

**LAPORAN PRAKTIKUM  
PEMROGRAMAN 1  
MODUL 8**



Oleh:

NAMA : GIDEON TORANAWA LADIYO

NIM : 2211104022

KELAS : SE-06A

**PRODI S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2023**

## I. DASAR TEORI

### a. Konsep Dasar Searching

Searching adalah proses mendapatkan informasi berdasarkan kunci tertentu dari sejumlah informasi yang telah disimpan. Cara pencariannya adalah dengan cara mengecek datanya satu persatu secara berurutan. Jika data ditemukan, maka selesai.

### b. Linear Search

Linear search merupakan program search yang mudah dipahami, linear search memiliki kelebihan apabila data yang di cari letaknya pada data – data awal sehingga prosesnya berjalan cepat, namun apabila data yang di cari letaknya pada data terakhir maka pencarian lebih memakan waktu yang cukup lama pula. karena di linear search mengunjungi setiap elemen data yang ada.

### c. Binary Search

Binary search merupakan algoritma pencarian yang paling efisien. Algoritmanya adalah dengan membagi data menjadi dua bagian dengan syarat data tersebut telah terurut. Jika telah terurut maka perlu mencari indeks yang paling tengah dengan rumus:

$$\text{int middle} = \text{array.length}/2$$

Setelah menentukan indeks tengah maka ambil datanya. Jika kita akan mencari data yang lebih kurang dari data middle maka cari sebelah kiri, jika mencari data yang lebih besar dari data middle maka cari sebelah kanan.

## II. GUIDED

### 1. Linear Search

```
1 def linear_search(keyword, data):
2     for i in range(len(data)):
3         if str(data[i]).lower() == keyword.lower():
4             print(f"Keyword: {keyword} has found at index {i}")
5             return i
6     print(f"Keyword {keyword} not found")
7     return -1
8
9 data = [23, 3, 4, 78, 9, 10, 32]
10 keyword = input("Input keyword: ")
11 linear_search(keyword, data)
```

Output :

```
Input keyword: 21
Keyword 21 not found
```

jika angka yang dicari tidak ada di dalam data

```
Input keyword: 3
Keyword: 3 has found at index 1
```

jika angka yang dicari ada di dalam data

## 2. Binary Search

```
1 def bubble_sort(data):
2     for i in range(len(data)):
3         for j in range(len(data)-i-1):
4             if data[j] > data[j+1]:
5                 data[j], data[j+1] = data[j+1], data[j]
6     return data
7
8 def binary_search(keyword, data):
9     sort_data = bubble_sort(data)
10    print(f"data sorted = {sort_data}")
11    left = 0
12    right = len(sort_data) - 1
13    while left <= right:
14        mid = (left + right) // 2
15        str_data = sort_data[mid]
16        if str_data > keyword:
17            right = mid - 1
18        elif str_data < keyword:
19            left = mid + 1
20        else:
21            print(f"keyword {keyword} has been found at index {mid}")
22            return mid
23
24    print(f"keyword {keyword} not found")
25    return -1
26
27 data = [23, 3, 4, 78, 9, 10, 32]
28 keyword = int(input("Input keyword: "))
29 binary_search(keyword, data)
```

Output:

```
Input keyword: 3
data sorted = [3, 4, 9, 10, 23, 32, 78]
keyword 3 has been found at index 0
```

jika angka yang dicari ada di dalam data

```
Input keyword: 6
data sorted = [3, 4, 9, 10, 23, 32, 78]
keyword 6 not found
```

jika angka yang dicari tidak ada di dalam

data

### III. UNGUIDED

#### 1. Program mencari plat nomor

Code:

```
1 def linear_search(keyword, data):
2     for i in range(len(data)):
3         if data[i] == keyword:
4             print(f"Keyword {keyword} has found at index {i} in database")
5             return i
6     print(f"Keyword {keyword} not found in database")
7     return -1
8
9 data = ['R 2477 SR', 'R 1234 DJ', 'R 7015 LP',
10        'R 0201 RR', 'R 3304 DA', 'R 2401 SK',
11        'R 2103 RT', 'R 1708 RI', 'R 1111 SR',
12        'R 4987 LH']
13 keyword = input("Input Keyword: ")
14 linear_search(keyword, data)
```

Output:

```
Input Keyword: [R 2488 SR]
Keyword [R 2488 SR] not found in database
```

#### 2. Program mencari NIM

Code:

```
1 def bubble_sort(data):
2     for i in range(len(data)):
3         for j in range(len(data)-i-1):
4             if data[j] > data[j+1]:
5                 data[j], data[j+1] = data[j+1], data[j]
6
7     return data
8
9 def binary_search(keyword, data):
10     left = 0
11     right = len(data) - 1
12     while left <= right:
13         mid = (left + right)//2
14         if data[mid] > keyword:
15             right = mid - 1
16         elif data[mid] < keyword:
17             left = mid + 1
18         else:
19             print(f"Keyword {keyword} has found in index {mid}")
20             return mid
21
22     print(f"Keyword {keyword} not found")
23     return -1
24
25 data = [20103023, 20103002, 20103019,
26         20103001, 20103017, 20103005,
27         20103011, 20103003, 20103009,
28         20103021, 20103006, 20103015,
29         20103013, 20103007]
30 keyword = int(input("Input Keyword: "))
31 data.sort()
32 binary_search(keyword, data)
```

Output:

```
Input Keyword: 20103015
Keyword 20103015 has found in index 9
```

### 3. Program generate bilangan acak

Code:

```
1  def bubble_sort(data):
2      for i in range(len(data)):
3          for j in range(len(data)-i-1):
4              if data[j] > data[j+1]:
5                  data[j], data[j+1] = data[j+1], data[j]
6
7      return data
8
9  def binary_search(keyword, data):
10     left = 0
11     right = len(data) - 1
12     while left <= right:
13         mid = (left + right)//2
14         if data[mid] > keyword:
15             right = mid - 1
16         elif data[mid] < keyword:
17             left = mid + 1
18         else:
19             print(f'Keyword {keyword} has found in index {mid}')
20             return mid
21
22     print(f'Keyword {keyword} not found')
23     return -1
24
25 data = [17, 2, 15, 7, 72, 31, 12, 57, 63, 71, 23, 92, 1]
26 keyword = int(input("Masukkan angka yang dicari: "))
27 data.sort()
28 result = binary_search(keyword, data)
```

Output:

```
Masukkan angka yang dicari: 72
Keyword 72 has found in index 11
```