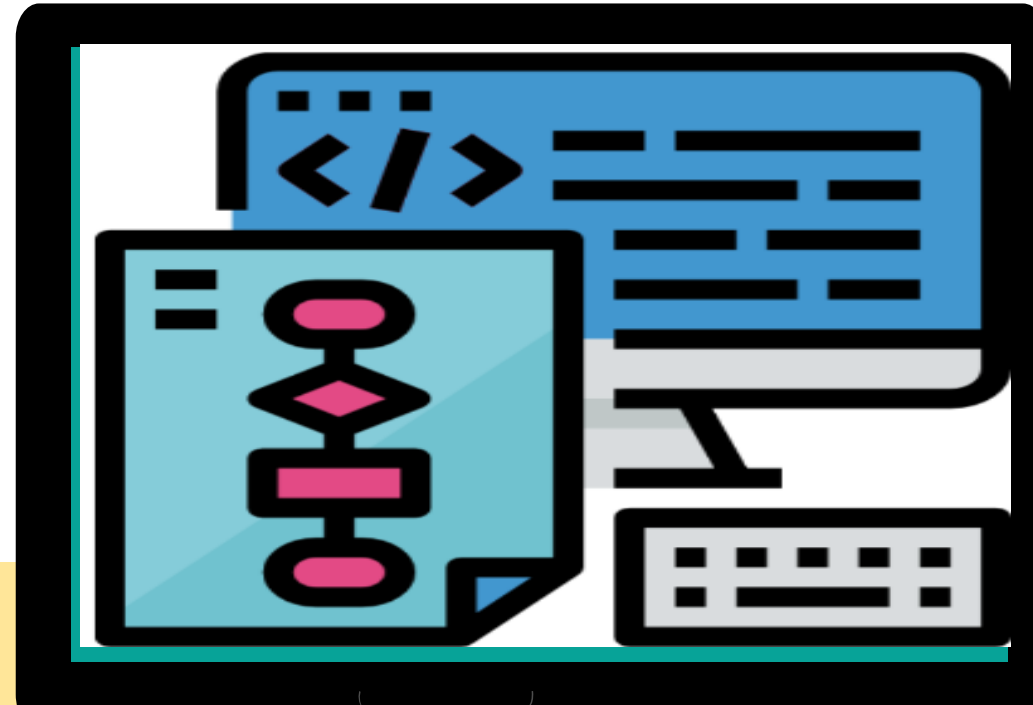




KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
DIREKTORAT JENDERAL PENDIDIKAN TINGGI, RISET, DAN TEKNOLOGI  
DIREKTORAT PEMBELAJARAN DAN KEMAHASISWAAN



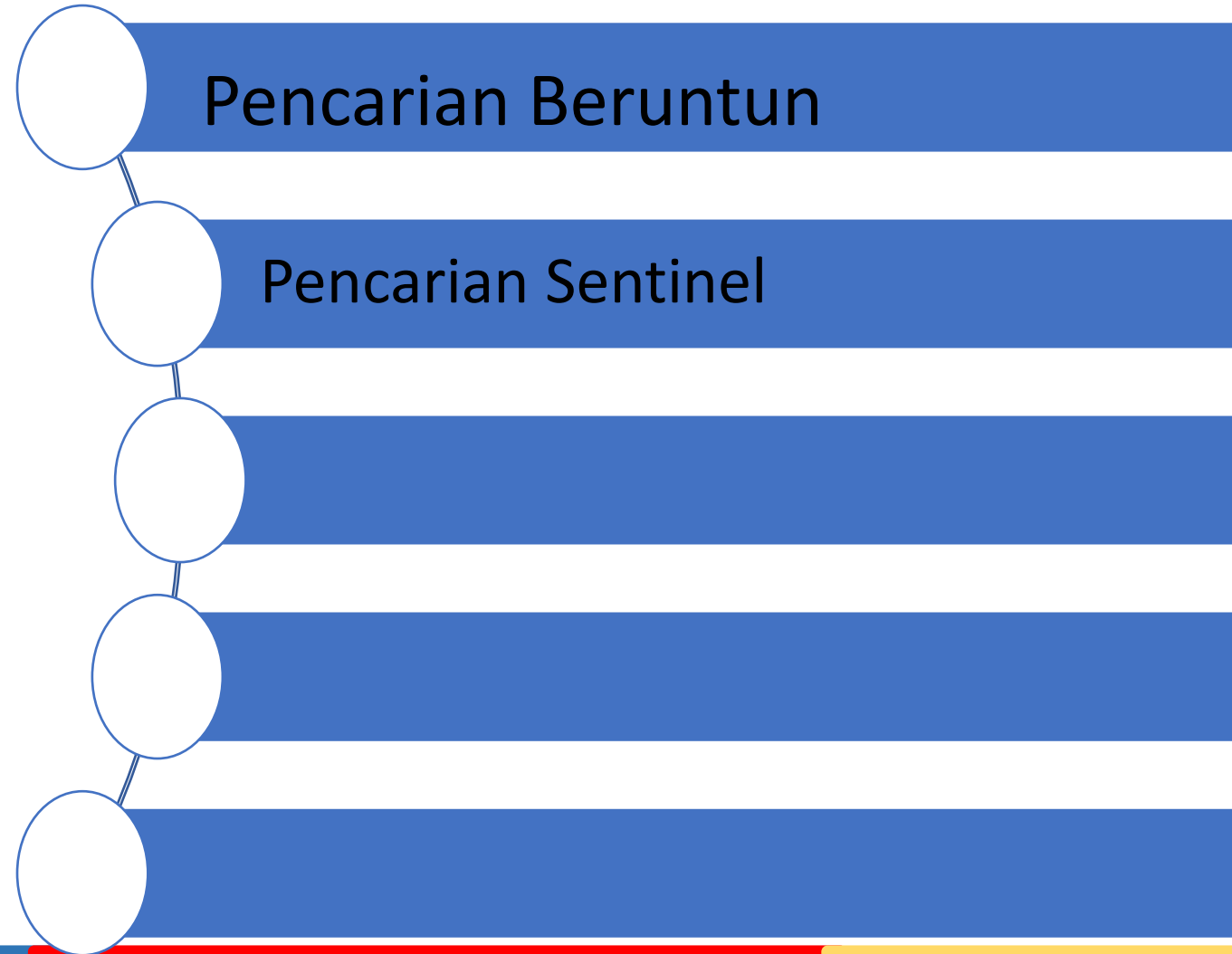
Kampus  
Merdeka  
INDONESIA JAYA



# ALGORITMA DAN PEMROGRAMAN

[Komang Aryasa | [Pertemuan 20]

# Outline



# Konsep Pencarian Data

- Pencarian elemen (searching) merupakan proses fundamental dalam pemrograman, dimana elemen akan dicari posisi-nya. Berbagai proses dalam aplikasi memerlukan searching, antara lain:
- Proses editing (perbaikan dan peremajaan data) selalu didahului dengan pencarian akan posisi data yang akan di-edit.
- Proses inserting (penyisipan data) memerlukan searching posisi dimana suatu data akan di-sisipkan.
- Program pengolah kata (word atau text processing) menyediakan fasilitas *Find*, *Replace*, dan *Goto* yang pada hakekatnya memerlukan teknik pencarian.

# Pencarian elemen Larik

- Diberikan sebuah larik A yang elemen-elemennya sudah ditetapkan. Terdapat sebuah elemen x yang tipe-nya sama dengan elemen larik A. Selanjutnya akan dicari apakah ada elemen dalam A yang sama nilainya dengan x. Bila ada maka tampilkan bahwa "x ditemukan". Bila terjadi sebaliknya, tampilkan "x tidak ditemukan".
- Diberikan sebuah larik A yang sudah terisi penuh. Terdapat sebuah elemen x yang tipe-nya sama dengan elemen larik A. Selanjutnya akan dicari apakah ada elemen dalam larik A yang sama nilainya dengan x. Bila ada maka tentukan posisinya (indeks) dari elemen tersebut. Bila terjadi sebaliknya, nyatakan indeks=0 atau -1.



# Pencarian elemen Larik

- Diberikan sebuah larik A yang sudah terisi penuh. Kemudian ada sebuah elemen x yang tipe-nya sama dengan elemen larik A sehingga akan dicari apakah ada elemen dalam larik A yang bernilai sama dengan x. Bila ada maka berikan nilai *true* kepada suatu variabel logis (boolean). Bila tidak ada maka berikan nilai *false* kepada variabel tersebut.



# Sequential Search

- Pencarian secara beruntun (sequential search) dilakukan dengan membandingkan  $x$  dengan setiap elemen larik mulai dari indeks pertama hingga indeks terakhir
- Apabila  $x$  ada dalam larik maka lokasi data tersebut memiliki indeks diantara 0 s/d  $(n-1)$
- Apabila indeks  $>$  indeks terakhir berarti  $x$  tidak ada dalam larik

# Contoh

- Andaikan larik :

$$A = [ 20 \quad 15 \quad 18 \quad 35 \quad 40 \quad 27 ],$$

dan  $x = 18$  dan  $45$ .

- Bila  $A$  diperiksa mulai dari indeks=1 maka  $x$  akan ditemukan pada indeks=3.
- Namun bila  $x=45$  maka pemeriksaan tidak berhasil menemukan karena hingga indeks=6 elemen ini tetap tidak ditemukan.



## Algoritma Sequential\_V.1



- { algoritma pencarian beruntun dimana elemen-elemen A tidak sorted dan akan ditampilkan kalimat bila elemen ditemukan }
- 
- **Deklarasi**
- **integer** m = 10;
- **integer** A[m], x;
- **integer** indeks;
- 
- **Deskripsi**
- { baca elemen larik A }
- **for** ( indeks=0 **to** m-1 **step** 1)
- **write** ( "masukkan elemen ke-", indeks );
- **read** ( A[indeks] );
- **endfor.**
- 
- { baca elemen x }
- **write** ( "masukkan elemen x : " );
- **read** ( x );
-





- { mencari x dalam A }
- indeks  $\leftarrow$  0;
- **while** ( indeks < m && x  $\neq$  A[indeks] ) **do**
- indeks  $\leftarrow$  indeks + 1;
- **endwhile.**
- 
- { menetapkan hasilnya }
- **if** ( x = A[indeks] )
- **then write** ( "x ditemukan " );
- **else write** ( "x tidak ditemukan " );
- **endif.**
- 
-



## Algoritma Sequential\_V.2



- { algoritma pencarian beruntun dimana elemen-elemen A tidak sorted dan akan ditampilkan posisi atau indeks bila elemen ditemukan }
- 
- **Deklarasi**
- **integer** m = 10;
- **integer** A[m], x;
- **integer** indeks, posisi;
- 
- **Deskripsi**
- { baca elemen matriks A }
- **for** ( indeks=0 **to** m-1 **step** 1)
- **write** ( "masukkan elemen ke-", indeks );
- **read** ( A[indeks] );
- **endfor.**
- 
- { baca elemen x }
- **write** ( "masukkan elemen x : " );
- **read** ( x );
-

- { mencari x dalam A }
- indeks  $\leftarrow$  0;
- **while** ( indeks < m && x  $\neq$  A[indeks] ) **do**
- indeks  $\leftarrow$  indeks + 1;
- **endwhile.**
- 
- { menetapkan hasilnya }
- posisi  $\leftarrow$  -1;
- **if** ( x = A[indeks] )
- **then** posisi  $\leftarrow$  indeks;
- **endif.**
- **write** ( "posisi data= ", posisi );



## Algoritma Sequential\_V.3



- { algoritma pencarian beruntun dimana elemen-elemen A tidak sorted dan akan ditetapkan variabel boolean apakah true atau false }
- 
- **Deklarasi**
- **integer** m = 10;
- **integer** A[m], x;
- **integer** indeks;
- **boolean** ketemu;
- 
- **Deskripsi**
- { baca elemen matriks A }
- **for** ( indeks=0 **to** m-1 **step** 1)
- **write** ( "masukkan elemen ke-", indeks );
- **read** ( A[indeks] );
- **endfor.**
- 
- { baca elemen x }
- **write** ( "masukkan elemen x : " );
- **read** ( x );



{ mencari x dalam A }

- indeks  $\leftarrow 0$ ;
- **while** ( indeks  $< m$  **&&** x  $\neq$  A[indeks] ) **do**
- indeks  $\leftarrow$  indeks + 1;
- **endwhile.**
- 
- { menetapkan hasilnya }
- ketemu  $\leftarrow$  **false**;
- **if** ( x = A[indeks] )
- **then** ketemu  $\leftarrow$  **true**;
- **endif.**
- 
- **write** ( "ketemu = ", ketemu);
-



# Pencarian dengan Sentinel

- Sentinel pada hakekatnya adalah elemen fiktif yang ditambahkan ke dalam suatu larik pada posisi terakhir yang nilainya sama dengan nilai dari elemen yang dicari.
- Pada proses pencarian ada dua kemungkinan yaitu:
- elemen ditemukan pada posisi indeks antara 1 sampai dengan  $m$  (indeks maksimum). Artinya, elemen yang dicari benar ada di dalam larik,
- Atau, elemen ditemukan pada posisi indeks  $(m+1)$  atau elemen sentinel. Artinya, sebenarnya elemen yang dicari tidak ada di dalam larik.



## Algoritma Sentinel\_Search

- { melakukan pencarian dengan menambahkan elemen sentinel }

•

### • Deklarasi

- **integer** m = 10;
- **integer** A[m+1], x;
- **integer** indeks;

•

### • Deskripsi

- { baca elemen larik A }
- **for** ( indeks=0 **to** m-1 **step** 1)
- **write** ( "masukkan elemen ke-", indeks );
- **read** ( A[indeks] );
- **endfor.**

•

- { baca elemen x }
- **write** ( "masukkan elemen x : " );
- **read** ( x );

•





## { menambahkan sentinel }

- $A[m] \leftarrow x;$
- 
- { mulai pencarian }
- indeks  $\leftarrow 0;$
- **while** (  $x \neq A[\text{indeks}]$  ) **do**
- indeks  $\leftarrow$  indeks + 1;
- **endwhile.**
- 
- { menetapkan hasil }
- **if** ( indeks < m )
- **then write** ( " x ditemukan pada posisi ",  
indeks );
- **else write** ( " x tidak ditemukan dalam larik "
- );
- **endif.**