(scanner);
        break;
    case 3:
        tampilkanBuku();
        break;
    case 4:
        System.out.println(x:"Program Selesai. Semoga Harimu Menyenangkan :)");
        break;
    default:
        System.out.println(x:"Pilihan tidak valid.");
}
} while (pilihan != 4);
scanner.close();
```

- **Deskripsi** : Struktur switch untuk memproses pilihan pengguna :
  - **case 1**: Memanggil metode tambahBuku.
  - **case 2**: Memanggil metode hapusBuku.
  - **case 3**: Memanggil metode tampilkanBuku.
  - **case 4**: Menampilkan pesan dan keluar dari loop.
  - **default**: Menangani pilihan yang tidak valid.
  - **do-while loop**: Menjalankan loop selama pilihan tidak sama dengan 4.
  - **scanner.close()**: Menutup objek Scanner.

## 7. Menambah Method 'tambahBuku'

```
private static void tambahBuku(Scanner scanner) {
    System.out.print(s:"ISBN : ");
    String isbn = scanner.nextLine();
    System.out.print(s:"Judul : ");
    String judul = scanner.nextLine();
    System.out.print(s:"Pengarang : ");
    String pengarang = scanner.nextLine();
    daftarBuku.add(new Buku01(isbn, judul, pengarang));
}
```

- **Deskripsi** : Berfungsi untuk menambah buku ke dalam daftar. Serta menerima input ISBN, judul dan pengarang dari pengguna

### 8. Menambah Method 'hapusBuku'

```
private static void hapusBuku(Scanner scanner) {  
    System.out.print(s:"ISBN yang akan dihapus : ");  
    String isbn = scanner.nextLine();  
    daftarBuku.removeIf(buku -> buku.getIsbn().equals(isbn));  
}
```

- **Deskripsi** : Berfungsi untuk menghapus buku dari daftar ISBN. Serta menerima input ISBN dari pengguna

### 9. Menambah Method 'tampilkanBuku'

```
private static void tampilkanBuku() {  
    for (Buku01 buku : daftarBuku) {  
        System.out.println(buku);  
    }  
}
```

- **Deskripsi** : Berfungsi untuk menampilkan semua buku dalam daftar serta mencetak setiap informasi buku menggunakan metode 'toString' dari objek 'Buku01'

## 3. Output Program Kedua

#### ➤ Tampilan Awal

```
Menu:  
1. Tambah Buku  
2. Hapus Buku  
3. Tampilkan Buku  
4. Keluar
```

#### ➤ Input '1'

```
Pilih : 1  
ISBN : 10  
Judul : Kewarganegaraan  
Pengarang : Bu Wida
```

#### > Input '4'

```
Pilih : 4  
Program Selesai. Semoga Harimu Menyenangkan :)  
PS D:\Algoritma dan Struktur Data\Kuis2>
```

#### ➤ Input '2'

```
Pilih : 2  
ISBN yang akan dihapus : 03  
Menu:  
1. Tambah Buku  
2. Hapus Buku  
3. Tampilkan Buku  
4. Keluar
```

#### ➤ Input '3'

```
Pilih : 3  
Buku {isbn = '01', judul = 'Algoritma Dan Struktur Data', pengarang = 'Bu Triana'  
Buku {isbn = '02', judul = 'Basis Data', pengarang = 'Pak Farid'}  
Buku {isbn = '04', judul = 'Analisis Proses Bisnis', pengarang = 'Pak Hendra'}  
Buku {isbn = '10', judul = 'Kewarganegaraan', pengarang = 'Bu Wida'}
```