

# MODUL X

## TEMA

Searching/pencarian(linier search dan binary search)

## TUJUAN PRAKTIKUM

Agar mahasiswa dapat mengimplementasikan dan membuat program dengan menggunakan algoritma searching, baik linier search maupun binary search, untuk menyelesaikan permasalahan dalam kehidupan sehari-hari.

### Linier Search/sequential

Prinsip dari algoritma ini adalah membandingkan setiap elemen larik satu persatu satu sampai ditemukan nilai yang diinginkan

Array L :	54	48	37	32	29	16
	0	1	2	3	4	5

Nilai yang dicari : 29

Proses : Membandingkan 29 dengan indeks dari 0 - 5

1.  $29 == L[0] ?? \rightarrow \text{false}$  , maka di cek indeks 1
2.  $29 == L[1] ?? \rightarrow \text{false}$  , maka di cek indeks 2
3.  $29 == L[2] ?? \rightarrow \text{false}$  , maka di cek indeks 3
4.  $29 == L[3] ?? \rightarrow \text{false}$  , maka di cek indeks 4
5.  $29 == L[4] ?? \rightarrow \text{TRUE}$  : ditemukan pada indeks ke 4

### Binary search

- Terdapat algoritma pencarian nilai dari larik dengan element terurut yang memiliki kinerja lebih baik dari pada algoritma sequence yang di modifikasi
- Nama algoritma tersebut adalah pencarian “bagi dua” atau “biner”

Pencarian ini hanya bisa dilakukan pada data yang sudah diurutkan (kecil  $\rightarrow$  besar, atau besar  $\rightarrow$  kecil)  
Array L

54	48	37	32	29	16
0	1	2	3	4	5

Misal di cari nilai  $x = 16$

**Langkah 1:**  $i = 0$  dan  $j = 5$

Di cari indeks tengah  $(t) = (i+j) \div 2 = (0+5) \div 2 = 2$

54	48	37	32	29	16
0	1	2	3	4	5

54	48	37	32	29	16
0	1	2	3	4	5

Di putuskan apakah pencarian akan dilakukan pada bagian kiri atau kanan

Jika nilai pada indeks saat ini lebih besar dari yang dicari dilakukan pencarian pada bagian kanan, sebaliknya pada bagian kiri.

Jika  $L[2] > 16 ? \rightarrow \text{true}: 37 > 16$

Maka pencarian akan dilakukan pada bagian kanan:

Sehingga :  $i = t + 1 = 3$  dan  $j = 5$  dan nilai  $t = (3 + 5) \div 2 = 4$

32	29	16
3	4	5

32	29	16
3	4	5

Apakah  $L[4] == 16 ? \rightarrow \text{false}$

Di putuskan apakah pencarian akan dilakukan pada bagian kiri atau kanan

Jika nilai pada indeks saat ini lebih besar dari yang dicari dilakukan pencarian pada bagian kanan, sebaliknya pada bagian kiri.

$L[4] > 16 = 29 > 16 ? \rightarrow \text{true}$

Maka pencarian dilakukan pada bagian kanan :

16
5

Apakah  $L[5] == 16 ?? \text{TRUE.. Nilai ditemukan}$

## PRAKTIKUM

### 1) Linier Search

Buatlah script program kemudian simpan file tersebut dengan nama “*linier-search.php*”

*Script program:*

```
1 <?php
2 //Deklarasi array yang berisi kumpulan data
3 $data =[10,9,13,20,11,8,7];
4
5 $cari=11; //nilai yang akan dicari
6 $hasil="tidak ditemukan";
7 $keterangan="";
8 $posisi=-1;
9
10 //untuk membaca seluruh elmen array
11 for($i=0;$i<count($data);$i++){
12     //untuk melakukan pengecekan
13     if ($data[$i]==$cari){
14         $hasil="ditemukan";
15         $posisi=$i;//mengambil posisi nilai yang ditemukan dalam array
16         break;}//menghentikan perulangan
17     }
18
19     if ($hasil=="ditemukan"){
20         $keterangan="Angka $cari <b>'".$hasil."</b> dalam array.<br>".
21             "Ditemukan pada indeks ke-$posisi";
22     }else{
23         $keterangan="Angka $cari <b>'".$hasil."</b> dalam array."; }
24
25     echo $keterangan;
26 ?>
```

Jalankan file dengan nama “*linier-search.php*” tersebut menggunakan browser sehingga akan tampil output program sebagai berikut:

*Output program:*

← → ↻ ⓘ localhost:8080/alpro/searching/linier-search.php

Angka 11 '**ditemukan**' dalam array.  
Ditemukan pada indeks ke-4

Silahkan ganti nilai data yang dicari pada variabel **\$cari** dalam script program pada *line* 5 menjadi **\$cari = 12** kemudian simpan ulang script tersebut, lalu jalankan kembali di browser.

**Script program:**

```
5 | $cari=12; //nilai yang akan dicari
```

**Output program:**

← → ↻ ⓘ localhost:8080/alpro/searching/linier-search.php

Angka 12 '**tidak ditemukan**' dalam array.

Selain dengan cara seperti percobaan sebelumnya, metode Liner Search juga bisa dikemas dalam bentuk fungsi untuk melakukan pencarian, adapun bentuk fungsinya sebagai berikut: Buatlah script program dan simpan dengan nama "*linier-search2.php*"

**Script program:**

```
1  <?php
2      //Deklarasi array yang berisi kumpulan data
3      $data =[10,9,13,20,11,8,7];
4      $cari=10;
5      $keterangan="";
6
7      //deklarasi fungsi pencarian dengan metode liner search
8      function linier_search($data,$cari){
9          $status="tidak ditemukan";
10         $posisi=-1;
11
12         for($i=0;$i<count($data);$i++){
13             if ($data[$i]==$cari){
14                 $status="ditemukan";
15                 $posisi=$i;
16                 break;
17             }
18         }
19         /*array asosiatif untuk menampung nilai yang akan
20         |dikembalikan, berupa status dan posisi*/
21         $hasil=[
22             "status"=>$status,
23             "posisi"=>$posisi
24         ];
25         return $hasil;
26     }
27
28     //pemanggilan fungsi pencarian liner_search
29     $searching=linier_search($data,$cari);
30     if ($searching["posisi"] >=0){
31         $keterangan="Angka $cari <b>". $searching["status"]. "'</b> dalam array.<br>".
32         | "Ditemukan pada indeks ke-". $searching["posisi"];
33     }else{
34         $keterangan="Angka $cari <b>". $searching["status"]. "'</b> dalam array."; }
35
36     echo $keterangan;
37     ?>
```

### Output program:



Angka 10 **'ditemukan'** dalam array.  
Ditemukan pada indeks ke-0

## 2) Binary Search

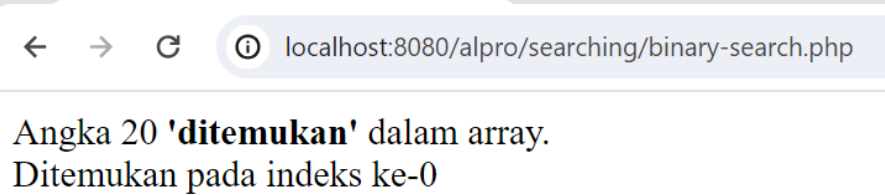
Buatlah script program kemudian simpan file tersebut dengan nama “*binary-search.php*”

### Script program:

```
1  <?php
2      /*Deklarasi array yang sudah diurutkan datanya,
3      |  bisa dari yang terbesar ke terkecil atau sebaliknya.
4      */
5      $data =[20,13,11,10,8,7,3];
6
7      $cari=11; //nilai yang akan dicari
8      $status="tidak ditemukan";
9      $hasil=-1;
10     $keterangan="";
11     //menampung nilai indek bawah, tengah dan atas
12     $bawah=0; $atas=count($data)-1; $tengah=-1;
13
14     //alur logika binary search
15     while($hasil== -1 && $bawah<=$atas){
16         $tengah=floor(($bawah+$atas)/2); //untuk mengetahui indek tengah
17
18         if ($data[$tengah]==$cari){
19             $hasil=$tengah;
20             $status="ditemukan";
21         }elseif ($data[$tengah]>$cari){
22             $bawah=$tengah+1; //untuk merubah nilai indek bawah
23         }elseif ($data[$tengah]<$cari){
24             $atas=$tengah-1; //untuk merubah nilai indek atas
25         }
26     }
27
28     if ($hasil>=0){
29         $keterangan="Angka $cari <b>'".$status."</b> dalam array.<br>".
30         |  "Ditemukan pada indeks ke-$hasil";
31     }else{
32         $keterangan="Angka $cari <b>'".$status."</b> dalam array."; }
33
34     echo $keterangan;
35     ?>
```

Jalankan file dengan nama “*binary-search.php*” tersebut menggunakan browser sehingga akan tampil output program sebagai berikut:

**Output program:**



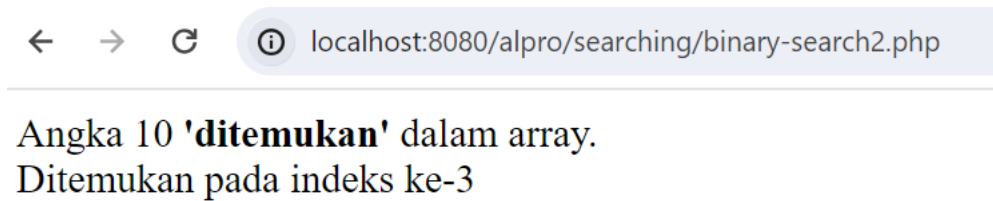
Selain dengan cara seperti percobaan sebelumnya, metode Binary Search juga bisa dikemas dalam bentuk fungsi untuk melakukan pencarian, adapun bentuk fungsinya sebagai berikut: Buatlah script program dan simpan dengan nama “*binary-search2.php*”

**Script program:**

```
1  <?php
2      /*Deklarasi array yang sudah diurutkan datanya,
3       | bisa dari yang terbesar ke terkecil atau sebaliknya.
4       */
5      $data =[20,13,11,10,8,7,3];
6      $cari=10; //nilai yang akan dicari
7      $keterangan="";
8
9      //deklarasi fungsi pencarian dengan metode binary search
10     function binary_search($data,$cari){
11         $status="tidak ditemukan";
12         $indek=-1;
13         $bawah=0; $atas=count($data)-1; $tengah=-1; //menampung nilai indeks bawah, tengah dan atas
14
15         //alur logika binary search
16         while($indek==-1 && $bawah<=$atas){
17             $tengah=floor(($bawah+$atas)/2); //untuk mengetahui indeks tengah
18
19             if ($data[$tengah]==$cari){
20                 $indek=$tengah;
21                 $status="ditemukan";
22             }elseif ($data[$tengah]>$cari){
23                 $bawah=$tengah+1; //untuk merubah nilai indeks bawah
24             }elseif ($data[$tengah]<$cari){
25                 $atas=$tengah-1; } //untuk merubah nilai indeks atas
26         }
27
28         $hasil=[
29             "status"=>$status,
30             "posisi"=>$indek
31         ];
32         return $hasil;
33     }
34
35     $searching=binary_search($data,$cari);
36     if ($searching["posisi"]>=0){
37         $keterangan="Angka $cari <b>". $searching["status"]. "</b> dalam array.<br>".
38         | "Ditemukan pada indeks ke-". $searching["posisi"];
39     }else{
40         $keterangan="Angka $cari <b>". $searching["status"]. "</b> dalam array."; }
41     echo $keterangan;
42     ?>
```

Jalankan file dengan nama “*binary-search2.php*” tersebut menggunakan browser sehingga akan tampil output program sebagai berikut:

**Output program:**



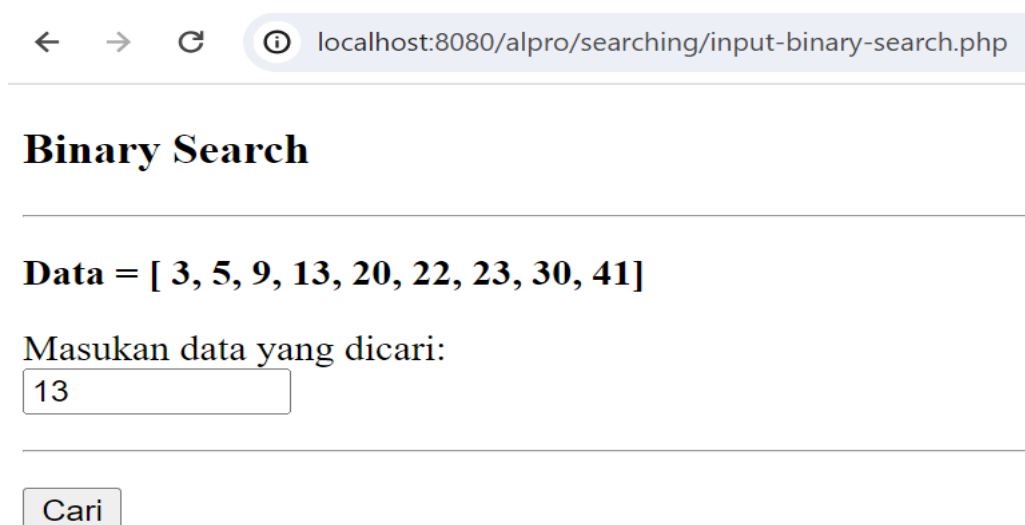
### **POS TEST:**

Buatlah program sederhana untuk proses pencarian data dengan menggunakan metode *Binary Search* dengan ketentuan sebagai berikut :

- 1) Nilai Array Data = [ 3, 5, 9, 13, 20, 22, 23, 30, 41]
- 2) Data yang dicari dan hasil pencarian menggunakan tampilan yang terpisah.
- 3) Hasil pencarian berisi informasi Angka yang dicari, status pencarian, posisi data yang dicari ada di indek ke berapa jika datanya ditemukan jika tidak ketemu tidak perlu info posisi dan berapa langkah untuk mencari data tersebut.
- 4) Logika seaching bisa menggunakan fungsi maupun tidak, diutamakan menggunakan fungsi.
- 5) Berikan coment program yang berisi informasi NIM, NAMA dan KELAS.

**Output program:**

➤ **Jika data yang dicari ditemukan**



## Informasi Pencarian :

---

Angka 13 '**ditemukan**' dalam array.  
Ditemukan pada indeks ke-3  
Proses pencarian sebanyak 4 langkah

[Kembali ke Form](#)

### ➤ Jika data yang dicari tidak ditemukan

## Binary Search

---

**Data = [ 3, 5, 9, 13, 20, 22, 23, 30, 41]**

Masukan data yang dicari:

Cari

## Informasi Pencarian :

---

Angka 21 '**tidak ditemukan**' dalam array.  
Proses pencarian sebanyak 3 langkah

[Kembali ke Form](#)