**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

**JOBSHEET PERTEMUAN KE-6**

****

**NAMA : ALVINO VALERIAN D.R**

**KELAS : 1A**

**NO. ABSEN : 05**

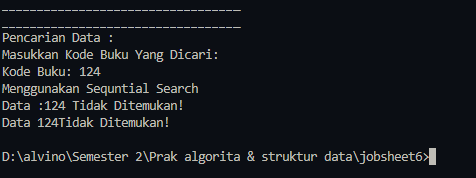
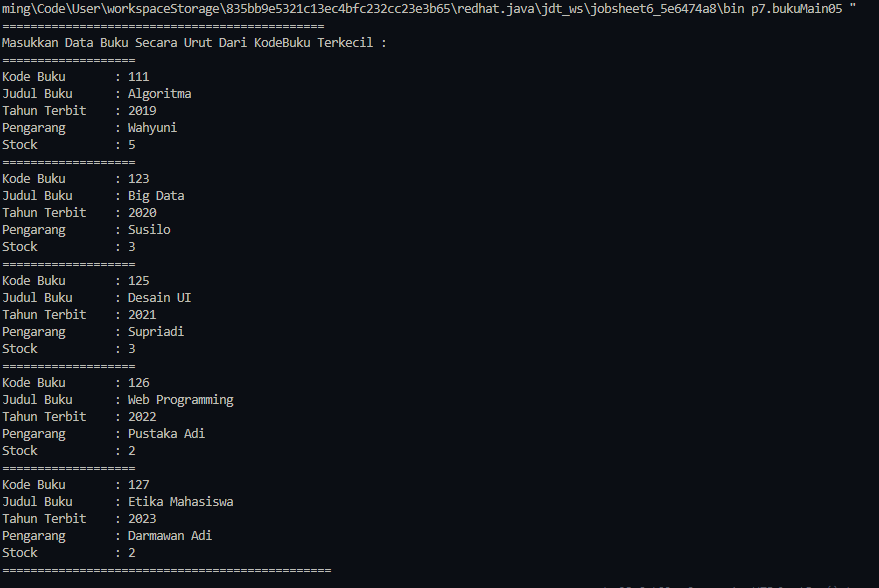
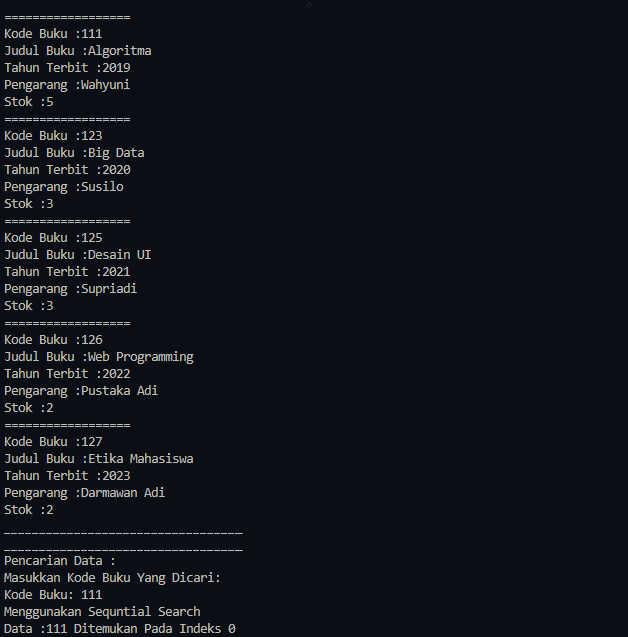
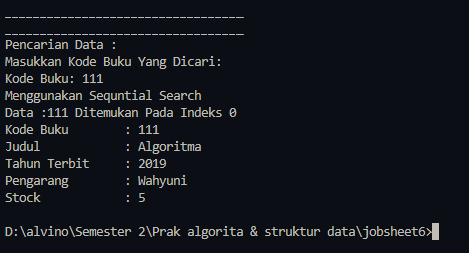
**NIM : 2341720027**

**PROGRAM STUDI TEKNIK INFORMATIKA**

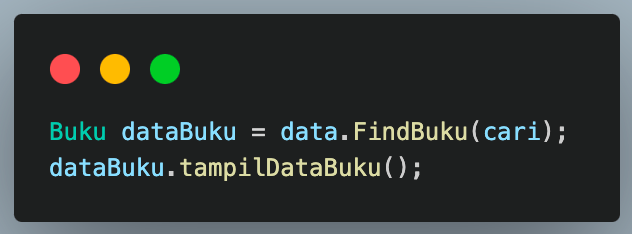
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**



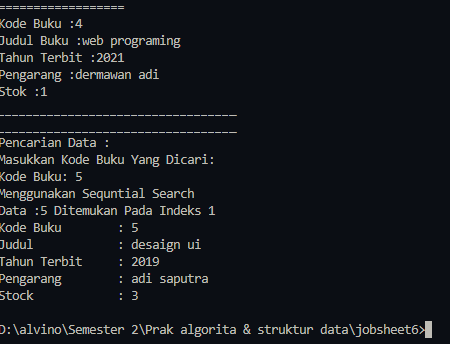
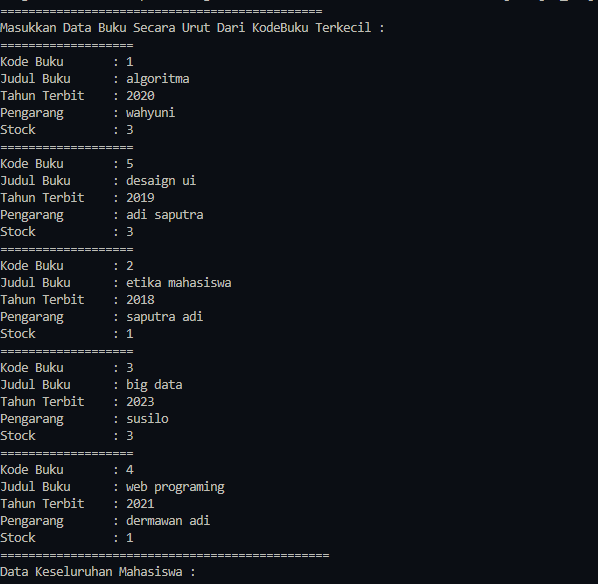
6.2.3. Pertanyaan1. Jelaskan fungsi break yang ada pada method FindSeqSearch!2. Jika Data Kode Buku yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih  
dapat berjalan? Apakah hasil yang dikeluarkan benar? Tunjukkan hasil screenshoot untuk bukti  
dengan kode Buku yang acak. Jelaskan Mengapa hal tersebut bisa terjadi?  
3. Buat method baru dengan nama FindBuku menggunakan konsep sequential search dengan tipe  
method dari FindBuku adalah BukuNoAbsen. Sehingga Anda bisa memanggil method  
tersebut pada class BukuMain seperti gambar berikut :



JAWABAN:

1. Fungsi break pada method findSeqSearch digunakan untuk keluar dari perulangan for ketika data yang dicari ditemukan.

2.



Hasil yang dikeluarkan program **tetap benar** dalam menemukan data buku yang dicari, meskipun urutan kode bukunya acak.

3.



public Buku05 FindBuku(int cari){

for(int j=0;j<listBk.length;j++){

if (listBk[j].kodeBuku==cari){

return listBk[j];

}

}

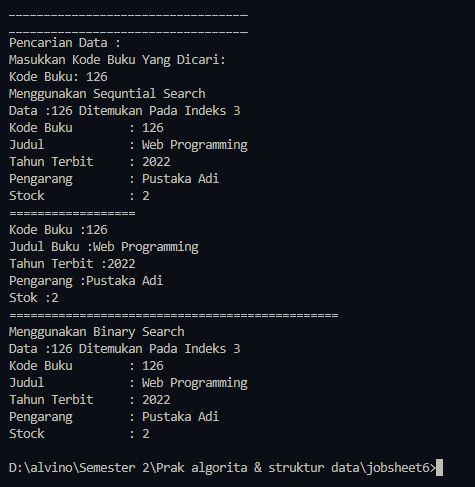
return null;

}

Buku05 dataBuku= data.FindBuku(cari);

dataBuku.tampilDataBuku();

6.3.2. Verifikasi Hasil Percobaan



6.3.3. Pertanyaan1. Tunjukkan pada kode program yang mana proses divide dijalankan!  
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!  
4. Jika data Kode Buku yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa  
demikian! Tunjukkan hasil screenshoot untuk bukti dengan kode Buku yang acak. Jelaskan  
Mengapa hal tersebut bisa terjadi?  
3. Jika Kode Buku yang dimasukkan dari Kode Buku terbesar ke terkecil (missal : 20215, 20214,  
20212, 20211, 20210) dan elemen yang dicari adalah 20210. Bagaimana hasil dari binary search?  
Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai!

JAWABAN:

1.

if (right >= left) {

mid = (right + left) / 2;

}

2.

if (cari == listBk[mid].kodeBuku) {

return mid;

} else if (listBk[mid].kodeBuku > cari) {

return FindBinarySearch(cari, left, mid - 1);

} else {

return FindBinarySearch(cari, mid + 1, right);

}

3

public int FindBinarySearch(int cari,int left,int right){

int mid;

if(right>=left){

mid = (right+left)/2;

if (cari == listBk[mid].kodeBuku){

return mid;

}else if(listBk[mid].kodeBuku < cari){

return FindBinarySearch(cari, left, mid-1);

}else {

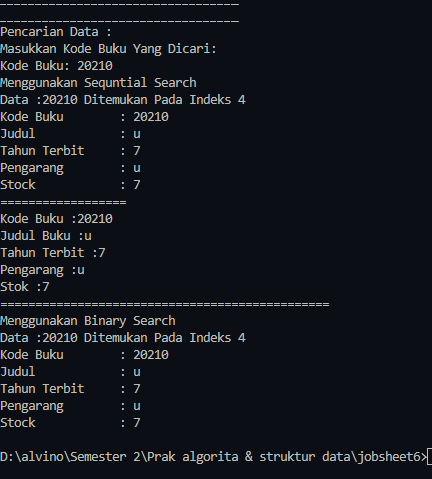
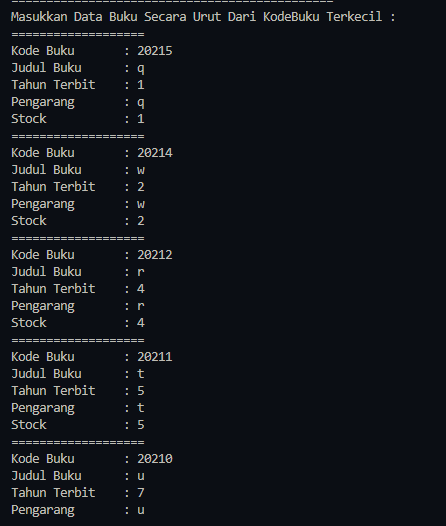
return FindBinarySearch(cari, mid + 1, right);

}

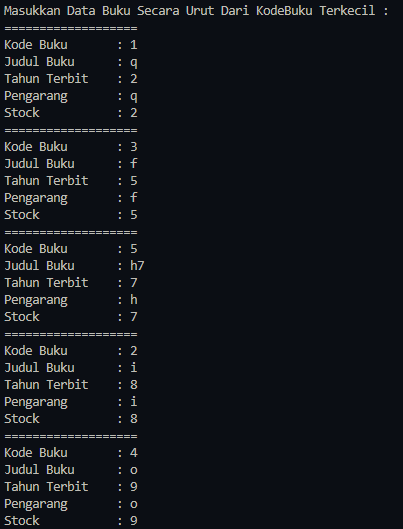
}

return -1;

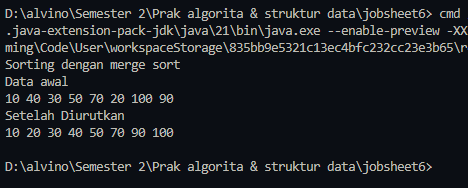
}



4. . program binary search masih dapat berjalan dengan benar meskipun data kode buku tidak urut. Algoritma binary search tidak bergantung pada urutan data dalam array.



**6.4.2. Verifikasi Hasil Percobaan**Cocokkan hasil compile kode program anda dengan gambar berikut ini



**6.5. Latihan Praktikum**1. Modifikasi percobaan searching diatas dengan ketentuan berikut ini  
- Ubah tipe data dari kode Buku yang awalnya int menjadi String  
- Tambahkan method untuk pencarian kode Buku (bertipe data String) dengan menggunakan  
sequential search dan binary search.  
2. Modifikasi percobaan searching diatas dengan ketentuan berikut ini

- Tambahkan method pencarian judul buku menggunakan sequential search dan binary  
search. Sebelum dilakukan searching dengan binary search data harus dilakukan pengurutan  
dengan menggunakan algoritma Sorting (bebas pilih algoritma sorting apapun)! Sehingga  
ketika input data acak, maka algoritma searching akan tetap berjalan  
- Buat aturan untuk mendeteksi hasil pencarian judul buku yang lebih dari 1 hasil dalam  
bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang  
diberikan!

JAWABAN:

BUKU05

package p7;

import java.util.Scanner;

public class bukuMain05 {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

Scanner s1 = new Scanner(System.in);

PencarianBuku05 data = new PencarianBuku05();

int jmlBuku = 5;

System.out.println("---------------------------------------");

System.out.println("Masukkan data Buku secara Urut dari Kode Buku Terkecil: ");

for (int i = 0; i < jmlBuku; i++) {

System.out.println("==========================");

System.out.print("Kode Buku \t: ");

String kodeBuku = s1.nextLine();

System.out.print("Judul Buku \t: ");

String judulBuku = s1.nextLine();

System.out.print("Tahun Terbit \t: ");

int tahunTerbit = s.nextInt();

System.out.print("Pengarang \t: ");

String pengarang = s1.nextLine();

System.out.print("Stok \t: ");

int stock = s.nextInt();

Buku05 m = new Buku05(kodeBuku, judulBuku, tahunTerbit, pengarang, stock);

data.tambah(m);

}

System.out.println("---------------------------------------");

System.out.println("Data keseluruhan Buku: ");

data.tampil();

System.out.print("Cari buku berdasarkan judul: ");

String cariJudul = s1.nextLine();

data.sort();

int posisi = data.findSeqSearchJudul(cariJudul);

System.out.println("sequential search: ");

data.tampilData(cariJudul, posisi);

System.out.println("binary search: ");

posisi = data.findJudulBinarySearch(cariJudul, 0, data.listBk.length - 1);

data.tampilData(cariJudul, posisi);

}

}

BUKUPENCARIAN

package p7;

public class PencarianBuku05 {

Buku05[] listBk = new Buku05[5];

int idx;

void tambah(Buku05 m) {

if (idx < listBk.length) {

listBk[idx] = m;

idx++;

} else {

System.out.println("Data sudah penuh");

}

}

void tampil() {

for (Buku05 m : listBk) {

m.tampilDataBuku();

}

}

int findSeqSearch(String cari) {

int posisi = -1;

for (int j = 0; j < listBk.length; j++) {

if (listBk[j].kodeBuku.equals(cari)) {

posisi = j;

break;

}

}

return posisi;

}

int findSeqSearchJudul(String cari) {

int count = 0;

int posisi = -1;

for (int j = 0; j < listBk.length; j++) {

if (listBk[j].judulBuku.toLowerCase().contains(cari.toLowerCase())) {

posisi = j;

count++;

}

}

if (count > 1) {

System.out.println(count + " data ditemukan dengan judul yang sama!");

}

return posisi;

}

void sort() {

int length = listBk.length;

for (int i = 0; i < length - 1; i++) {

for (int j = 0; j < length - i - 1; j++) {

if (listBk[j].judulBuku.compareToIgnoreCase(listBk[j + 1].judulBuku) > 0) {

Buku05 temp = listBk[j];

listBk[j] = listBk[j + 1];

listBk[j + 1] = temp;

}

}

}

}

int findJudulBinarySearch(String cari, int left, int right) {

if (right >= left) {

int mid = left + (right - left) / 2;

int comparison = listBk[mid].judulBuku.compareToIgnoreCase(cari);

if (comparison == 0) {

return mid;

} else if (comparison < 0) {

return findJudulBinarySearch(cari, mid + 1, right);

} else {

return findJudulBinarySearch(cari, left, mid - 1);

}

}

return -1;

}

public Buku05 findBuku(String cari) {

int posisi = findSeqSearch(cari);

return (posisi != -1) ? listBk[posisi] : null;

}

int findBinarySearch(String cari, int left, int right) {

if (right >= left) {

int mid = (left + right) / 2;

int comparison = cari.compareTo(listBk[mid].kodeBuku);

if (comparison == 0) {

return mid;

} else if (comparison < 0) {

return findBinarySearch(cari, left, mid - 1);

} else {

return findBinarySearch(cari, mid + 1, right);

}

}

return -1;

}

void tampilPosisi(String x, int pos) {

if (pos != -1) {

System.out.println("Data : " + x + " ditemukan pada indeks " + pos);

} else {

System.out.println("Data " + x + " tidak ditemukan");

}

}

void tampilData(String x, int pos) {

if (pos != -1) {

System.out.println("Kode Buku\t : " + x);

System.out.println("Judul\t : " + listBk[pos].judulBuku);

System.out.println("Tahun Terbit\t : " + listBk[pos].tahunTerbit);

System.out.println("Pengarang\t : " + listBk[pos].pengarang);

System.out.println("Stok\t : " + listBk[pos].stock);

} else {

System.out.println("Data " + x + " tidak ditemukan");

}

}

}

BUKUMAIN

package p7;

import java.util.Scanner;

public class bukuMain05 {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

Scanner s1 = new Scanner(System.in);

PencarianBuku05 data = new PencarianBuku05();

int jmlBuku = 5;

System.out.println("---------------------------------------");

System.out.println("Masukkan data Buku secara Urut dari Kode Buku Terkecil: ");

for (int i = 0; i < jmlBuku; i++) {

System.out.println("==========================");

System.out.print("Kode Buku \t: ");

String kodeBuku = s1.nextLine();

System.out.print("Judul Buku \t: ");

String judulBuku = s1.nextLine();

System.out.print("Tahun Terbit \t: ");

int tahunTerbit = s.nextInt();

System.out.print("Pengarang \t: ");

String pengarang = s1.nextLine();

System.out.print("Stok \t: ");

int stock = s.nextInt();

Buku05 m = new Buku05(kodeBuku, judulBuku, tahunTerbit, pengarang, stock);

data.tambah(m);

}

System.out.println("---------------------------------------");

System.out.println("Data keseluruhan Buku: ");

data.tampil();

System.out.print("Cari buku berdasarkan judul: ");

String cariJudul = s1.nextLine();

data.sort();

int posisi = data.findSeqSearchJudul(cariJudul);

System.out.println("sequential search: ");

data.tampilData(cariJudul, posisi);

System.out.println("binary search: ");

posisi = data.findJudulBinarySearch(cariJudul, 0, data.listBk.length - 1);

data.tampilData(cariJudul, posisi);

}

}