

Praktikum 11

Pencarian (Searching)

POKOK BAHASAN:

- ✓ Konsep pencarian dengan *sequential search* dan *binary search*
- ✓ Struktur data proses pencarian
- ✓ Implementasi algoritma pencarian *sequential search* dan *binary search* dalam Bahasa C

TUJUAN BELAJAR:

Setelah melakukan praktikum dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami konsep pencarian dengan metode *sequential search* dan *binary search*.
- ✓ Mengimplementasikan algoritma *sequential search* dan *binary search* dalam bentuk flowchart.
- ✓ Membuat diagram alir dan mengimplementasikan algoritma pada suatu permasalahan.

DASAR TEORI:

Algoritma pencarian (*searching algorithm*) adalah algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).

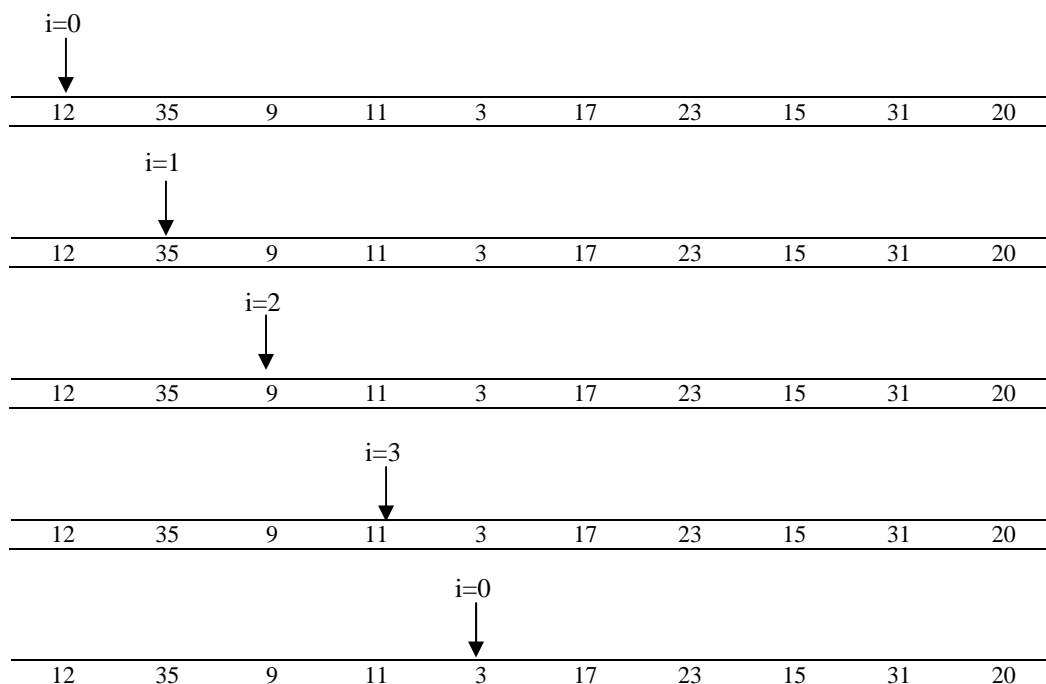
Ada dua macam teknik pencarian yaitu pencarian sekuensial (*sequential search*) dan pencarian biner (*binary search*). Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak terurut. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

1. PENCARIAN BERURUTAN (*SEQUENTIAL SEARCH*)

Algoritma pencarian dapat dijelaskan sebagai berikut : pencarian dimulai dari data paling awal, kemudian ditelusuri dengan menaikkan indeks data, apabila data sama dengan kunci pencarian dihentikan dan diberikan nilai pengembalian true, apabila sampai indeks terakhir data tidak ditemukan maka diberikan nilai pengembalian false.

Ilustrasi dari algoritma pencarian biner adalah sebagai berikut :

Kunci=3



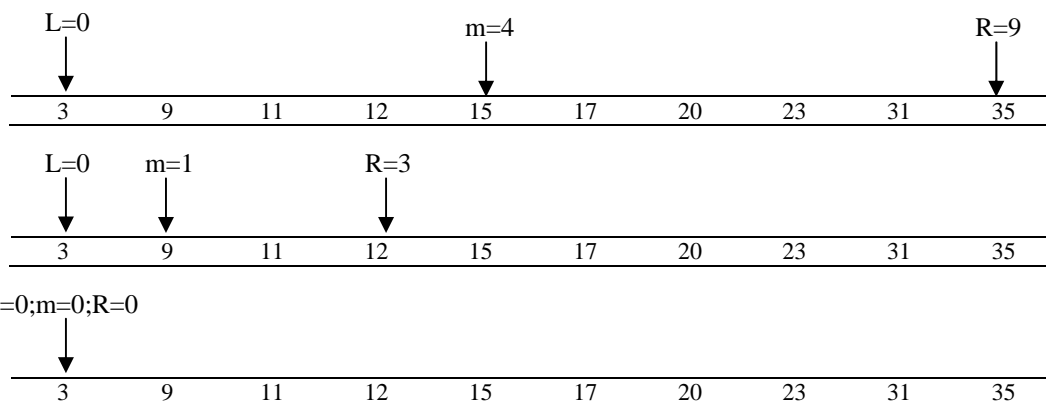
Data[4]=3 sama dengan kunci=3 maka data ditemukan dan diberikan nilai pengembalian i (posisi) dan proses dihentikan. Apabila data tidak ditemukan, maka fungsi akan mengembalikan nilai -1

2. PENCARIAN BINER (*BINARY SEARCH*)

Salah satu syarat agar pencarian biner dapat dilakukan adalah data sudah dalam keadaan urut. Dengan kata lain, apabila data belum dalam keadaan urut, pencarian biner tidak dapat dilakukan.

Ilustrasi dari algoritma pencarian biner adalah sebagai berikut :

Kunci=3



Data[1]=3 sama dengan kunci=3 maka data ditemukan dan diberikan nilai pengembalian i (posisi) dan proses dihentikan. Apabila data tidak ditemukan, maka fungsi akan mengembalikan nilai -1

TUGAS PENDAHULUAN:

1. Buatlah flowchart untuk operasi pencarian bilangan bulat dengan metode *sequential search* dan *binary search*.
2. Buatlah flowchart untuk operasi penyisipan sebelum dan sesudah serta operasi penghapusan data kunci.
3. Buatlah deklarasi data pegawai dan flowchart fungsi untuk pencarian data dengan metode *sequential search* dan *binary search* sebagai berikut :
 - Data bertipe rekaman bernama Pegawai yang mempunyai 4 data yaitu :
 - NIP, bertipe bulat
 - Nama, bertipe string

- Alamat, bertipe string
- Golongan, bertipe char
- Kunci yang digunakan untuk pencarian data adalah berdasarkan NIP dan Nama

PERCOBAAN:

1. Buatlah workspace menggunakan Visual C++.
2. Buatlah project baru *SEARCHING* yang berisi file *C source* untuk metode pencarian *sequential search* dan *binary search*.
3. Cobalah untuk masing-masing percobaan di bawah dengan menambahkan menu pilihan metode pencarian pada program utama.

Percobaan 1 : Implementasi pencarian dengan metode *sequential search*

```
#include <stdio.h>
#include <stdlib.h>

#define      MAX      10

int Data[MAX];

int SequentialSearch(int x)
{
    int i = 0;
    bool ketemu = false;

    while ((!ketemu) && (i < MAX)){
        if(Data[i] == x)
            ketemu = true;
        else
            i++;
    }
    if(ketemu)
        return i;
    else
        return -1;
}
```

```
void main()
{
    int i;

    //pembangkit bilangan random
    srand(0);

    //membangkitkan bilangan integer random
    printf("\nDATA : ");
    for (i = 0; i < MAX; i++)
    {
        Data[i] = rand()/1000+1;
        printf("%d  ", Data[i]);
    }

    int Kunci;

    printf("\nKunci : ");
    scanf("%d", &Kunci);

    int ketemu = SequentialSearch(Kunci);
    if(ketemu>0)
        printf("Data ditemukan pada posisi %d", ketemu);
    else
        printf("Data tidak ditemukan");
}
```

Percobaan 2 : Implementasi pencarian dengan metode *binary seach*

```
#include <stdio.h>
#include <stdlib.h>

#define      MAX      10

int Data[MAX];

// Prosedur menukar data

void Tukar (int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

```
// Prosedur pengurutan metode Quick Sort

void QuickSort(int L, int R)
{
    int i, j, x;
    x = Data[(L+R)/2];
    i = L;
    j = R;
    while (i <= j){
        while(Data[i] < x)
            i++;
        while(Data[j] > x)
            j--;
        if(i <= j){
            Tukar(&Data[i], &Data[j]);
            i++;
            j--;
        }
    }
    if(L < j)
        QuickSort(L, j);
    if(i < R)
        QuickSort(i, R);
}

// Fungsi pencarian biner

int BinarySearch(int x)
{
    int L = 0, R = MAX-1, m;
    bool ketemu = false;

    while((L <= R) && (!ketemu))
    {
        m = (L + R) / 2;
        if(Data[m] == x)
            ketemu = true;
        else if (x < Data[m])
            R = m - 1;
        else
            L = m + 1;
    }
    if(ketemu)
        return m;
    else
        return -1;
}
```

```
void main()
{
    int i;

    //pembangkit bilangan random
    srand(0);

    //membangkitkan bilangan integer random
    printf("\nDATA : ");
    for (i = 0; i < MAX; i++)
    {
        Data[i] = rand()/1000+1;
        printf("%d ", Data[i]);
    }

    //mengurutkan data
    QuickSort(0, MAX-1);

    int Kunci;

    printf("\nKunci : ");
    scanf("%d", &Kunci);

    int ketemu = BinarySearch(Kunci);
    if(ketemu>0)
        printf("Data ditemukan pada posisi %d", ketemu);
    else
        printf("Data tidak ditemukan");
}
```

LATIHAN:

1. Tambahkan kode program untuk menghitung jumlah perbandingan yang dilakukan dengan metode *sequential search* dan *binary search*.
2. Bandingkan kinerja pencarian dengan *sequential search* dan *binary search* berdasarkan latihan point 1.
3. Implementasikan prosedur untuk menyisipkan data sebelum dan sesudah data kunci serta prosedur untuk menghapus data kunci pada tugas pendahuluan.
4. Implementasikan pencarian data Pegawai pada tugas pendahuluan dengan ketentuan :
 - a. Metode pencarian dapat dipilih.
 - b. Pencarian dapat dipilih berdasarkan NIP, Nama dan gabungan keduanya.
 - c. Gunakan struktur data array.

- d. Lakukan penyisipan data sebelum dan sesudah data kunci serta penghapusan data kunci.
5. Berikan kesimpulan dari percobaan dan latihan yang telah Anda lakukan.