

**TUGAS PERTEMUAN 12**

**STRUKTUR DATA**

**BINARY SEARCH**



**Disusun oleh:**

**Rama Pramudya Wibisana**

**2022320019**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS INFORMATIKA**

**UNIVERSITAS BINA INSANI**

**BEKASI**

**2023**

## A. SYNTAX

```
1  #include <iostream>
2  using namespace std;
3
4  int BinarySearch(int array[], int x, int awal, int akhir)
5  {
6      while (awal <= akhir)
7      {
8          int tengah = (akhir + awal) / 2;
9
10         if (array[tengah] == x)
11             return tengah;
12         else if (array[tengah] < x)
13         {
14             awal = tengah + 1;
15         }
16         else if (array[tengah] > x)
17         {
18             akhir = tengah - 1;
19         }
20     }
21     return -1;
22 }
23
24 int main(void)
25 {
26     int array[] = {10, 12, 14, 16, 18, 20, 22, 24, 26};
27     int x = 20;
28     int n = sizeof(array) / sizeof(array[0]);
29     int result = BinarySearch(array, x, 0, n - 1);
30
31     if (result == -1)
32         printf("Nilai tidak ditemukan\n");
33     else
34         printf("Nilai ditemukan pada index %d\n", result);
35 }
```

## B. PENJELASAN

Pada tugas kali ini, kita akan membahas sedikit mengenai struktur searching pada c++. Kali ini yang akan kita gunakan sebagai contoh yakni Binary Search. Mari kita langsung masuk ke pembahasan mengenai Binary Search.

```
4  int BinarySearch(int array[], int x, int awal, int akhir)
5  {
6      while (awal <= akhir)
7      {
8          int tengah = (akhir + awal) / 2;
9
10         if (array[tengah] == x)
11             return tengah;
12         else if (array[tengah] < x)
13         {
14             awal = tengah + 1;
15         }
16         else if (array[tengah] > x)
17         {
18             akhir = tengah - 1;
19         }
20     }
21     return -1;
22 }
```

Kita membuat fungsi **BinarySearch** yang bertipe data integer, yang memiliki 3 parameter, yaitu **array[]**, **x**, **awal**, dan **akhir** yang masing-masing memiliki tipe data integer.

Selanjutnya kita akan melakukan perulangan **while** untuk menentukan nilai awal kurang dari sama dengan nilai akhir. Lalu kita membuat variabel **tengah** dengan tipe data integer yang melakukan operasi **(akhir + awal) / 2** untuk menentukan nilai tengah dari index.

Kemudian kita membuat percabangan, jika **array[tengah]** sama dengan **x** maka ia akan kembali ke nilai tengah, jika **array[tengah]** kurang dari **x** maka akan menentukan nilai awal dengan menjalankan operasi **tengah + 1**, jika **array[tengah]** lebih dari **x** maka akan menentukan nilai akhir dengan menjalankan operasi **tengah - 1**.

```

24  int main(void)
25  {
26      int array[] = {10, 12, 14, 16, 18, 20, 22, 24, 26};
27      int x = 20;
28      int n = sizeof(array) / sizeof(array[0]);
29      int result = BinarySearch(array, x, 0, n - 1);
30
31      if (result == -1)
32          printf("Nilai tidak ditemukan\n");
33      else
34          printf("Nilai ditemukan pada index %d\n", result);
35  }

```

Kita berlanjut ke fungsi utama, di sini kita akan menentukan array-nya yang menggunakan tipe data integer, memiliki 9 nilai dan 8 index **{10, 12, 14, 16, 18, 20, 22, 24, 26}**. Lalu kita akan menentukan nilai yang akan kita cari dalam index, di sini kita menentukan **x = 20**. Kemudian kita akan menentukan nilai **n**, yaitu jumlah ukuran dari index array di atas **[8]**. Setelah itu akan menampilkan hasilnya dengan memanggil fungsi **BinarySearch**.

Maka dapat ditentukan :

- nilai awal = 0
- nilai akhir =  $n - 1 = 8$
- nilai tengah =  $(8 + 0) / 2 = 4$
- nilai tengah terdapat pada index ke-4 yaitu 18

Kemudian bilangan dibandingkan :

- memanggil operasi **(array[tengah] == x)**, maka  $18 = 20 \rightarrow$  false, nilai belum sesuai
- memanggil operasi **(array[tengah] < x)**, maka  $18 < 20 \rightarrow$  true, nilai belum sesuai

Pencarian dilakukan ke sebelah kiri dengan nilai awal = tengah + 1 yaitu index ke-5, karena nilai awal tadi masih lebih kecil, maka perulangan akan dilakukan kembali, maka :

- memanggil operasi (*array[tengah] == x*), maka  $20 = 20 \rightarrow \text{true}$ ,  
nilai sudah sesuai

Nilai sudah ditemukan pada index ke-5 yaitu 20, sesuai dengan apa yang kita cari, maka proses pencarian pun dihentikan dan akan menampilkan output seperti ini :

### C. OUTPUT

```
PS D:\DATA KULIAH\TUGAS\SEMESTER 1\STRUKTUR DATA\112) Binary Search> cd "d:\DATA KULIAH\TUGAS\SEMESTER 1\STRUKTUR DATA\112) Binary Search\" ; if ($?) { g++ BinarySearch.cpp -o BinarySearch } ; if ($?) { .\BinarySearch }  
Nilai ditemukan pada index 5  
PS D:\DATA KULIAH\TUGAS\SEMESTER 1\STRUKTUR DATA\112) Binary Search> █
```