Algoritma Pemrograman dan Struktur Data

Materi 3: LINEAR SEARCH DAN BINARY SEARCH

Dosen pengampu:
Suamanda Ika Novichasari, M.Kom.
Imam Adi Nata, M.Kom

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR

Jl. Kapten Suparman No.39, Tuguran, Potrobangsan, Kec. Magelang Utara, Kota Magelang, Jawa Tengah 56116

Learning Objective

Mahasiswa mampu menjelaskan konsep Linear Search

Mahasiswa mampu menjelaskan konsep Binary Search

Course Material

Linear Search

Binary Search

Pre Test 5 Menit

- Apa perbedaan dari **Linear Search** dan **Binary Search** ??

BAB MATERI

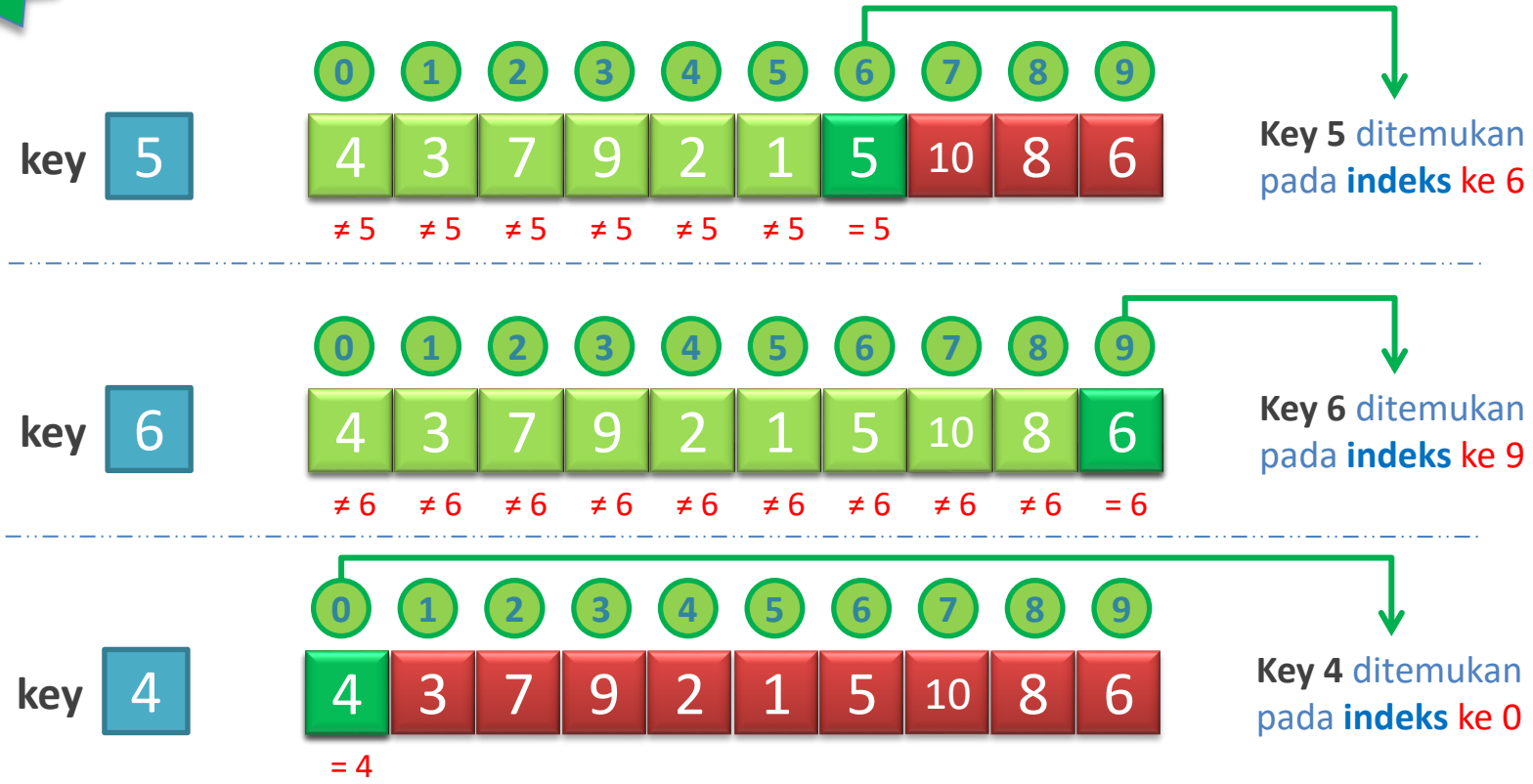
Konsep Linear Search

Subbab ini mempelajari konsep linear search

Apa itu *Linear Search ?*

- Linear Search = Sequential Search
- Merupakan algoritma pencarian brute force
- Proses pencarian membandingkan nilai Kunci dengan semua elemen nilai
- Membandingkan nilai Kunci dengan elemen pertama sampai elemen terakhir, atau
- Proses terhenti jika nilai kunci cocok dengan nilai elemen tanpa harus membandingkan semua elemen

Contoh *Linear Search*



Pseudocode : *Linear Search*

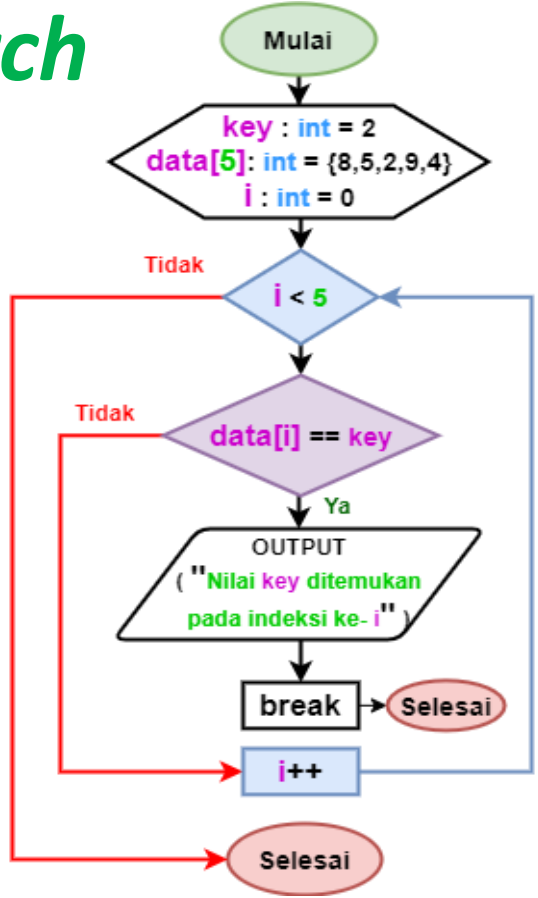
Algoritma Pencarian Linear
{ Mencari data dalam array bertipe data integer. Algoritma menerima masukan kata kunci, mencari kata kunci dalam array, lalu cetak indeks kata kunci yang ditemukan}
Deklarasi :
Data[5] : Integer = {8,5,2,9,4} {tipe data bilangan bulat}
key : integer {tipe data bilangan bulat}
i: integer {tipe data bilangan bulat}
Algoritma :

```
1.  INPUT (key)
2.  FOR (i = 0 ; i < 5 ; i++)
3.      IF (data[i] == key)
4.          OUTPUT (“Nilai” + key + “di indeks” + i)
5.      ENDIF
6.  ENDFOR
```

02

Flowchart : *Linear Search*

Contoh flowchart teknik pencarian Linear untuk *array* 1 dimensi yang memiliki panjang 5 elemen dengan nilai awal



Karakteristik *Linear Search*

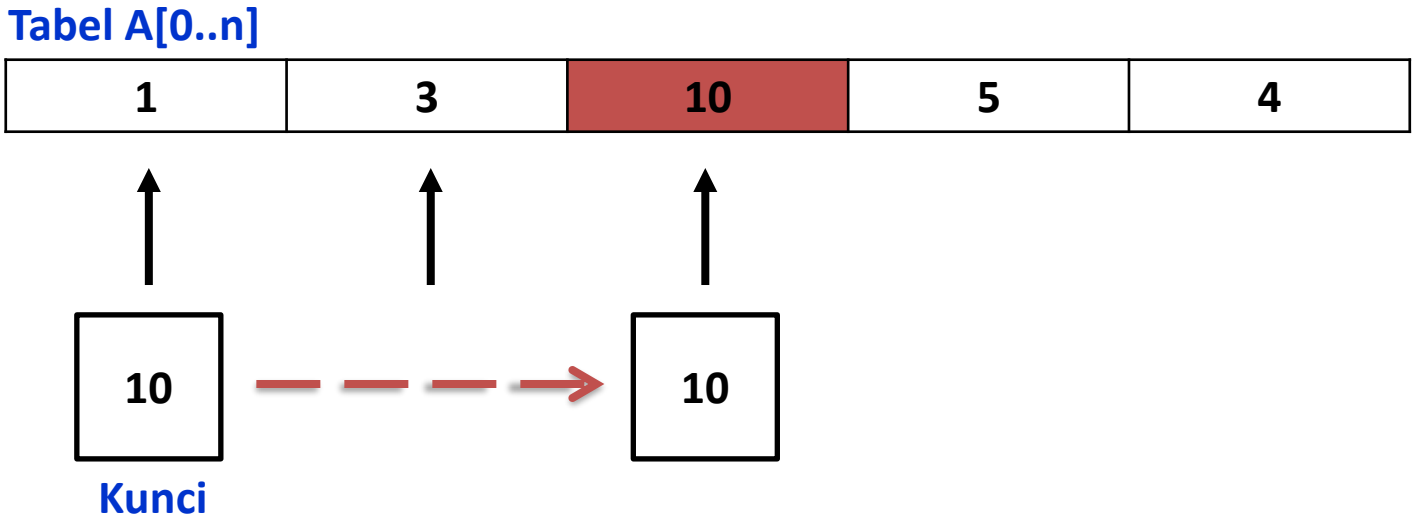
Kelebihan

- Metode pengurutan yang paling sederhana
- Dapat dilakukan pada data yang acak
- Efisien jika data yang dicari terdapat pada elemen pertama

Kekurangan

- **Tidak efisien** jika data banyak dan kata kunci yang dicari terletak pada elemen terakhir
- Kecepatan penemuan kunci tergantung pada besar data dan letak kunci.

Contoh



Proses pencarian berhenti jika nilai kunci cocok dengan nilai elemen tanpa harus membandingkan semua elemen

Latihan

Diberikan tabel berisi integer A[0..n] yang telah diisi
n=5, Tabel A berisi {3,1,2,4,6} dengan nilai kunci adalah K= 5

Diberikan tabel berisi integer A[0..n] yang telah diisi
n=5, tabel A berisi {10,12,9,7,20} dengan nilai kunci adalah K=9

Lakukan proses pencarian sequential search dari contoh diatas dan jelaskan outputnya ?

BAB MATERI

Konsep Binary Search

Subbab ini mempelajari konsep binary search

Apa itu *binary search*?

Binary Search adalah metode/algoritma pencarian.

Proses = dengan mencari suatu data yang **terurut**.

1) Membagi data tersebut menjadi dua bagian yang sama
2) Fokus pada data tengah suatu data
3) Membandingkan dengan data yang kita cari.

Metode *Binary Search*

1) Tentukan batas bawah (***low***), nilai tengah (***mid***), batas atas (***high***).

   mid = ***low*** + (***high*** - ***low***) / 2

   low = indeks awal

   high = indeks akhir

2) Bandingkan nilai ***key*** dengan nilai ***mid*** elemen *array*.

3) Jika nilai ***key*** **sama dengan** nilai pada indeks ***mid***, maka pencarian selesai.

4) Jika nilai ***key*** **tidak sama dengan** nilai pada indeks ***mid***, maka:

   - **Jika** nilai ***key*** **>** nilai indeks ***mid***
   - ***low*** = *mid + 1*

   - **Jika** nilai ***key*** **<** nilai indeks ***mid***
   - ***high*** = *mid - 1*

5) Kembali lagi ke langkah 1

key 3

0	1	2	3	4	5	6	7	8	9
4	3	7	9	2	1	5	10	8	6

1	2	3	4	5	6	7	8	9	10

diurutkan

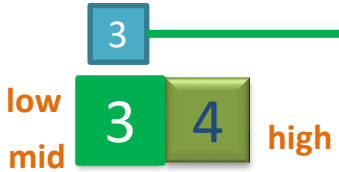
- Bandingkan nilai **mid** dengan **key**
- **Jika tidak cocok**, bagi menjadi dua sisi

- **key < mid**, maka abaikan sisi high

- Bandingakan nilai **mid** dengan **key**
- **Jika tidak cocok**, bagi menjadi dua sisi

- **key > mid**, maka abaikan sisi low

- Bandingakan nilai **mid** dengan **key**
- **Jika cocok**, maka key ditemukan



Key 3 ditemukan pada **indeks** ke 2

Pseudocode : *Binary Search*

Algoritma Binary Search
{ Mencari suatu data dalam array pada array yang sudah terurut. Membandingkan key dengan nilai tengah array sampai ditemukan datanya, kemudian menampilkan datanya}
Deklarasi :

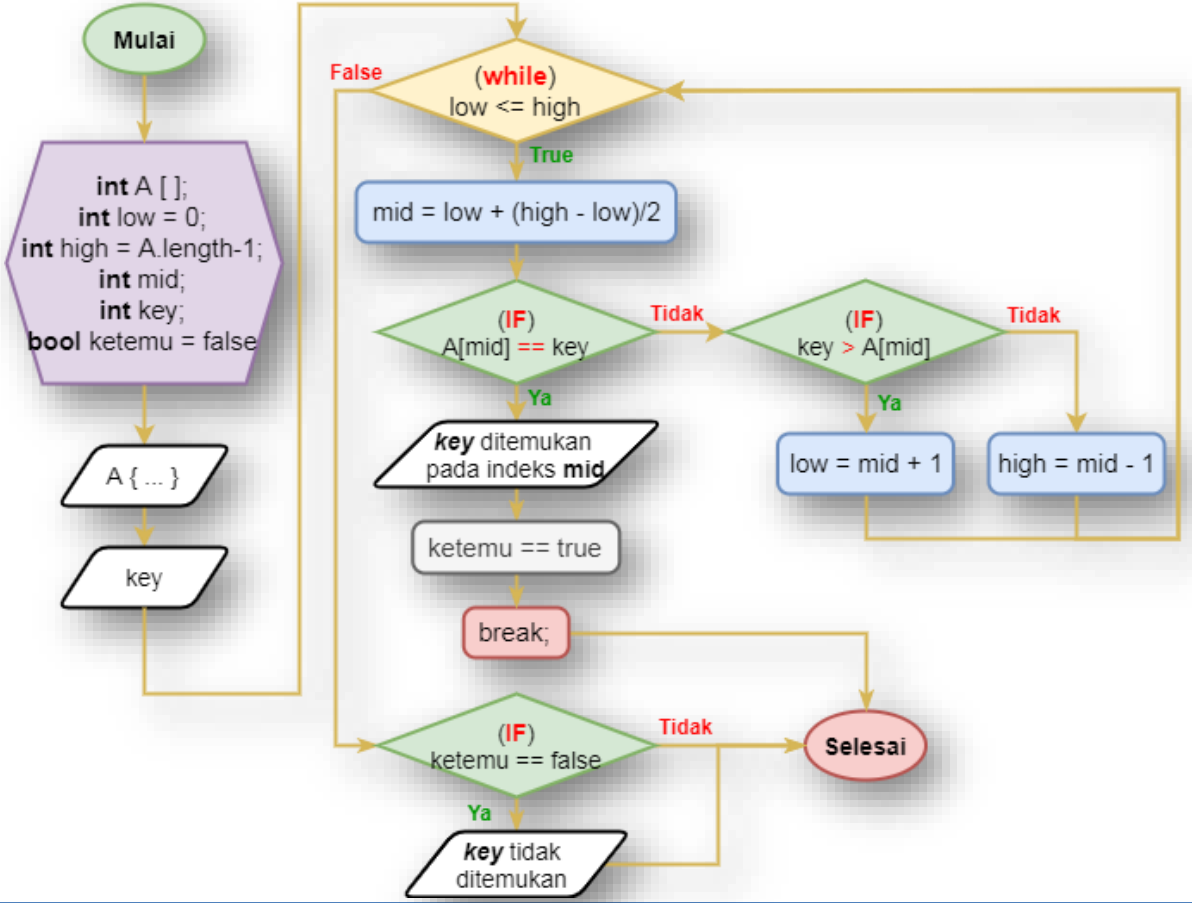
```
data[n] : integer          {tipe data bilangan bulat, n adalah panjang array}
low  : integer = 0         {tipe data bilangan bulat}
high : integer = n-1       {tipe data bilangan bulat}
mid  : float = 0  {tipe data bilangan pecahan}
key  : integer    {tipe data bilangan bulat}
```

Algoritma :

```
1.      INPUT (data[...], key)
2.      WHILE (low <= high)
3.          mid ← low + ((high - low)/2)
4.          IF (key == data[mid] )
5.              OUTPUT (“key ditemukan pada indeks ke-” mid)
6.              BREAK
7.          ELSEIF (key > data[mid])
8.              low ← mid + 1
9.          ELSE
10.             high ← mid – 1
11.         ENDIF
12.     ENDWHILE
13.     IF (low > high )
14.         OUTPUT (“key tidak ditemukan)
15.     ENDIF
```

Flowchart

Binary Search



Karakteristik *Binary Search*

Kelebihan

- Metode pengurutan yang lebih cepat dibandingkan linear search
- Efisien digunakan pada data yang besar
- Efisien jika data yang dicari terdapat pada elemen tengah

Kekurangan

- **Tidak efisien** jika data yang dicari terletak pada elemen pertama atau terakhir.
- Hanya dapat digunakan pada data yang sudah terurut.

Contoh

Diberikan sebuah tabel integer A[0..n]

{3, 14, 27, 31, 39, 42, 55, 70, 74, 80, 85, 93, 98}

lakukan pencarian menggunakan binary search dengan nilai kunci

K = 70

K = 70

index	0	1	2	3	4	5	6	7	8	9	10	11	12
Nilai	3	14	27	31	39	42	55	70	74	80	85	93	98
iterasi-1	*l*						*m*						*r*

l = left ; m = middle ; r = right

Nilai tengah dapat ditentukan dengan rumus

(l + r) / 2

K = 70

index	0	1	2	3	4	5	6	7	8	9	10	11	12
Nilai	3	14	27	31	39	42	55	70	74	80	85	93	98
iterasi-1	*l*						*m*						*r*
iterasi-2								*l*		*m*			*r*

l = left ; m = middle ; r = right

Karena K > A[m] maka pencarian dilakukan setelah indeks nilai tengah dengan ketentuan *l = m + 1*

K = 70

index	0	1	2	3	4	5	6	7	8	9	10	11	12
Nilai	3	14	27	31	39	42	55	70	74	80	85	93	98
iterasi-1	*l*						*m*						*r*
iterasi-2								*l*		*m*			*r*
iterasi-3								*l,m*	*r*				

l = left ; m = middle ; r = right

Karena K < A[m] maka pencarian dilakukan sebelum indeks nilai tengah dengan ketentuan *r = m - 1*

Tugas Binary Search

- Diberikan tabel berisi integer A[n] yang telah diisi

  n=5, Tabel A berisi {1,3,5,6,9} dengan nilai kunci adalah K= 6

- Diberikan tabel berisi integer A[n] yang telah diisi

  n=5, tabel A berisi {10,12,15,17,20} dengan nilai kunci adalah K=9

Jelaskan outputnya ? Bandingkan outputnya dengan Sequential Search

Rangkuman

Binary Search

Kelebihan

- Metode pengurutan yang lebih cepat dibandingkan linear search
- Efisien digunakan pada data yang besar
- Efisien jika data yang dicari terdapat pada elemen tengah

Kekurangan

- **Tidak efisien** jika data yang dicari terletak pada elemen pertama atau terakhir.
- Hanya dapat digunakan pada data yang sudah terurut.

Linear Search

Kelebihan

- Metode pengurutan yang paling sederhana
- Dapat dilakukan pada data yang acak
- Efisien jika data yang dicari terdapat pada elemen pertama

Kekurangan

- **Tidak efisien** jika data banyak dan kata kunci yang dicari terletak pada elemen terakhir
- Kecepatan penemuan kunci tergantung pada besar data dan letak kunci.

TERIMA KASIH

#NEXT... Running Time & Big O Notation