

POST-TEST

1. Buat program C++ yang melakukan pencarian linier untuk menemukan elemen dalam sebuah array.

Spesifikasi:

- a. Program meminta pengguna untuk memasukkan ukuran array dan elemen- elemen array.
 - b. Buat fungsi `linearSearch` yang menerima array dan elemen yang dicari sebagai parameter, lalu mengembalikan indeks elemen tersebut jika ditemukan atau -1 jika tidak ditemukan.
 - c. Tampilkan hasil pencarian berupa indeks elemen yang dicari atau pesan "Elemen tidak ditemukan".
2. Buat program C++ yang melakukan pencarian biner pada array yang sudah terurut.

Spesifikasi:

- a. Program meminta pengguna untuk memasukkan ukuran array dan elemen- elemen array.
- b. Sebelum melakukan pencarian biner, array harus diurutkan menggunakan algoritma pengurutan pilihan (Selection Sort).
- c. Buat fungsi `binarySearch` yang menerima array yang sudah terurut dan elemen yang dicari, lalu mengembalikan indeks elemen tersebut jika ditemukan atau -1 jika tidak ditemukan.

Jawaban:

1.

```
eighth > postTest1.cpp > findNumber(const vector<int>& arr, int n, int target)
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  void findNumber(const vector<int>& arr, int n, int target) {
7      int position;
8      int foundit = 0;
9      int i = 0;
10
11     while ((foundit == 0) && (i < n))
12     {
13         if (arr[i] == target)
14         {
15             foundit = 1;
16             position = i;
17         } else {
18             i = i + 1;
19         }
20     }
21
22     if (foundit == 0)
23     {
24         cout << "angka yang dicari tidak ditemukan" << endl;
25     } else{
26         cout << "angka " << target << " ditemukan pada index ke-" << position << endl;
27     }
28 }
29
```

```
30 int main () {
31     int i, n, target;
32
33     cout << "masukan jumlah array: ";
34     cin >> n;
35
36     vector<int> arr(n);
37
38     for ( i = 0; i < n; i++)
39     {
40         cout << "masukan nilai ke-" << i << ": ";
41         cin >> arr[i];
42     }
43
44     cout << "masukkan angka yang ingin dicari: ";
45     cin >> target;
46
47     findNumber(arr, n, target);
48
49     return 0;
50
51 }
```

Jika Benar:

```

masukan jumlah array: 5
masukan nilai ke-0: 32
masukan nilai ke-1: 87
masukan nilai ke-2: 78
masukan nilai ke-3: 45
masukan nilai ke-4: 65
masukan angka yang ingin dicari: 78
angka 78 ditemukan pada index ke-2

```

Jika salah:

```

masukan jumlah array: 5
masukan nilai ke-0: 32
masukan nilai ke-1: 34
masukan nilai ke-2: 45
masukan nilai ke-3: 55
masukan nilai ke-4: 19
masukan angka yang ingin dicari: 20
angka yang dicari tidak ditemukan

```

```

eighth > C++ postTest2.cpp > findNumber(const vector<int>&, int, int)
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  void findNumber(const vector<int>& arr, int n, int target) {
7      int left, right, result, mid;
8      left = 0;
9      right = n - 1;
10     result = -1;
11     while (left <= right)
12     {
13         mid = left + (right - left) / 2;
14
15         if (arr[mid] == target) {
16             result = mid;
17             break;
18         } else if (arr[mid] < target){
19             left = mid + 1;
20         } else{
21             right = mid - 1;
22         }
23     }
24
25     if (result != -1)
26     {
27         cout << "bilangan " << target << " ditemukan pada indeks " << result << endl;
28     } else {
29         cout << result << endl;
30     }
31 }
32

```

2.

```

33 int main () {
34     int i, j, n, target;
35
36     cout << "masukan jumlah array: ";
37     cin >> n;
38
39     vector<int> arr(n);
40
41     for ( i = 0; i < n; i++)
42     {
43         cout << "masukan nilai ke-" << i << ": ";
44         cin >> arr[i];
45     }
46
47     for ( i = 0; i < n - 1; i++)
48     {
49         for ( j = i; j < n; j++)
50         {
51             if (arr[i] > arr[j])
52             {
53                 int temp = arr[i];
54                 arr[i] = arr[j];
55                 arr[j] = temp;
56             }
57         }
58     }
59
60     cout << "masukan angka yang ingin dicari: ";
61     cin >> target;
62
63     findNumber(arr, n, target);
64
65     return 0;
66
67 }

```

Jika benar:

```

masukan jumlah array: 5
masukan nilai ke-0: 23
masukan nilai ke-0: 23
masukan nilai ke-1: 21
masukan nilai ke-2: 21
masukan nilai ke-3: 12
masukan nilai ke-4: 67
❗ masukan angka yang ingin dicari: 12
bilangan 12 ditemukan pada indeks 0

```

Jika salah:

```
masukan jumlah array: 5  
masukan nilai ke-0: 12  
masukan nilai ke-1: 23  
masukan nilai ke-2: 32  
masukan nilai ke-3: 21  
masukan nilai ke-4: 98  
masukan angka yang ingin dicari: 100  
-1
```