**LAPORAN PRAKTIKUM VIII**

**ALGORITMA STRUKTUR DATA**

**“Map & Hash Map”**



Nama : Muhammad Azka Raki

NIM : 2311016110005

**KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI**

**UNIVERSITAS LAMBUNG MANGKURAT**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**PROGRAM STUDI S1 ILMU KOMPUTER**

**BANJARBARU**

**2024**

1. **Tinjauan Pustaka**

Map memungkinkan untuk mengaitkan elemen dari kumpulan kunci dengan elemen dari kumpulan nilai. Map dapat digunakan saat ingin mencari objek dengan menggunakan kunci. Kedua Class HashMap dan TreeMap mengimplementasikan *interface* Map.

HashMap adalah *in-built collection* atau koleksi bawaan Java yang berdasarkan implementasi Hash table. Implementasi ini menghasilkan kinerja waktu konstan untuk operasi *get* (mengambil nilai) dan operasi *put* (menyimpan nilai). HashMap adalah struktur data yang umum dan efisien yang digunakan menyimpan dan mengakses data yang dapat diorganisir sebagai pasangan (K, C), dimana K adalah kunci unik dengan konten terkait C. HashMap mengizinkan satu kunci *null* dan apa saja sejumlah nilai *null*. HashMap tidak mengizinkan kunci duplikat dan setiap kunci dapat diasosiasikan atau dipetakan ke hanya satu nilai.

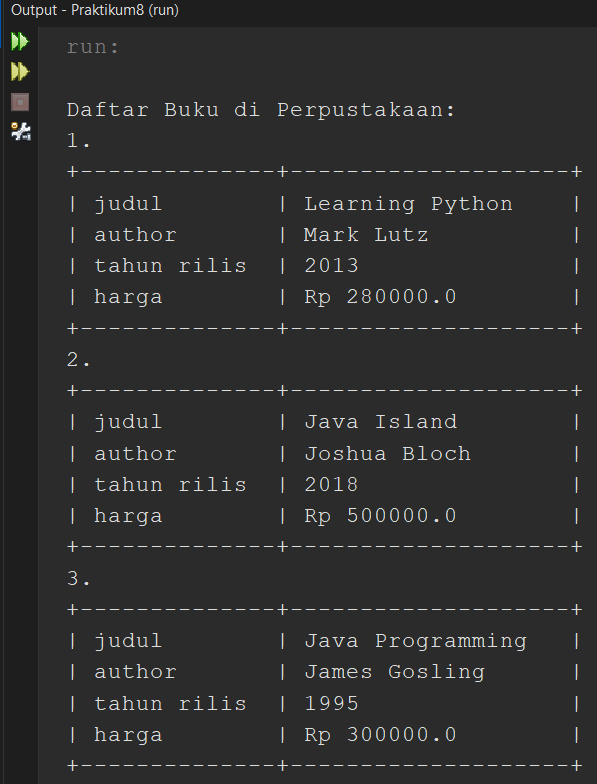
1. **Source Code**
2. Perpustakaan.java

|  |
| --- |
| package praktikum8;  import java.util.HashMap;  public class Perpustakaan {    // Studi Kasus: Sistem Pencatatan Buku Perpustakaan  // Key: ISBN  // Value: Buku  public static void main(String[] args) {  HashMap<String, Buku> perpus = new HashMap<>();  perpus.put("978-3-16-148410-0", new Buku("Java Programming", "James Gosling", 1995, 300000));  perpus.put("978-0-596-52068-7", new Buku("Learning Python", "Mark Lutz", 2013, 280000));  perpus.put("978-1-491-99455-3", new Buku("Java Island", "Joshua Bloch", 2018, 500000));  printBukuDiPerpustakaan(perpus);    String isbn = "978-3-16-148410-0";  if (perpus.containsKey(isbn)) {  System.out.println("Data Buku dengan isbn " + isbn + " :");  perpus.get(isbn).printBuku();  } else {  System.out.println("Buku dengan ISBN " + isbn + " tidak ditemukan.");  }  System.out.println("");    if (perpus.containsKey("978-0-596-52068-7")) {  Buku buku = perpus.get("978-0-596-52068-7");  buku.setHarga(400000);  System.out.println("Data buku setelah update harga:");  perpus.get("978-0-596-52068-7").printBuku();  }  System.out.println("");    perpus.remove("978-1-491-99455-3");  System.out.println("Buku dengan ISBN 978-1-491-99455-3 telah dihapus.");    printBukuDiPerpustakaan(perpus);  }  static void printBukuDiPerpustakaan(HashMap<String, Buku> perpus) {  System.out.println("\nDaftar Buku di Perpustakaan:");  int i = 1;  for (String key : perpus.keySet()) {  System.out.println(i + ".");  perpus.get(key).printBuku();  i++;  }  System.out.println("");  }    } |

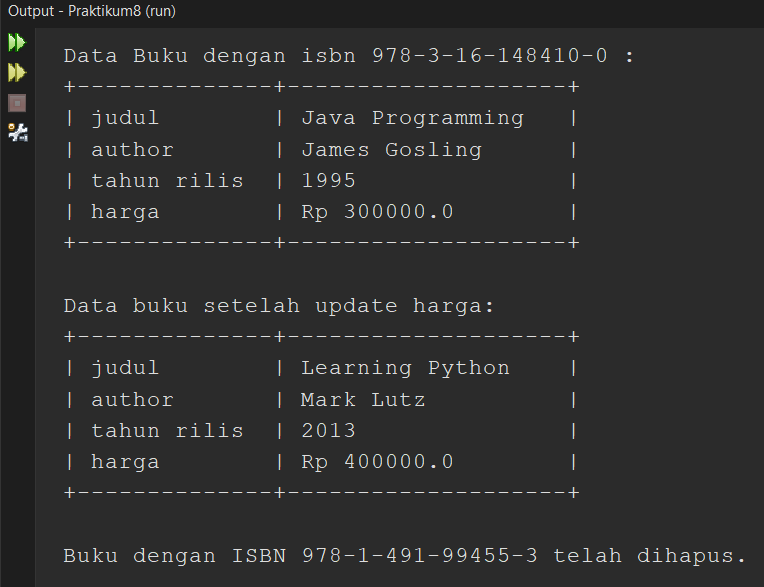
1. Buku.java

|  |
| --- |
| package praktikum8;  public class Buku {  private String judul;  private String author;  private int tahunRilis;  private double harga;  public Buku(String judul, String author, int tahunRilis, double harga) {  this.judul = judul;  this.author = author;  this.tahunRilis = tahunRilis;  this.harga = harga;  }  public String getJudul() {  return judul;  }  public void setJudul(String judul) {  this.judul = judul;  }  public String getAuthor() {  return author;  }  public void setAuthor(String author) {  this.author = author;  }  public int getTahunRilis() {  return tahunRilis;  }  public void setTahunRilis(int tahunRilis) {  this.tahunRilis = tahunRilis;  }  public double getHarga() {  return harga;  }  public void setHarga(double harga) {  this.harga = harga;  }    public void printBuku() {  System.out.println("+--------------+--------------------+");  System.out.printf("| %-12s | %-18s |\n", "judul", judul);  System.out.printf("| %-12s | %-18s |\n", "author", author);  System.out.printf("| %-12s | %-18s |\n", "tahun rilis", tahunRilis);  System.out.printf("| %-12s | Rp %-15s |\n", "harga", harga);  System.out.println("+--------------+--------------------+");  }  } |

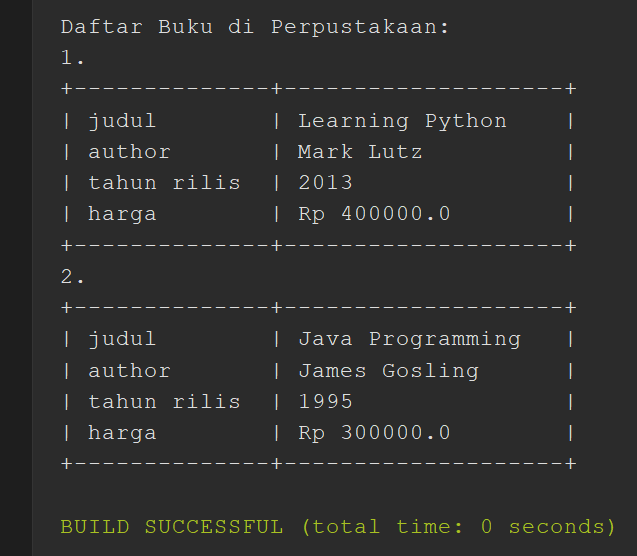
1. **Output**



Gambar 3.1 hasil output



Gambar 3.2 hasil output (lanjutan gambar 3.1)



Gambar 3.3 hasil output (lanjutan gambar 3.2)

**Sumber :**

Areias, M., & Rocha, R. (2021). On the correctness and efficiency of a novel lock-free hash trie map design. Journal of Parallel and Distributed Computing, 150, 184–195. <https://doi.org/10.1016/j.jpdc.2021.01.001>

Bajracharya, D. (2020). A REVIEW ON JAVA HASHMAP AND TREEMAP. International Journal of Engineering Applied Sciences and Technology, 5(1), 134–138. <https://doi.org/10.33564/ijeast.2020.v05i01.018>

Horstmann, C. S. (2017). *Big Java: Late Objects*. John Wiley & Sons.