**PROGRAM RAK BUKU**

**PRAKTEK ALGORITMA PEMROGRAMAN**

**UJIAN AKHIR SEMESTER**

Dosen Pengampu : Bonita Destiana, S.Pd., M.Pd.



Oleh :

| M. Zafif Hilmi Al-hadi | 22520241005 |
| --- | --- |
| Miftakhul Huda Dwi Setiawan | 22520241023 |
| Nastiti Chusna Haliza | 22520241025 |
| Anggata Sakti Bhatara Herindra | 22520244024 |

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI YOGYAKARTA**

**2023**

**DAFTAR ISI**

[A. DESKRIPSI 3](#_heading=h.jm06uqohjpbb)

[B. FLOWCHART 3](#_heading=h.cvnpehnwt8a6)

[C. KODE PROGRAM 5](#_heading=h.bctspyb507ch)

[D. HASIL PROGRAM 22](#_heading=h.wot4r7wia1fs)

[E. JOBDESK ANGGOTA 24](#_heading=h.89l4f0n5t84y)

[F. KESIMPULAN 25](#_heading=h.ml1ln8sacq02)

# 

# DESKRIPSI

Aplikasi Rak Buku adalah sebuah aplikasi sederhana yang ditulis dalam bahasa pemrograman Java. Aplikasi ini dirancang untuk mengelola informasi buku, termasuk menampilkan daftar buku, menambahkan buku baru, menghapus buku, dan memperbarui informasi buku. Setiap buku direpresentasikan oleh kelas Buku yang memiliki atribut seperti nomor, judul, penulis, tahun terbit, dan status (apakah buku sudah dibaca atau belum). Selain itu, program menyediakan fungsionalitas pengurutan daftar buku berdasarkan tahun terbit.

Aplikasi ini memberikan pengguna opsi untuk melakukan berbagai tindakan terkait buku melalui menu interaktif yang ditampilkan pada konsol. Pengguna dapat melihat informasi spesifik tentang suatu buku, menambahkan buku baru dengan menginputkan detailnya, memperbarui informasi buku yang sudah ada, menghapus buku berdasarkan nomor, serta menampilkan daftar buku yang diurutkan berdasarkan tahun terbit, serta pengguna dapat memberikan status sudah dibaca/belum dibaca pada setiap buku.

Secara keseluruhan, aplikasi ini memberikan fungsionalitas dasar untuk pengelolaan dan pemeliharaan koleksi buku, membuatnya berguna untuk pengguna yang ingin mengatur dan melacak informasi buku secara terstruktur.

# FLOWCHART

* 1. Utama
     1. Class Buku
     2. Class rakBuku
  2. Fungsi
     1. tampilkanBuku
     2. lihatInformasiBuku
     3. tambahBuku
     4. perbaruiBuku
     5. hapusBuku
  3. Sub Fungsi
     1. temukanBuku
     2. temukanIndeksBuku

Semua Flowchart dapat diakses pada tautan berikut: <https://unyku.id/FlowchartKel2alpro>

# KODE PROGRAM

Program rak buku yang diimplementasikan menggunakan bahasa pemrograman Java. Program menyediakan fungsionalitas manajemen buku, seperti memungkinkan pengguna untuk menampilkan, menambah, memperbarui, dan menghapus informasi buku dalam rak. Program ini menggunakan konsep ArrayList untuk menyimpan objek-objek buku dan menyediakan tampilan sederhana melalui terminal untuk berinteraksi dengan rak buku tersebut. Berikut penjelasan programnya.

* 1. Library yang digunakan

| **Source Code** |
| --- |
| **import java.util.ArrayList;**  **import java.util.Collections;**  **import java.util.Comparator;**  **import java.util.Scanner;** |
| **Penjelasan** |
| * **import java.util.ArrayList:** digunakan untuk mengimpor ArrayList yang digunakan untuk menyimpan data objek. * **import java.util.Collections:** digunakan untuk mengimpor Collection yang digunakan untuk melakukan operasi pengurutan, pencarian. * **import java.util.Comparator:** digunakan untuk mengimpor Comparator yang digunakan untuk sorting. Implementasinya pada program digunakan untuk mengurutkan buku berdasarkan tahun terbit. * **import java.util.Scanner:** digunakan untuk mengimpor Scanner yang digunakan untuk jika user ingin memasukkan data atau pilihan. |

* 1. Class Buku

| Source Code |
| --- |
| **class** Buku {  *int* nomor;  *int* tahun;  *String* judul;  *String* penulis;  *boolean* status;  **public** Buku(*int* *nomor*, *String* *judul*, *String* *penulis*, *int* *tahun*, *boolean* *status*) {  this.nomor **=** nomor;  this.judul **=** judul;  this.penulis **=** penulis;  this.tahun **=** tahun;  this.status **=** status;  }  **public** **static** *Comparator*<*Buku*> UrutkanBerdasarkanTahun **=** **new** *Comparator*<*Buku*>() {  **public** *int* compare(*Buku* *buku1*, *Buku* *buku2*) {  **return** buku1.tahun **-** buku2.tahun;  }  };  } |
| Penjelasan |
| Class ini digunakan untuk mengimplementasikan comparator berdasarkan tahun buku tersebut diterbitkan, jika **buku1.tahun** kurang dari **buku2.tahun** maka akan bernilai negatif. jika **buku1.tahun** sama dengan **buku2.tahun** maka nilainya akan 0 dan jika **buku1.tahun** lebih besar dari **buku2.tahun** maka nilai akan menjadi positif. Comparator ini kemudian dapat mengurutkan tahun buku secara *ascending* atau dari tahun lama ke tahun yang paling baru. |

* 1. Class rakBuku

| Source Code |
| --- |
| **public** **class** rakBuku {  **private** **static** **final** *ArrayList*<*Buku*> rakBuku **=** **new** *ArrayList*<>();  **private** **static** **final** *Scanner* scanner **=** **new** Scanner(System.in);  **public** **static** *void* main(*String*[] *args*) {  System.out.println();  muatDataDariPenyimpanan();  *int* pilihan **=** 0;  **while** (true) {  System.out.println("-------------------------------");  System.out.println(" === Aplikasi Rak Buku ===");  System.out.println("-------------------------------");  System.out.println("1. Tampilkan Buku");  System.out.println("2. Lihat Informasi Buku");  System.out.println("3. Tambah Buku Baru");  System.out.println("4. Perbarui Informasi Buku");  System.out.println("5. Hapus Buku");  System.out.println("6. Keluar");  System.out.println("-------------------------------");  System.out.print("Pilih opsi: ");  **if** (scanner.hasNextInt()) { // ga masuk  pilihan **=** scanner.nextInt();  } **else** { // ga masuk  scanner.nextLine(); // ga masuk  System.out.println("\n\n======Input harus berupa angka!!======\n"); // ga masuk  **continue**; // ga masuk  }  **switch** (pilihan) {  **case** 1**:**  tampilkanBuku();  **break**;  **case** 2**:**  lihatInformasiBuku();  **break**;  **case** 3**:**  tambahBuku();  **break**;  **case** 4**:**  perbaruiBuku();  **break**;  **case** 5**:**  hapusBuku();  **break**;  **case** 6**:**  System.exit(0);  **break**;  **default:**  System.out.println("Pilihan tidak valid. Silakan coba lagi.");  }  }  } |
| Penjelasan |
| Class ini digunakan untuk menampilkan pilihan manajemen buku yang akan menjalankan fungsi tertentu. Terdapat opsi untuk melihat rak buku, lihat informasi buku, tambah buku, perbarui buku, hapus buku, dan keluar program. Kemudian jika user menginputkan pilihan yang tidak tersedia maka otomatis akan menampilkan teks pilihan tidak valid “Silakan coba lagi” dan mengarahkan user untuk input pilihan baru. |

* 1. Fungsi tampilkanBuku()

| Source Code |
| --- |
| **private** **static** *void* tampilkanBuku() {  Collections.sort(rakBuku, Buku.UrutkanBerdasarkanTahun);  System.out.println("\n === Daftar Buku ===\n-------------------------------");  **for** (*Buku* buku **:** rakBuku) {  System.out.println(buku.nomor **+** " " **+** buku.judul **+** " (" **+** buku.tahun **+** ")");  }  System.out.println();  } |
| Penjelasan |
| Fungsi ini digunakan untuk menampilkan daftar buku yang sudah diurutkan berdasarkan tahun terbit. Jika daftar buku tersebut sudah disiapkan dan diisi sebelumnya dalam koleksi **rakBuku**, outputnya akan menampilkan informasi buku dengan format yang sudah diberikan. |

* 1. Fungsi lihatInformasiBuku()

| Source Code |
| --- |
| **private** **static** *void* lihatInformasiBuku() {  System.out.println("\n === L ihat Buku ===\n-------------------------------");  System.out.print("Masukkan nomor buku: ");  *int* nomorBuku **=** scanner.nextInt();  *Buku* bukuDitemukan **=** temukanBuku(nomorBuku);  **if** (bukuDitemukan **==** null) {  System.out.println("Buku tidak ditemukan.\n");  **return**;  }  System.out.println("\n === Informasi Buku ===\n-------------------------------");  System.out.println("nomor: " **+** bukuDitemukan.nomor);  System.out.println("Judul: " **+** bukuDitemukan.judul);  System.out.println("Penulis: " **+** bukuDitemukan.penulis);  System.out.println("Tahun Terbit: " **+** bukuDitemukan.tahun);  System.out.println("Sudah Dibaca: " **+** bukuDitemukan.status **+** "\n");  } |
| Penjelasan |
| Fungsi ini digunakan untuk menerima input nomor dari pengguna dan mencari buku dengan nomor tersebut dan menampilkan informasi detailnya. Fungsi memeriksa apakah buku dengan nomor yang dimasukkan ditemukan. Jika tidak ditemukan, cetak pesan dan keluar dari fungsi. Jika buku ditemukan, cetak informasi rinci tentang buku, termasuk nomor, judul, penulis, tahun terbit, dan status (sudah dibaca atau belum). |

* 1. Fungsi tambahBuku()

| Source Code |
| --- |
| **private** **static** *void* tambahBuku() {  System.out.println("\n === Tambah Buku ===\n-------------------------------");  System.out.print("Masukkan judul buku: ");  *String* judul **=** scanner.nextLine();  System.out.print("Masukkan nama penulis: ");  *String* penulis **=** scanner.nextLine();  System.out.print("Masukkan tahun terbit: ");  *int* tahun **=** scanner.nextInt();  System.out.print("Apakah buku sudah dibaca? (true/false): ");  *boolean* status **=** scanner.nextBoolean();  *int* nomorBaru **=** rakBuku.isEmpty() **?** 1 **:** rakBuku.get(rakBuku.size() **-** 1).nomor **+** 1;  *Buku* bukuBaru **=** **new** Buku(nomorBaru, judul, penulis, tahun, status);  rakBuku.add(bukuBaru);  System.out.println("Buku berhasil ditambahkan.");  tampilkanBuku();  } |
| Penjelasan |
| Fungsi ini digunakan untuk menerima input judul, penulis, tahun terbit, dan status dari pengguna, kemudian membuat objek Buku baru dan menambahkannya ke **rakBuku**, dan menampilkan daftar buku setelah penambahan.  Terdapat fungsi untuk generate nomor baru, yaitu menghasilkan nomor baru untuk buku yang akan ditambahkan. Jika **rakBuku** kosong, maka nomor baru diatur ke 1. Jika tidak kosong, diambil nomor buku terakhir dalam koleksi dan ditambahkan 1. Sehingga objek buku baru bisa ditambahkan pada koleksi. |

* 1. Fungsi perbaruiBuku()

| Source Code |
| --- |
| **private** **static** *void* perbaruiBuku() {  System.out.println("\n === Perbarui Buku ===\n-------------------------------");  System.out.print("Masukkan nomor buku: ");  *int* nomorBuku **=** scanner.nextInt();  *Buku* bukuDitemukan **=** temukanBuku(nomorBuku);  **if** (bukuDitemukan **==** null) {  System.out.println("Buku tidak ditemukan.\n");  **return**;  }  System.out.println("\nDetail Saat Ini:\n-------------------------------");  System.out.println("Judul : " **+** bukuDitemukan.judul);  System.out.println("Penulis : " **+** bukuDitemukan.penulis);  System.out.println("Tahun Terbit: " **+** bukuDitemukan.tahun);  System.out.println("Sudah Dibaca: " **+** bukuDitemukan.status);  System.out.println("-------------------------------");  System.out.println("\nDetail Baru:\n-------------------------------");  scanner.nextLine();  System.out.print("Masukkan judul baru (tekan Enter untuk tetap sama): ");  *String* judulBaru **=** scanner.nextLine();  **if** (**!**judulBaru.isEmpty()) {  bukuDitemukan.judul **=** judulBaru;  }  System.out.print("Masukkan penulis baru (tekan Enter untuk tetap sama): ");  *String* penulisBaru **=** scanner.nextLine();  **if** (**!**penulisBaru.isEmpty()) {  bukuDitemukan.penulis **=** penulisBaru;  }  System.out.print("Masukkan tahun terbit baru (tekan 0 untuk tetap sama): ");  *int* tahunBaru **=** scanner.nextInt();  **if** (tahunBaru **!=** 0) {  bukuDitemukan.tahun **=** tahunBaru;  }  System.out.print("Apakah buku sudah dibaca? (true/false, tekan Enter untuk tetap sama): ");  scanner.nextLine();  *String* statusInput **=** scanner.nextLine();  **if** (**!**statusInput.isEmpty()) {  bukuDitemukan.status **=** Boolean.parseBoolean(statusInput);  }  System.out.println("-------------------------------\nBuku berhasil diperbarui.");  tampilkanBuku();  } |
| Penjelasan |
| Fungsi ini digunakan untuk menerima input nomor dari pengguna, mencari buku dengan nomor tersebut, menampilkan detail buku saat ini dan meminta input untuk pembaruan, kemudian memperbarui atribut buku yang diinginkan dan menampilkan daftar buku setelah pembaruan. |

* 1. Fungsi hapusBuku()

| Source Code |
| --- |
| **private** **static** *void* hapusBuku() {  System.out.println("\n === Hapus Buku ===\n-------------------------------");  System.out.print("Masukkan nomor buku yang ingin dihapus: ");  *int* nomorBuku **=** scanner.nextInt();  *int* indeksBuku **=** temukanIndeksBuku(nomorBuku);  **if** (indeksBuku **==** **-**1) {  System.out.println("Buku tidak ditemukan.\n");  **return**;  }  rakBuku.remove(indeksBuku);  System.out.println("Buku berhasil dihapus.");  tampilkanBuku();  } |
| Penjelasan |
| Fungsi ini digunakan untuk menerima input nomor dari pengguna, mencari indeks buku dengan nomor tersebut, menghapus buku dari **rakBuku** berdasarkan indeks yang diberikan dan menampilkan daftar buku setelah penghapusan. |

* 1. Fungsi muatDataDariPenyimpanan()

| Source Code |
| --- |
| **private** **static** *void* muatDataDariPenyimpanan() {  rakBuku.add(**new** Buku(978602647, "Cinta paling Rumit", "Boy Chandra", 2018, false));  rakBuku.add(**new** Buku(978979122, "Perahu Kertas", "Dewi Lestari Dee", 2010, true));  rakBuku.add(**new** Buku(978602947, "Sepotong Hati Yang Baru", "Tere Liye", 2012, false));  } |
| Penjelasan |
| Fungsi di atas digunakan untuk memuat 3 data buku dummy ke dalam koleksi **rakBuku**. Data ini dimasukkan secara manual, dan akan muncul sebagai buku-buku yang sudah ada pada rak buku. |

* 1. Sub Fungsi temukanBuku()

| Source Code |
| --- |
| **private** **static** *Buku* temukanBuku(*int* *nomorBuku*) {  **for** (*Buku* buku **:** rakBuku) {  **if** (buku.nomor **==** nomorBuku) {  **return** buku;  }  }  **return** null;  } |
| Penjelasan |
| Sub fungsi ini digunakan untuk mencari buku berdasarkan nomor dalam **rakBuku**. Pengulangan ini melakukan iterasi melalui setiap objek buku dalam koleksi **rakBuku** menggunakan **for-each** loop.   * Memeriksa apakah nomor dari buku saat ini (buku.nomor) sama dengan nomor yang dicari (nomorBuku). * Jika ditemukan buku dengan nomor yang sesuai, maka buku tersebut dikembalikan (return), menghentikan iterasi. * Jika iterasi selesai tanpa menemukan buku dengan nomor yang sesuai, maka fungsi mengembalikan null untuk menandakan bahwa buku tidak ditemukan. |

* 1. Sub Fungsi temukanIndeksBuku()

| Source Code |
| --- |
| **private** **static** *int* temukanIndeksBuku(*int* *nomorBuku*) {  **for** (*int* i **=** 0; i **<** rakBuku.size(); i**++**) {  **if** (rakBuku.get(i).nomor **==** nomorBuku) {  **return** i;  }  }  **return** **-**1;  } |
| Penjelasan |
| Sub fungsi ini digunakan untuk mencari indeks buku berdasarkan nomor dalam **rakBuku**. Perulangan melakukan iterasi melalui setiap indeks dalam koleksi **rakBuku** menggunakan loop for dengan variabel indeks **i**.   * Pada setiap iterasi, kode ini memeriksa apakah nomor dari buku pada indeks saat ini (rakBuku.get(i).nomor) sama dengan nomor yang dicari (nomorBuku). * Jika ditemukan buku dengan nomor yang sesuai, maka fungsi mengembalikan indeks buku tersebut (return i;), menghentikan iterasi. * Jika iterasi selesai tanpa menemukan buku dengan nomor yang sesuai, maka fungsi mengembalikan -1 untuk menandakan bahwa buku tidak ditemukan. |

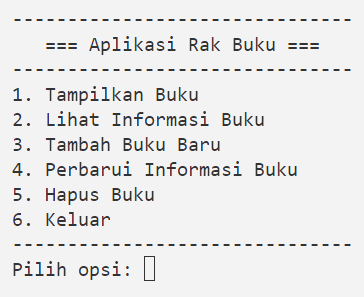
* **Sintak Program keseluruhan**

| import java.util.ArrayList;  import java.util.Collections;  import java.util.Comparator;  import java.util.Scanner;  class Buku {  int nomor;  int tahun;  String judul;  String penulis;  boolean status;  public Buku(int nomor, String judul, String penulis, int tahun, boolean status) {  this.nomor = nomor;  this.judul = judul;  this.penulis = penulis;  this.tahun = tahun;  this.status = status;  }  public static Comparator<Buku> UrutkanBerdasarkanTahun = new Comparator<Buku>() {  public int compare(Buku buku1, Buku buku2) {  return buku1.tahun - buku2.tahun;  }  };  }  public class rakBuku {  private static final ArrayList<Buku> rakBuku = new ArrayList<>();  private static final Scanner scanner = new Scanner(System.in);  public static void main(String[] args) {  System.out.println();  muatDataDariPenyimpanan();  int pilihan = 0;  while (true) {  System.out.println("-------------------------------");  System.out.println(" === Aplikasi Rak Buku ===");  System.out.println("-------------------------------");  System.out.println("1. Tampilkan Buku");  System.out.println("2. Lihat Informasi Buku");  System.out.println("3. Tambah Buku Baru");  System.out.println("4. Perbarui Informasi Buku");  System.out.println("5. Hapus Buku");  System.out.println("6. Keluar");  System.out.println("-------------------------------");  System.out.print("Pilih opsi: ");  if (scanner.hasNextInt()) { // ga masuk  pilihan = scanner.nextInt();  } else { // ga masuk  scanner.nextLine(); // ga masuk  System.out.println("\n\n======Input harus berupa angka!!======\n"); // ga masuk  continue; // ga masuk  }  switch (pilihan) {  case 1:  tampilkanBuku();  break;  case 2:  lihatInformasiBuku();  break;  case 3:  tambahBuku();  break;  case 4:  perbaruiBuku();  break;  case 5:  hapusBuku();  break;  case 6:  System.exit(0);  break;  default:  System.out.println("Pilihan tidak valid. Silakan coba lagi.");  }  }  }  private static void tampilkanBuku() {  Collections.sort(rakBuku, Buku.UrutkanBerdasarkanTahun);  System.out.println("\n === Daftar Buku ===\n-------------------------------");  for (Buku buku : rakBuku) {  System.out.println(buku.nomor + " " + buku.judul + " (" + buku.tahun + ")");  }  System.out.println();  }  private static void lihatInformasiBuku() {  System.out.println("\n === L ihat Buku ===\n-------------------------------");  System.out.print("Masukkan nomor buku: ");  int nomorBuku = scanner.nextInt();  Buku bukuDitemukan = temukanBuku(nomorBuku);  if (bukuDitemukan == null) {  System.out.println("Buku tidak ditemukan.\n");  return;  }  System.out.println("\n === Informasi Buku ===\n-------------------------------");  System.out.println("nomor: " + bukuDitemukan.nomor);  System.out.println("Judul: " + bukuDitemukan.judul);  System.out.println("Penulis: " + bukuDitemukan.penulis);  System.out.println("Tahun Terbit: " + bukuDitemukan.tahun);  System.out.println("Sudah Dibaca: " + bukuDitemukan.status + "\n");  }  private static void tambahBuku() {  System.out.println("\n === Tambah Buku ===\n-------------------------------");  System.out.print("Masukkan judul buku: ");  String judul = scanner.nextLine();  System.out.print("Masukkan nama penulis: ");  String penulis = scanner.nextLine();  System.out.print("Masukkan tahun terbit: ");  int tahun = scanner.nextInt();  System.out.print("Apakah buku sudah dibaca? (true/false): ");  boolean status = scanner.nextBoolean();  int nomorBaru = rakBuku.isEmpty() ? 1 : rakBuku.get(rakBuku.size() - 1).nomor + 1;  Buku bukuBaru = new Buku(nomorBaru, judul, penulis, tahun, status);  rakBuku.add(bukuBaru);  System.out.println("Buku berhasil ditambahkan.");  tampilkanBuku();  }  private static void perbaruiBuku() {  System.out.println("\n === Perbarui Buku ===\n-------------------------------");  System.out.print("Masukkan nomor buku: ");  int nomorBuku = scanner.nextInt();  Buku bukuDitemukan = temukanBuku(nomorBuku);  if (bukuDitemukan == null) {  System.out.println("Buku tidak ditemukan.\n");  return;  }  System.out.println("\nDetail Saat Ini:\n-------------------------------");  System.out.println("Judul : " + bukuDitemukan.judul);  System.out.println("Penulis : " + bukuDitemukan.penulis);  System.out.println("Tahun Terbit: " + bukuDitemukan.tahun);  System.out.println("Sudah Dibaca: " + bukuDitemukan.status);  System.out.println("-------------------------------");  System.out.println("\nDetail Baru:\n-------------------------------");  scanner.nextLine();  System.out.print("Masukkan judul baru (tekan Enter untuk tetap sama): ");  String judulBaru = scanner.nextLine();  if (!judulBaru.isEmpty()) {  bukuDitemukan.judul = judulBaru;  }  System.out.print("Masukkan penulis baru (tekan Enter untuk tetap sama): ");  String penulisBaru = scanner.nextLine();  if (!penulisBaru.isEmpty()) {  bukuDitemukan.penulis = penulisBaru;  }  System.out.print("Masukkan tahun terbit baru (tekan 0 untuk tetap sama): ");  int tahunBaru = scanner.nextInt();  if (tahunBaru != 0) {  bukuDitemukan.tahun = tahunBaru;  }  System.out.print("Apakah buku sudah dibaca? (true/false, tekan Enter untuk tetap sama): ");  scanner.nextLine();  String statusInput = scanner.nextLine();  if (!statusInput.isEmpty()) {  bukuDitemukan.status = Boolean.parseBoolean(statusInput);  }  System.out.println("-------------------------------\nBuku berhasil diperbarui.");  tampilkanBuku();  }  private static void hapusBuku() {  System.out.println("\n === Hapus Buku ===\n-------------------------------");  System.out.print("Masukkan nomor buku yang ingin dihapus: ");  int nomorBuku = scanner.nextInt();  int indeksBuku = temukanIndeksBuku(nomorBuku);  if (indeksBuku == -1) {  System.out.println("Buku tidak ditemukan.\n");  return;  }  rakBuku.remove(indeksBuku);  System.out.println("Buku berhasil dihapus.");  tampilkanBuku();  }  private static void muatDataDariPenyimpanan() {  rakBuku.add(new Buku(1, "Cinta paling Rumit", "Boy Chandra", 2018, false));  rakBuku.add(new Buku(2, "Perahu Kertas", "Dewi Lestari Dee", 2010, true));  rakBuku.add(new Buku(3, "Sepotong Hati Yang Baru", "Tere Liye", 2012, false));  }  private static Buku temukanBuku(int nomorBuku) {  for (Buku buku : rakBuku) {  if (buku.nomor == nomorBuku) {  return buku;  }  }  return null;  }  private static int temukanIndeksBuku(int nomorBuku) {  for (int i = 0; i < rakBuku.size(); i++) {  if (rakBuku.get(i).nomor == nomorBuku) {  return i;  }  }  return -1;  }  } |
| --- |

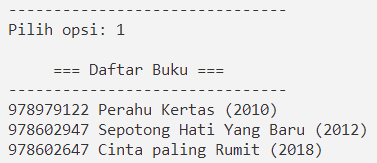
# 

# HASIL PROGRAM

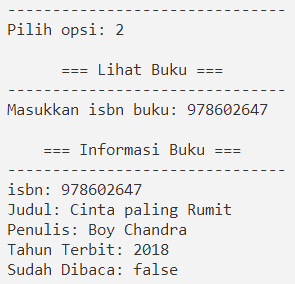
* 1. Tampilan pilihan dan opsi aplikasi rak buku



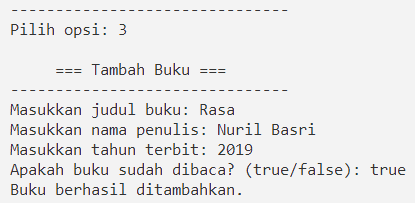
* 1. Opsi 1: Tampilkan Buku

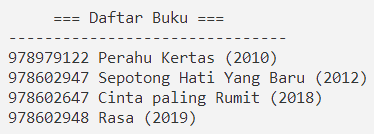


* 1. Opsi 2: Lihat Informasi Buku

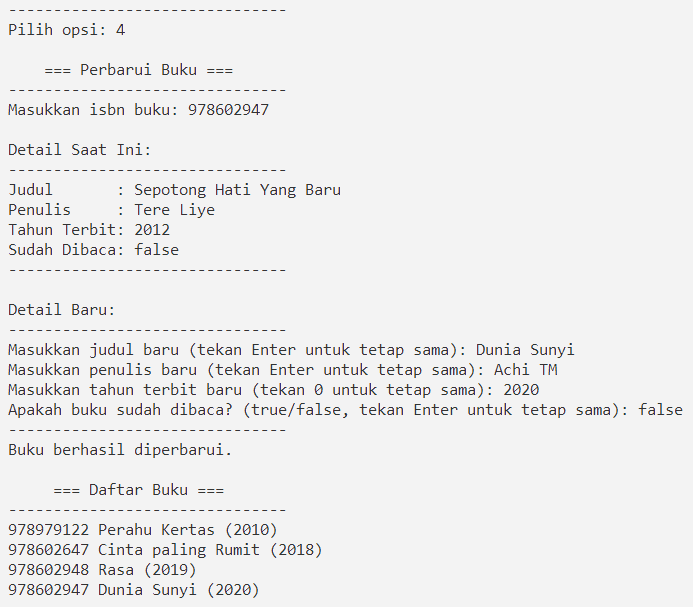


* 1. Opsi 3: Tambah Buku Baru

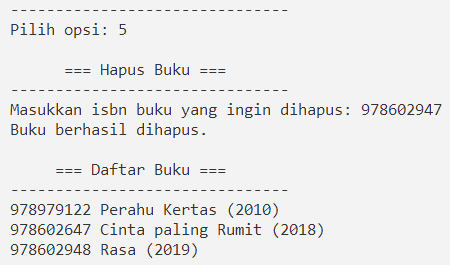




* 1. Opsi 4: Perbarui Informasi Buku



* 1. Opsi 5: Hapus Buku



* 1. Opsi 6: Keluar



# JOBDESK ANGGOTA

| **NO** | **NAMA** | **JOB DESCRIPTION** |
| --- | --- | --- |
| 1 | M. Zafif Hilmi Al-hadi | * Ketua kelompok * Membuat Judul * Membuat flowchart utama * Membuat flowchart fungsi tampilkanBuku() * Membuat kerangka laporan * Membuat deskripsi aplikasi yang dibuat |
| 2 | Miftakhul Huda Dwi Setiawan | * Membuat flowchart fungsi lihatInformasiBuku() * Membuat flowchart fungsi hapusBuku() * Membuat flowchart sub fungsi temukanBuku() * Membuat flowchart sub fungsi temukanIndeksBuku() |
| 3 | Nastiti Chusna Haliza | * Membuat konversi sintak program java * Membuat flowchart fungsi tambahBuku() |
| 4 | Anggata Sakti B. H | * Membuat flowchart fungsi perbaruiBuku() * Membantu meyusun laporan |

# KESIMPULAN

Aplikasi Rak Buku merupakan program sederhana untuk memberikan fungsionalitas dasar dalam pengelolaan dan pemeliharaan koleksi buku. Aplikasi ini menggunakan struktur data ArrayList untuk menyimpan objek-objek buku dan menyediakan tampilan sederhana berbasis teks pada konsol untuk pengguna berinteraksi dengan rak buku.

Pengguna dapat melakukan berbagai tindakan terkait buku melalui menu interaktif, termasuk menampilkan daftar buku, melihat informasi spesifik tentang suatu buku, menambahkan buku baru, memperbarui informasi buku yang sudah ada, dan menghapus buku dari koleksi. Setiap buku direpresentasikan oleh kelas Buku, yang memiliki atribut seperti nomor, judul, penulis, tahun terbit, dan status (sudah dibaca atau belum).

Aplikasi ini juga memberikan opsi pengurutan daftar buku berdasarkan tahun terbit menggunakan konsep Comparator. Selain itu, terdapat fungsi untuk memuat data awal buku ke dalam koleksi, memberikan beberapa contoh buku sebagai data dummy.

Keseluruhan, Aplikasi Rak Buku dapat menjadi alat yang berguna bagi pengguna yang ingin mengorganisir dan melacak informasi buku secara terstruktur.