

**MODUL MATA KULIAH**

# **ALGORITMA DAN STRUKTUR DATA 1**

**KP002 – 3 SKS**



**FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA  
SEPTEMBER 2019**

**TIM PENYUSUN**

Painem, S.Kom, M.Kom  
Reva Ragam, S.Kom., M.Kom  
Ir. Moch. Sjukani



## MODUL PERKULIAHAN #3

# BINARY SEARCH

Capaian Pembelajaran	:	Mahasiswa mampu memahami konsep dasar binary search, dan mengimplementasikan dalam bahasa pemrograman
Sub Pokok Bahasan	:	1.1. Pengertian binary search 1.2. Binary Search Data yang sudah Urut 1.3. Binary Search Data belum urut
Daftar Pustaka	:	1. Sjukani M, "Struktur data dengan C++(Algoritma dan Struktur Data 2 dengan C, C++)", Mitra Wacana Media, 2007 2. Kristanto Andri, "Algoritma dan Pemrograman dengan C++", Graha Ilmu, 2003 3. Darmawan Erico, "Pemrograman Dasar C-Java-C#", Informatika, 2012

# BINARY SEARCH

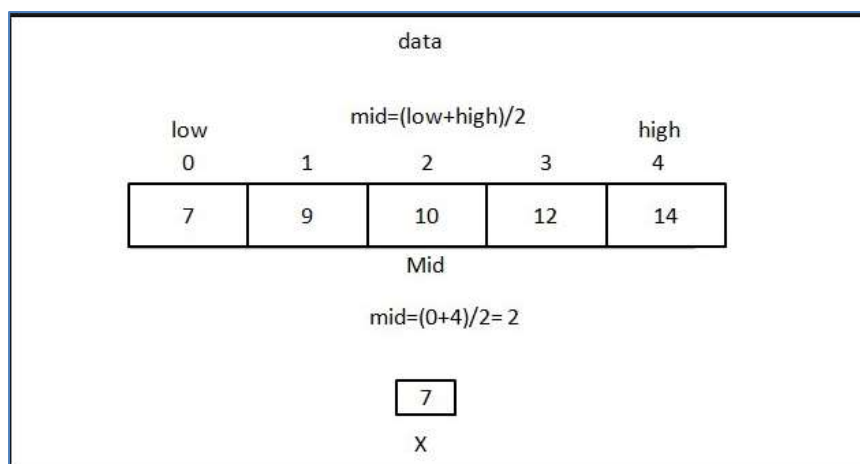
## 1.1. PENGENGERTIAN BINARY SEARCH

Metode pencarian Binary yaitu mencari data dengan melakukan mengelompokkan array menjadi bagian-bagian. Binary Search ini hanya dapat diimplementasikan pada data yang telah terurut baik ascending maupun descending dalam suatu array.

Proses Binary Search yang urutan datanya ascending:

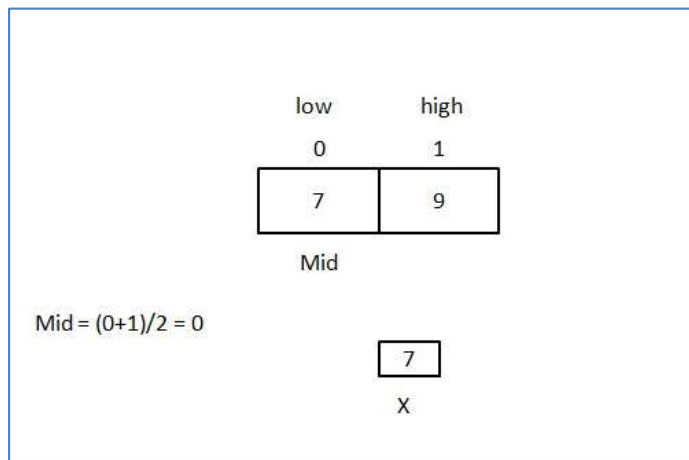
1. Pertama buat perulangan lalu menentukan posisi low yaitu posisi yang menandakan index paling rendah kemudian menentukan posisi high. Kemudian mencari posisi  $\text{mid} = (\text{high} + \text{low})/2$
2. Lalu membandingkan data yang dicari dengan nilai yang ada di posisi mid.
3. Jika data yang dicari sama dengan nilai yang ada di posisi mid berarti data ditemukan.
4. Jika data yang dicari lebih kecil dari nilai yang ada di posisi mid maka pencarian data akan dilakukan dibagian kiri mid dengan melakukan pembandingan. dengan kondisi posisi high berubah yaitu  $(\text{mid} - 1)$  dan posisi low tetap.
5. Jika data yang dicari lebih besar dari nilai yang ada mid maka pencarian data akan dilakukan di bagian kanan dari mid dengan posisi low yang berubah yaitu  $(\text{mid} + 1)$  dan posisi high tetap.

### Ilustrasi :



setelah menentukan low dan high kemudian menentukan mid. perhitungan  $\text{mid} = (\text{low} + \text{high})/2$  jadi  $\text{mid} = (0 + 4)/2$ , artinya mid berada di  $\text{data}[2]$ . Kemudian

data yang dicari yang ditampung di variabel x dibandingkan dengan data/nilai yang berada di mid. data yang dicari ialah  $7 < 10$  kemudian pencarian data dilakukan di bagian kiri mid.



Pencarian di bagian kiri mid masih dalam perulangan yang sama, namun indeksnya yang dipersempit. Karena data yang dicari lebih kecil dari data mid yang sebelumnya maka posisi high yang berubah yaitu  $(mid - 1)$  yang sebelumnya dan low tetap pada posisi yang sama. Kemudian menentukan  $mid = (0+1)/2 = 0$

artinya mid sekarang terletak di data[0]. lalu data yang dicari dibandingkan dengan mid.  $7=7$  artinya data telah ditemukan

#### **Kelebihan dan Kekurangan Binary Search:**

- **Kelebihannya** yaitu tidak perlu membandingkan data yang dicari dengan seluruh data array yang ada, cukup melalui titik tengah kemudian kita bisa menentukan ke mana selanjutnya mencari data yang ingin dicari.
- **Kekurangan** implementasi agak sedikit lebih rumit karena tidak bisa digunakan pada data array yang masih acak. Jadi harus melakukan sorting terlebih dahulu dalam implementasinya.

## 1.2. BINARY SEARCH DATA URUT ASCENDING

### Contoh (kondisi array sudah urut)

Sudah ada array satu dimensi yang dibuat dengan int array[5], sudah ada isinya dengan ilustrasi sebagai berikut :

	0	1	2	3	4
array	1	2	3	4	5

Susun program untuk menginput sebuah nilai integer (misal N). Kemudian mencari isi array, apakah ada isi array yang (nilainya) sama dengan N. Bila ada, cetak perkataan "ADA". Bila tidak ada cetak perkataan "TIDAK ADA" . Metode pencarian dengan metode binary search

### Program binary search :


```
1  #include <stdio.h>
2  main()
3  {
4      int array[5] = {1,2,3,4,5};
5      int cari;
6      int posisiTerendah=0, posisiTertinggi=5, posisiTengah;
7
8      for(int i=0;i<5;i++)
9      {
10         printf("Data %i tersimpan pada urutan ke -%i",array[i],i+1);
11         printf("\n");
12     }
```

```

13 //proses pencarian dengan binary search
14 printf("Masukkan Bilangan yang dicari=");scanf("%i",&cari);
15 while (posisiTerendah <= posisiTertinggi)
16 {
17     posisiTengah = (posisiTerendah + posisiTertinggi)/2;
18     if (cari > array[posisiTengah] )
19     {
20         posisiTerendah = posisiTengah + 1;
21     }
22     else if (cari < array[posisiTengah])
23     {
24         posisiTertinggi= posisiTengah - 1;
25     }
26     else
27     {
28         posisiTerendah = posisiTertinggi +1;
29     }
30 }
31 if (cari == array[posisiTengah])
32 {
33     printf("Data ditemukan pada urutan ke -%i",posisiTengah+1);
34     printf("\n");
35 }
36 else{
37     printf("Data yang dicari tidak ditemukan\n");
38 }
39 }
40 }

```

### Output program binary search :

 C:\Users\iin\SkyDrive\Documents\binarysearchprintf.exe

```

Data 1 tersimpan pada urutan ke -1
Data 2 tersimpan pada urutan ke -2
Data 3 tersimpan pada urutan ke -3
Data 4 tersimpan pada urutan ke -4
Data 5 tersimpan pada urutan ke -5
Masukkan Bilangan yang dicari=3
Data ditemukan pada urutan ke -3
-----

```

```
C:\Users\iin\SkyDrive\Documents\binarysearchprintf.exe
Data 1 tersimpan pada urutan ke -1
Data 2 tersimpan pada urutan ke -2
Data 3 tersimpan pada urutan ke -3
Data 4 tersimpan pada urutan ke -4
Data 5 tersimpan pada urutan ke -5
Masukkan Bilangan yang dicari=9
Data yang dicari tidak ditemukan
-----
```

### 1.3. BINARY SEARCH DATA BELUM URUT

Untuk binary search jika data belumurut baik secara ascending maupun descending maka harus dilakukan proses pengurutan terlebih dahulu.

**program kondisi array belum urut :**

```
1  #include<iostream>
2  using namespace std;
3  int main () {
4      int n, angka[12], kiri, kanan, tengah, temp, key;
5      bool ketemu = false;
6
7      cout << "Masukan jumlah data= ";
8      cin >> n;
9
10     for(int i=0; i<n; i++)
11     {
12
13         cout << "Angka ke - [" << i << "]" : ";
14         cin >> angka[i];
15     }
```

```

16 //proses mengurutkan array
17 for (int i=0; i<n; i++)
18 {
19     for(int j=0; j< n-1; j++)
20     {
21
22         if(angka [j] > angka [j+1])
23         {
24             temp = angka[j];
25             angka[j] = angka[j+1];
26             angka[j+1] = temp;
27         }
28     }
29 }
30 cout << "=====\\n";
31 cout << "Data yang telah diurutkan adalah:\\n";
32 for(int i=0; i<n; i++)
33 {
34     cout << angka[i] << " ";
35 }
36 cout << "\\n=====\\n";
37 cout << "Masukan angka yang dicari: ";
38 cin >> key;

39 kiri = 0;
40 kanan = n-1;
41 while(kiri<=kanan)
42 {
43     tengah=(kiri + kanan) /2;
44     if(key == angka[tengah])
45     {
46
47         ketemu=true;
48         break;
49     }
50     else if (key < angka [tengah])
51     {
52         kanan = tengah -1;
53     }
54     else
55     {
56         kiri = tengah +1;
57     }
58 }

```



```

59     if (ketemu == true)
60     {
61
62         cout << "Angka ditemukan! ";
63         cout << "Data berada pada indek ke-" << tengah;
64     }
65     else
66         cout << "Angka tidak ditemukan!";
67     return 0;
68 }

```

**Output program kondisi belum urut :**

```

D:\Raker RPS 13 -14 Feb 2018\program\binarysearhsort1.exe
Masukan jumlah data= 5
Angka ke - [0] : 10
Angka ke - [1] : 30
Angka ke - [2] : 16
Angka ke - [3] : 14
Angka ke - [4] : 19
=====
Data yang telah diurutkan adalah:
10 14 16 19 30
=====
Masukan angka yang dicari: 30
Angka ditemukan! Data berada pada indek ke-4
=====

```

## SOAL LATIHAN

### Soal -1

**Sudah ada array satu dimensi yang dibuat dengan int A[5], sudah ada isinya dengan ilustras sebagai berikut :**

	0	1	2	3	4
A	15	2	14	35	5

Susun program untuk menginput sebuah nilai integer (misal N). Kemudian mencari isi array, apakah ada isi array yang (nilainya) sama dengan N. Bila ada, cetak perkataan "ADA". Bila tidak ada cetak perkataan "TIDAK ADA" . Metode pencarian dengan metode binary search

### Contoh output :

Data array sebelum disorting

15 2 14 35 5

Data array setelah di sorting

2 5 14 15 35

Masukkan data yang ingin di cari : 15

DATA DITEMUKAN

Posisi data di urutan = 3

## KESIMPULAN

1. Metode pencarian Binary yaitu mencari data dengan melakukan mengelompokkan array menjadi bagian-bagian. Binary Search ini hanya dapat diimplementasikan pada data yang telah terurut baik ascending maupun descending dalam suatu array.
2. Jika kondisi array belum urut sebelum melakukan binary search maka data array harus diurutkan terlebih dahulu
3. Proses Binary Search yang urutan datanya ascending:
  - a. Pertama buat perulangan lalu menentukan posisi low yaitu posisi yang menandakan index paling rendah kemudian menentukan posisi high. Kemudian mencari posisi  $mid = (high + low)/2$
  - b. Lalu membandingkan data yang dicari dengan nilai yang ada di posisi mid.
  - c. Jika data yang dicari sama dengan nilai yang ada di posisi mid berarti data ditemukan.
  - d. Jika data yang dicari lebih kecil dari nilai yang ada di posisi mid maka pencarian data akan dilakukan dibagian kiri mid dengan melakukan pembandingan. dengan kondisi posisi high berubah yaitu  $(mid - 1)$  dan posisi low tetap.
  - e. Jika data yang dicari lebih besar dari nilai yang ada mid maka pencarian data akan dilakukan di bagian kanan dari mid dengan posisi low yang berubah yaitu  $(mid + 1)$  dan posisi high tetap.



**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS BUDI LUHUR**

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>