# Searching

## Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Memahami skema-skema *searching* (pencarian) data dengan benar.
2. Memahami skema Linear Search.
3. Memahami skema Binary Search.
4. Mengimplementasikan skema-skema tersebut terhadap kasus-kasus tertentu.

## Alat & Bahan

Alat & Bahan Yang digunakan adalah hardware perangkat PC beserta Kelengkapannya berjumlah 40 PC serta Software IntelliJ IDEA yang telah terinstall pada masing-masing PC

## Dasar Teori

### Searching

Pada suatu data seringkali dibutuhkan pembacaan kembali informasi (information retrieval) dengan cara searching. Searching adalah pencarian data dengan cara menelusuri data-data tersebut. Perhatikan contoh kode program berikut.

|  |
| --- |
| package praktikum12\_1;  import java.util.Scanner;  class OverviewSearch{  public static void main (String args[]){  Scanner sc = new Scanner(System.in);  int A[100], hasil;  System.out.println("Masukkan n:");  int n = sc.nextInt();  System.out.println("Jumlah data: " + n);  for(i = 0; i < n; i++){  System.out.printn("A[" + i + "]" + " =");  A[i] = sc.nextInt();  }  System.out.println("Masukkan angka yang dicari:");  int cari = sc.nextInt();    /\*\*Gunakan salah satu fungsi pencarian disini\*\*/    If (hasil >= 0)  System.out.println("Data ditemukan pada indeks" + hasil);  else  System.out.println("Data tidak ditemukan");  }  }  } |
| ………………………… |

Program ‎121 Overview Searching

Pada contoh kode di atas, program akan meminta sekumpulan set angka yang akan dimasukkan ke dalam array lalu mencarinya, setelah itu menampilkan hasil index angka itu berada dalam array. Jika hasil bernilai -1, artinya data tidak ditemukan. Ada dua metode yang dibahas pada modul ini, yaitu Sequential Search (Linear Search) dan Binary Search.

### Linear Search

Linear Search adalah suatu teknik pencarian data dalam array (1 dimensi) yang akan menelusuri semua elemen-elemen array dari awal sampai akhir. Kemungkinan terbaik (best case) adalah jika data yang dicari terletak di indeks array terdepan (elemen array pertama) sehingga waktu yang dibutuhkan untuk pencarian data sangat sebentar (minimal). Kemungkinan terburuk (worst case) adalah jika data yang dicari terletak di indeks array terakhir (elemen array terakhir) sehingga waktu yang dibutuhkan untuk pencarian data sangat lama (maksimal).

Pada linear search, data yang dicari (key) akan disamakan dengan elemen array, mulai dari elemen pertama. Jika key dengan elemen tersebut bernilai sama, algoritma akan mengembalikan indeks dimana nilai tersebut ditemukan. Namun bila tidak sama, pencarian akan diteruskan pada elemen berikutnya, sampai elemen terakhir array. Ketika sudah mencapai elemen terakhir dan tidak ditemukan nilaiyang sama, algoritma akan memberitahu bahwa nilai tersebut tidak ditemukan. Perhatikan Program 12-2 untuk lebih jelasnya.

|  |
| --- |
| int linearSearch(int key) {  for (int i = 0; i < A.length; i++) {  if(A[i] == key)  return key; //mengembalikan indeks  }  return -1; //data yang dicari tidak ditemukan  } |
| Apakah hasilnya jika A = [1,2,...,10] dan cari = 3?  ………………………… |

Program ‎122 Linear Search

Sumber: Deitel and Deitel, 2012

### Binary Search

Binary Search adalah teknik pencarian data dalam dengan cara membagi data menjadi dua bagian setiap kali terjadi proses pencarian. Data yang ada harus diurutkan terlebih dahulu berdasarkan suatu urutan tertentu yang dijadikan kunci pencarian.

Prinsip Binary Search adalah:

1. Data diambil dari posisi 1 sampai posisi akhir N
2. Kemudian cari posisi data tengah dengan rumus: (posisi awal + posisi akhir) / 2
3. Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih besar
4. Jika lebih besar, maka proses pencarian dicari dengan posisi awal adalah posisi tengah + 1
5. Jika lebih kecil, maka proses pencarian dicari dengan posisi akhir adalah posisi tengah – 1
6. Jika data sama, berarti ketemu.

Untuk lebih jelasnya, perhatikan Program 12-3.

|  |
| --- |
| int binarySearch(int key) {  int awal = 0;  int akhir = data.length -1;  int tengah = (awal + akhir + 1)/2;//tengah awal  int ketemu = -1// -1 jika data tidak ditemukan  do{  if(key == data[tengah])  ketemu = tengah;  else if(key < data[tengah])  akhir = tengah – 1;  else  awal = tengah + 1;    tengah = (awal + akhir + 1)/2;//perhitungan tengah kembali  }while((awal<=akhir) && (ketemu== -1))    return ketemu;//mengembalikan nilai indeks ketemu  } |

Program ‎123 Binary Search

Sumber: Deitel and Deitel, 2012

## Studi Kasus

1. Buatlah sebuah program untuk mengurutkan data-data mahasiswa (Nama, NIM, Nilai) berdasarkan nilai lalu menampilkan hasil urutannya.
2. Buatlah sebuah program untuk menghitung mean, median, dan modus dalam suatu set angka.
3. Binary search dapat dibuat dengan menggunakan method rekursif. Buatlah method binary search dengan menggunakan rekursif.