

# ARRAY DAN PEMROSESANNYA – PHYTON

Tim Penyusun Materi WI1102 Berpikir Komputasional  
Institut Teknologi Bandung © 2024

# TUJUAN

Mahasiswa mampu:

- Menjelaskan definisi dan kegunaan array
- Menggunakan menggunakan notasi pendefinisian dan pengacuan array dengan benar
- Melakukan pemrosesan sekuensial pada array

# KOMBINASI PASANGAN NAMA – 3 NAMA

- Tuliskan program yang menerima 3 nama, lalu menampilkan semua kombinasi pasangan nama.
- Contoh keluaran:

Ali  
Budi  
Caca  
**Ali - Budi**  
**Ali - Caca**  
**Budi - Caca**

```
# KAMUS
# nama1, nama2, nama3 : string

# ALGORITMA
nama1 = input()
nama2 = input()
nama3 = input()

print(nama1, " - ", nama2)
print(nama2, " - ", nama3)
print(nama3, " - ", nama1)
```



# KOMBINASI PASANGAN NAMA – 10 NAMA

- Tuliskan program yang menerima 10 nama, lalu menampilkan semua kombinasi pasangan nama.

- Contoh keluaran:

**Ali - Budi**  
**Ali - Caca**  
...  
**Ina - Jaja**

```
# KAMUS
# nama1,nama2,nama3,nama4,nama5 : string
# nama6,nama7,nama8,nama9,nama10 : string

# ALGORITMA
nama1 = input()
nama2 = input()
nama3 = input()
# ... Lanjutkan sendiri
nama10 = input()

print(nama1, " - ", nama2)
print(nama2, " - ", nama3)
print(nama3, " - ", nama4)
# ... Lanjutkan sendiri
print(nama10, " - ", nama1)
```





# BAGAIMANA KALAU...

Anda diminta menampilkan semua kombinasi pasangan nama yang mungkin dari ...

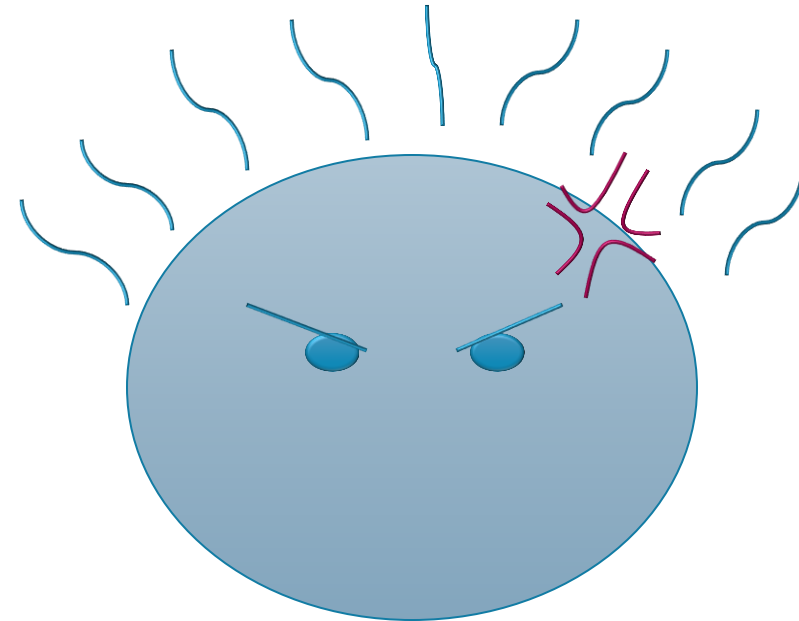
**100 nama ???**

**1000 nama ???**

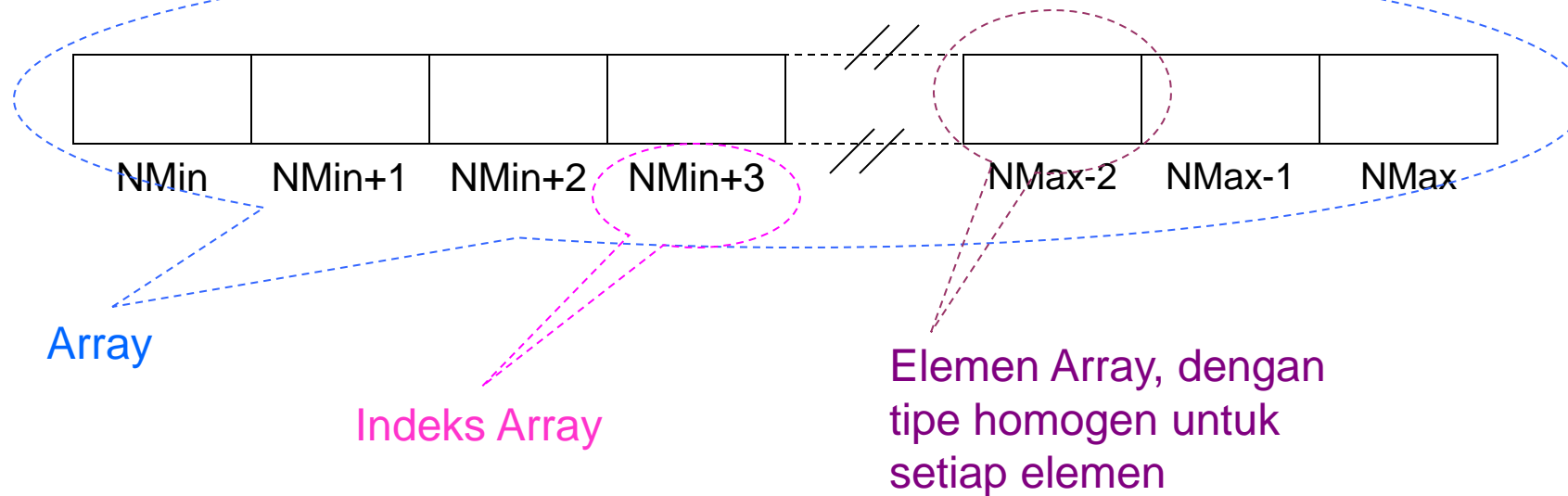
**10000 nama ???**

**1000000 nama ???**

....



# ARRAY / TABEL / VEKTOR / LARIK



01

**Array** mendefinisikan **sekumpulan** (satu atau lebih) elemen **bertipe sama**

02

Setiap elemen tersusun secara teratur (kontigu) dan dapat diakses dengan menggunakan **indeks**

03

Dalam Python, ada beberapa cara mendeklarasikan array → dalam kuliah ini, array didefinisikan menggunakan *collection type* **list**

# DEKLARASI ARRAY DALAM PYTHON (1)

## Cara-1:

Deklarasi variabel array sekaligus mendefinisikan isi array:

$$\langle \text{nama-var} \rangle = [\langle \text{val}_0 \rangle, \langle \text{val}_1 \rangle, \langle \text{val}_2 \rangle, \dots, \langle \text{val}_{n-1} \rangle]$$

- Deklarasi array dengan nama  $\langle \text{nama-var} \rangle$  dengan ukuran  $n$  dengan elemen  $\langle \text{val}_0 \rangle, \langle \text{val}_1 \rangle, \langle \text{val}_2 \rangle, \dots, \langle \text{val}_{n-1} \rangle$
- Type elemen tergantung pada nilai elemen yang diberikan
- Elemen terurut berdasarkan **indeks dari 0 s.d.  $n-1$** .

# DEKLARASI ARRAY DALAM PYTHON (2)

- Contoh-1:

**TabInt = [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]**

Array bernama **TabInt** di mana setiap elemen bertipe **integer**, dengan ukuran **10** elemen, dan alamat setiap elemen array (indeks) adalah dari **indeks ke-0 s.d. 9**

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



# DEKLARASI ARRAY DALAM PYTHON (3)

Jika belum diketahui nilai (*value*) apa yang akan diberikan pada array, maka dapat diberikan suatu nilai default seragam terlebih dahulu

- Contoh: Array berelemen integer: nilai elemen default = 0

**Cara-2:** Mendeklarasikan array dan mengisi dengan nilai default:

```
<nama-var> = [<default-val> for i in range (<n>)]
```

- Deklarasi array dengan nama <nama-var> berukuran <n> di mana nilai setiap elemen <default-val>. *i* adalah variabel untuk loop pengisian nilai default ke tiap elemen array.
- Type elemen tergantung pada type nilai <default-val>
- Elemen terurut berdasarkan **indeks dari 0 s.d. n-1**.

# DEKLARASI ARRAY DALAM PYTHON (4)

- Contoh-2: Array of integer

**TabInt = [0 for i in range(10)]**

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

Array bernama **TabInt** dengan setiap elemen bertipe **integer** dan dengan nilai default elemen **0**, dengan ukuran **10** elemen dan setiap elemen array diakses dengan menggunakan **indeks ke-0 s.d. 9**

# DEKLARASI ARRAY DALAM PYTHON (5)

- Contoh-3: Array of character

```
TabChar = ['*' for i in range(10)]
```

Array bernama **TabChar** dengan setiap elemen bertipe **char** dan dengan nilai default elemen **\***, dengan ukuran **10** elemen dan setiap elemen array diakses dengan menggunakan **indeks ke-0 s.d. 9**

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

# MENGAKSES ELEMEN ARRAY DALAM PYTHON

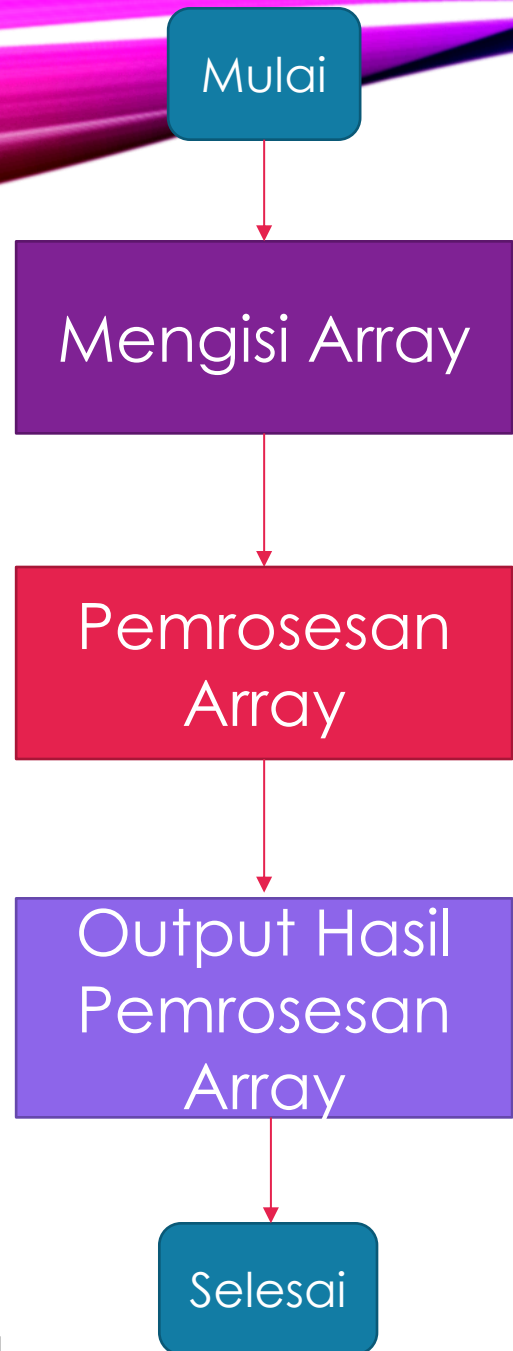


- Cara akses sebuah elemen: `<namatabel>[<indeks>]`
- Contoh: `TabInt = [1,2,4,-1,100,2,0,-1,3,9]`

1	2	4	-1	100	2	0	-1	3	9
0	1	2	3	4	5	6	7	8	9

```
print(TabInt[5])           # akan tercetak: 2
x = TabInt[1] + TabInt[6]  # x = 2 + 0 = 2
TabInt[9] = 9              # Elemen array indeks 9 menjadi 9
TabInt[10] ???             # Berada di luar range, tidak terdefinisi!!
```

- **Perhatian: Tidak boleh** mengakses elemen dengan **indeks berada di luar definisi**.
  - Pada contoh di atas, misalnya: `TabInt[10]`, `TabInt[-1]`, dll



# Pemrosesan Array



# PEMROSESAN SEKUENSIAL PADA ARRAY (1)

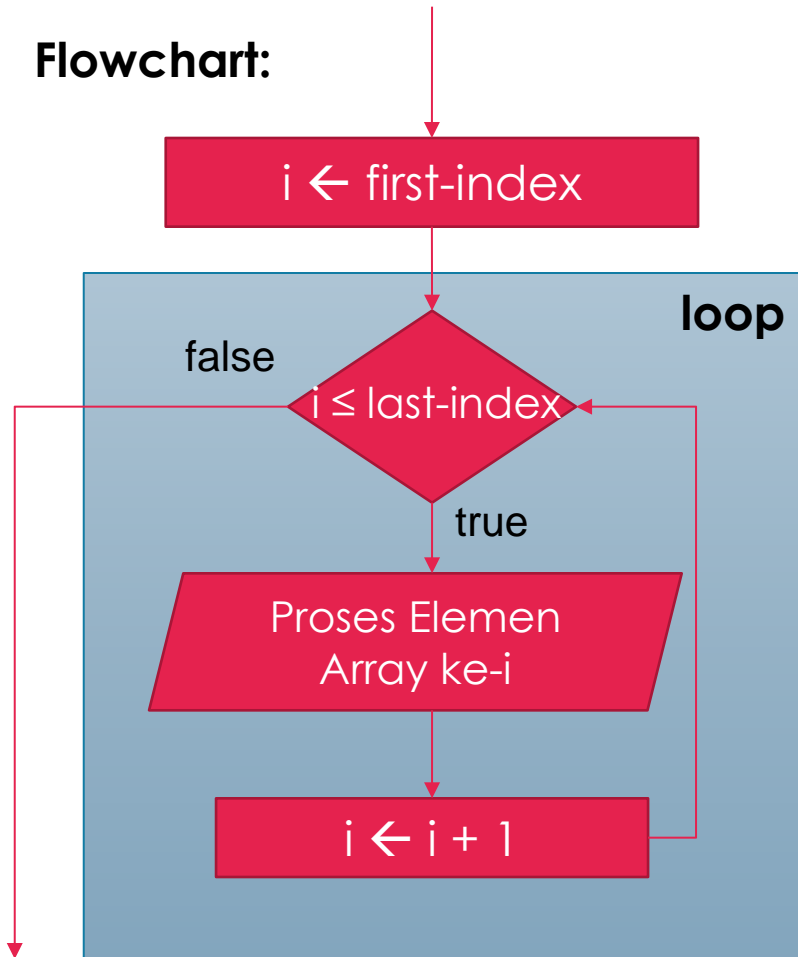


# PEMROSESAN SEKUENSIAL PADA ARRAY (2)

- Contoh-contoh persoalan pemrosesan sekuensial pada array:
  - Mengisi array secara sekuensial
  - Mencetak elemen array
  - Menghitung nilai rata-rata elemen array
  - Mengalikan elemen array dengan suatu nilai
  - Mencari nilai terbesar/terkecil pada array
  - Mencari indeks di mana suatu nilai ditemukan pertama kali di array
  - ...

# FLOWCHART + PSEUDOCODE UMUM PEMROSESAN SEKUENSIAL ARRAY

Flowchart:



Pseudocode:

```
i traversal [first-index..last-index]
{ Proses elemen array ke-i }
...
```

# MENGISI ARRAY (1)

- Buatlah program yang **mendeklarasikan** sebuah **array of integer** (array dengan elemen bertipe integer) sebesar **10** buah dan mengisinya dengan nilai yang dibaca dari *keyboard*.
- **Hati-hati** untuk tidak mengakses elemen di luar batas indeks array!

```
# Program IsiArray
# Mengisi array dengan nilai dari
# pengguna

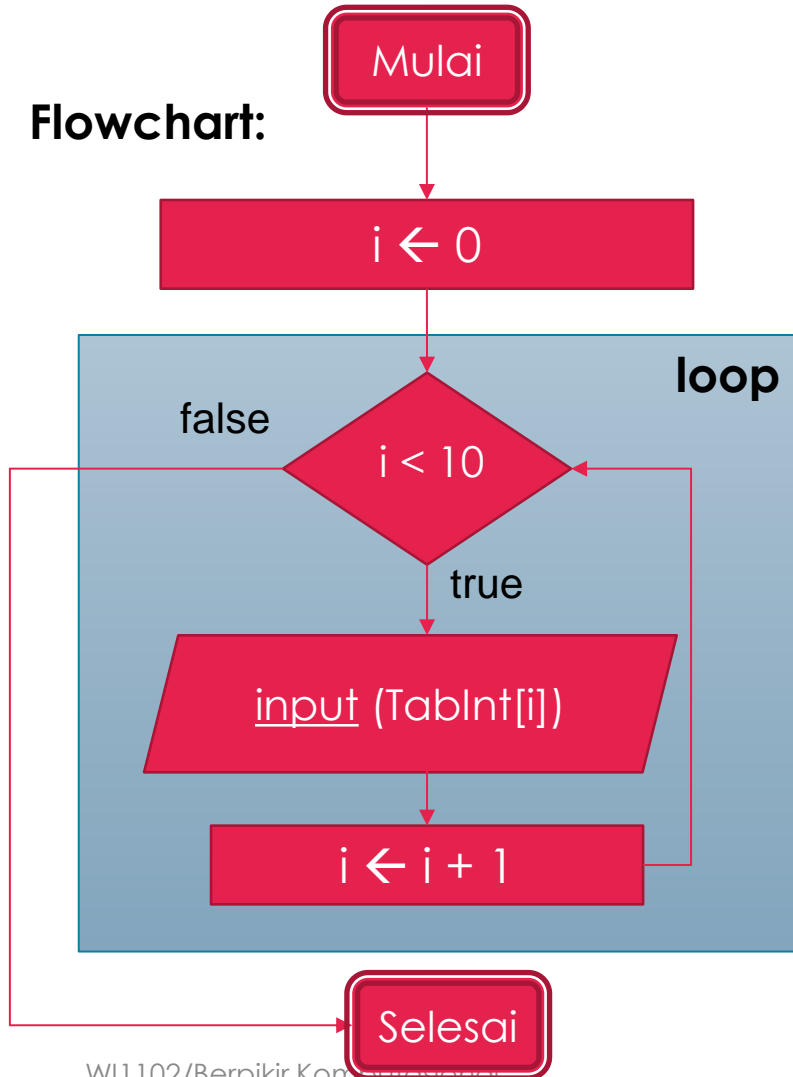
# KAMUS
# TabInt : array [0..9] of int
# i : int

# ALGORITMA
# Deklarasi array TabInt dan
# mengisinya dengan nilai default 0
TabInt = [0 for i in range(10)]

# Mengisi array dari pembacaan nilai
# dari keyboard
for i in range(0,10):
    TabInt[i] = int(input())
```

# MENGISI ARRAY (2): FLOWCHART + PSEUDOCODE

Flowchart:



Pseudocode:

```
i traversal [0..9]
  input(TabInt[i])
```



# MENULISKAN ISI ARRAY (1)

- Buatlah program yang:
- **mendeklarasikan** sebuah **array of integer** (array dengan elemen bertipe integer) sebesar **10** buah
  - **mengisinya** dengan nilai yang dibaca dari keyboard
  - **menuliskan** kembali apa yang disimpan dalam array ke layar
- **Hati-hati** untuk tidak mengakses elemen di luar batas indeks array!

```
# Program TulisArray
# Mengisi array dengan nilai dari
# pengguna dan menuliskan isinya ke
# layar

# KAMUS
# TabInt : array [0..9] of int
# i : int

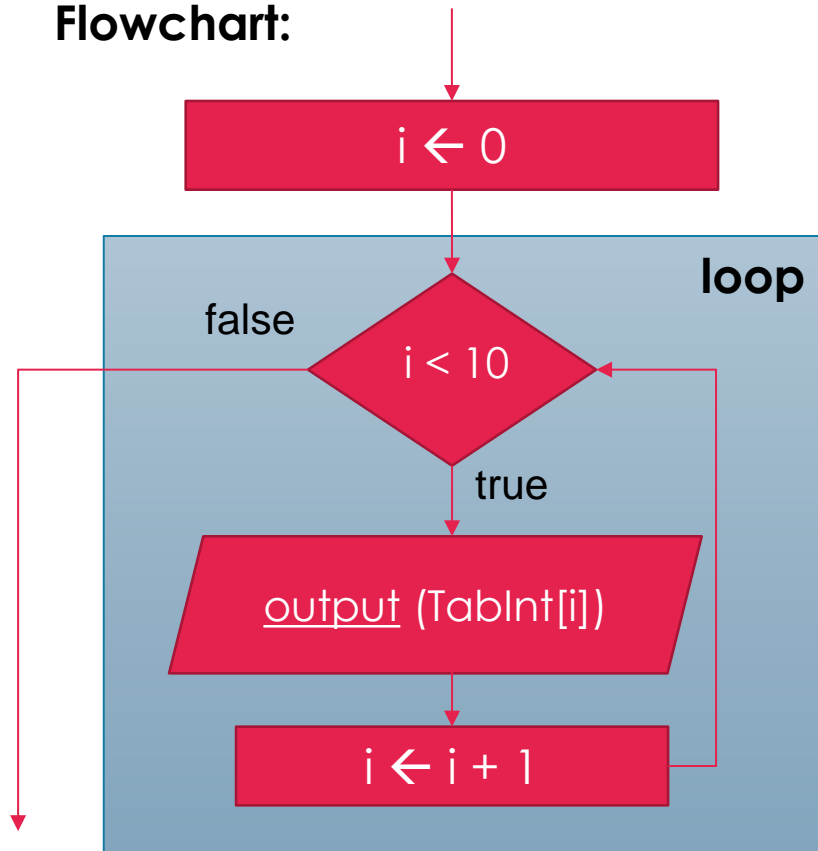
# ALGORITMA
# Deklarasi array TabInt dan
# mengisinya dengan nilai default 0
TabInt = [0 for i in range(10)]

# Mengisi array dari pembacaan nilai
# dari keyboard
for i in range(0,10):
    TabInt[i] = int(input())

# Mencetak isi array
for i in range(0,10):
    print(TabInt[i])
```

# MENULISKAN ISI ARRAY (2)

Flowchart:



Pseudocode:

```
... { Bagian mengisi array }  
i traversal [0..9]  
  output(TabInt[i])
```

Bagian mengisi array buat sendiri sebagai latihan

# MENGHITUNG RATA-RATA (1

- Buatlah program untuk menghitung rata-rata nilai elemen suatu array.
- Tahapan:
  - Deklarasikan array, contoh array of integer ukuran 10
  - Isi elemen array
  - Jumlahkan semua elemen array
  - Bagi hasil penjumlahan elemen array dengan banyaknya elemen array dan tampilkan hasilnya

```
# Program AverageArray
# Menghitung nilai rata-rata elemen array

# KAMUS
# TabInt : array [0..9] of int
# i : int
# sum : int

# ALGORITMA
# Deklarasi array TabInt dan mengisinya dengan
# nilai default 0
TabInt = [0 for i in range(10)]

# Mengisi array dari pembacaan nilai dari keyboard
for i in range(0,10):
    TabInt[i] = int(input())

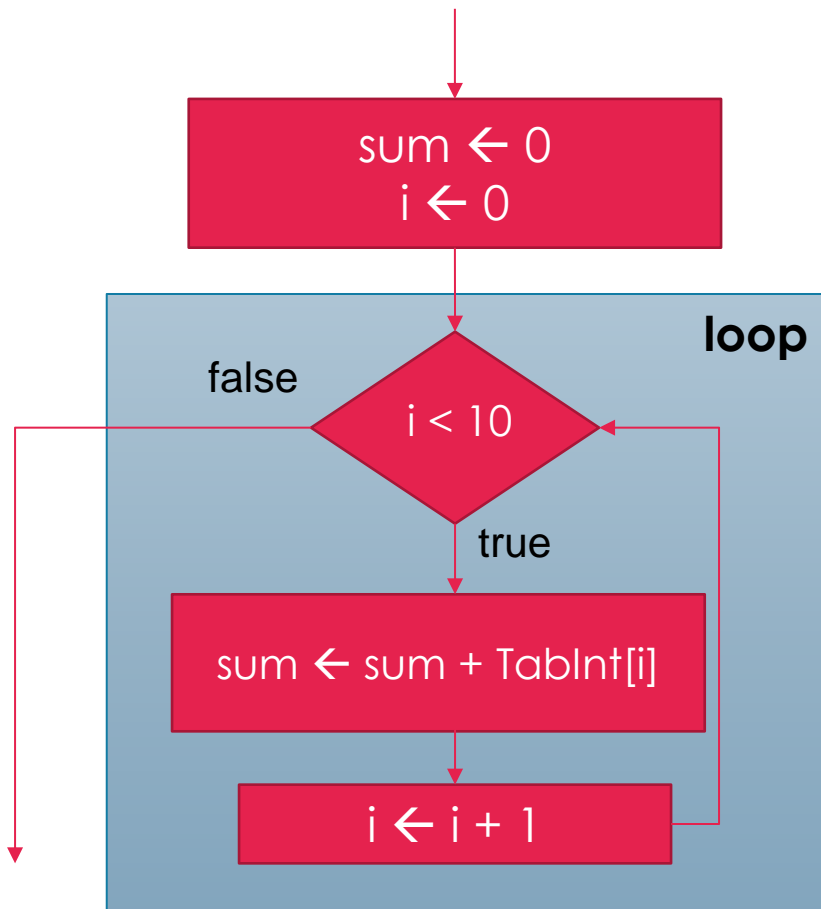
# Menjumlahkan elemen array
sum = 0
for i in range(0,10):
    sum = sum + TabInt[i]

# Menghitung nilai rata-rata dan menampilkannya
rata = sum/10
print ("Nilai rata-rata = " + str(rata))
```

# MENGHITUNG RATA-RATA (2)



Flowchart – Bagian penjumlahan elemen:



Pseudocode – Bagian penjumlahan elemen:

```
... { Bagian mengisi array }  
sum ← 0  
i traversal [0..9]  
    sum ← sum + TabInt[i]  
...
```

Bagian yang lain silakan dibuat sebagai latihan.

# LATIHAN

- Untuk setiap soal latihan, buatlah flowchart/pseudocode/kode program dalam Python



# LATIHAN-1

- Buatlah sebuah program yang berisi sebuah array dengan elemen integer berukuran 20, misalnya **T**
- Anggaplah sudah ada bagian program yang digunakan untuk mengisi array **T**
  - Lihat slide sebelumnya untuk pengisian array dari keyboard
- Program menerima masukan sebuah integer, misalnya **X**
- Selanjutnya, program mengalikan semua elemen array **T** dengan **X** dan mencetak semua elemen **T** yang baru ke layar.

Contoh:

$T = [4, 1, 3, 4, 5, 6, 8, 9, 12, 30, -1, 0, 4, -1, 3, 10, 14, 6, 7, 0]$

$X = 3$

Setelah elemen **T** dikalikan **X**

$T = [12, 3, 9, 12, 15, 18, 24, 27, 36, 90, -3, 0, 12, -3, 9, 30, 42, 18, 21, 0]$

# LATIHAN-2

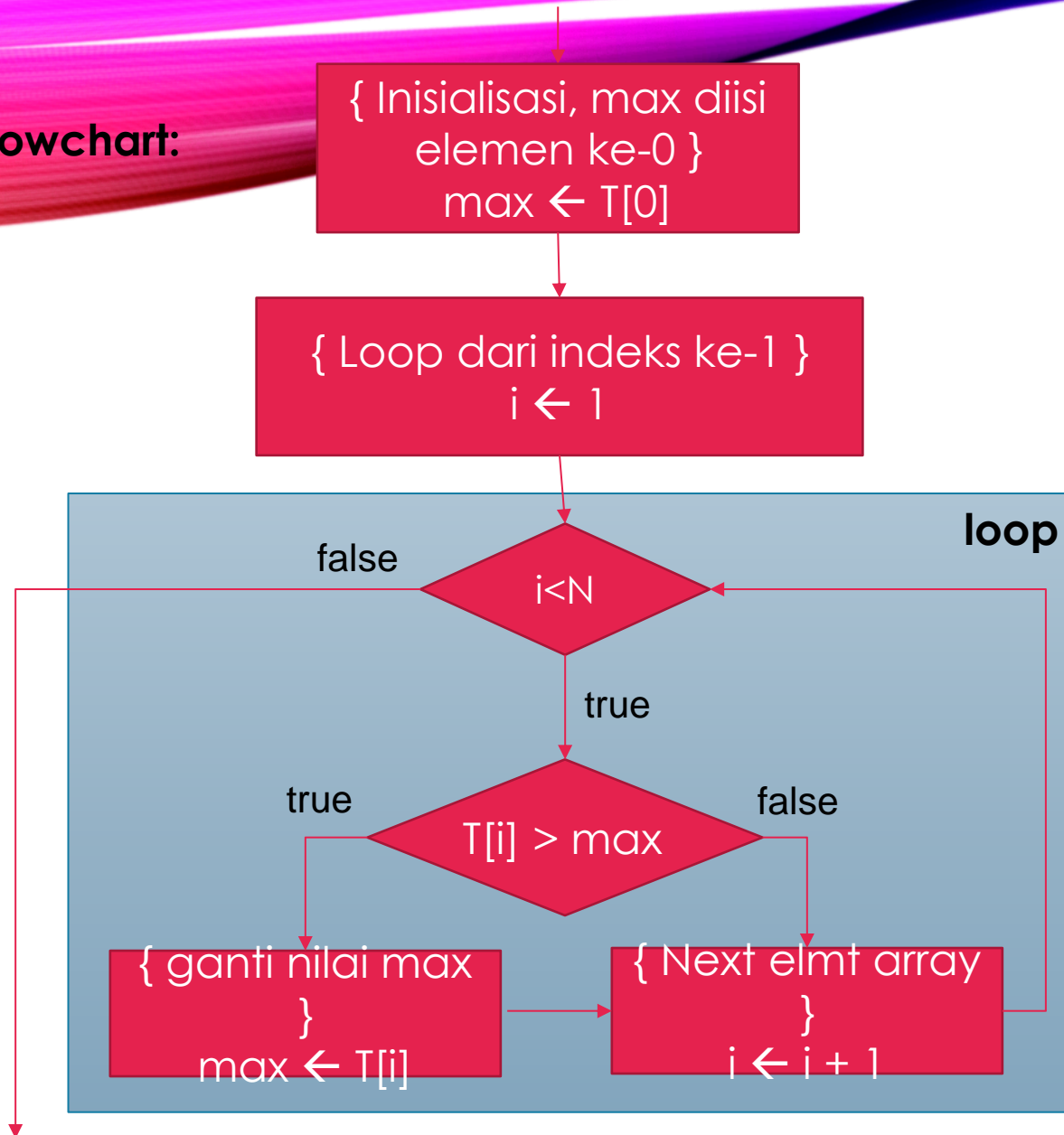
- Nilai mahasiswa dinyatakan dalam bentuk huruf, yaitu A, B, C, D, dan E.
  - Di ITB ada nilai AB dan BC, tapi untuk menyederhanakan persoalan kedua nilai tersebut diabaikan
- Sebuah program menerima data nilai 50 mahasiswa di sebuah kelas dalam bentuk indeks huruf seperti di atas dan disimpan dalam sebuah **array of character**.
- Tentukanlah **berapa banyak** mahasiswa yang lulus dan berapa yang tidak lulus. Mahasiswa dinyatakan lulus jika mendapatkan nilai A, B, atau C. Selebihnya, tidak lulus.

# PENCARIAN NILAI EKSTREM (1) (MINIMUM ATAU MAKSIMUM)

- Mencari nilai **terbesar** atau **terkecil** dari elemen suatu array
- Diketahui:
  - Sebuah array T dengan ukuran N elemen
- Buatlah program untuk menuliskan ke layar **nilai terbesar** dari elemen T
- Asumsi: T tidak kosong (minimum 1 elemen,  $N > 0$ )
- Contoh:
  - $N = 10$ ; T berisi: [9,12,30,-1,0,4,-1,3,30,14] maka nilai terbesar = 30
  - $N = 8$ ; T berisi: [1, 3, 5, 8, -12, 90, 3, 5] maka nilai terbesar = 90

# PENCARIAN NILAI EKSTREM (2)

Flowchart:



Pseudocode:

```

...
{ inisialisasi max dgn elemen ke-0 }
max ← T[0]

i traversal [1..N-1]
  { ganti kalau ketemu nilai T[i]>max }
  if (T[i]>max) then
    max ← T[i]
  ...
  
```

```
# Program MaxArray
# Mencari nilai terbesar pada array

# KAMUS
# N : int
# T : array [0..N-1] of int
# i : int
# max : int

# ALGORITMA
N = 10 # assign N dengan ukuran T
# Asumsi: pengisian array sudah dibuat
# Tetap harus dibuat untuk mengetes program

# Mencari nilai maksimum
max = T[0] # init max dgn elemen pertama

# Pencarian dimulai dari elemen ke-2
for i in range(1,N):
    # jika ada elemen > max, ganti nilai max
    if (T[i] > max):
        max = T[i]

# Cetak nilai terbesar
print ("Nilai terbesar = " + str(max))
```

# PENCARIAN NILAI EKSTREM (3) PROGRAM PYTHON



# LATIHAN-3

- Berdasarkan contoh soal sebelumnya, buatlah program untuk mencari nilai **terkecil** dari elemen T.

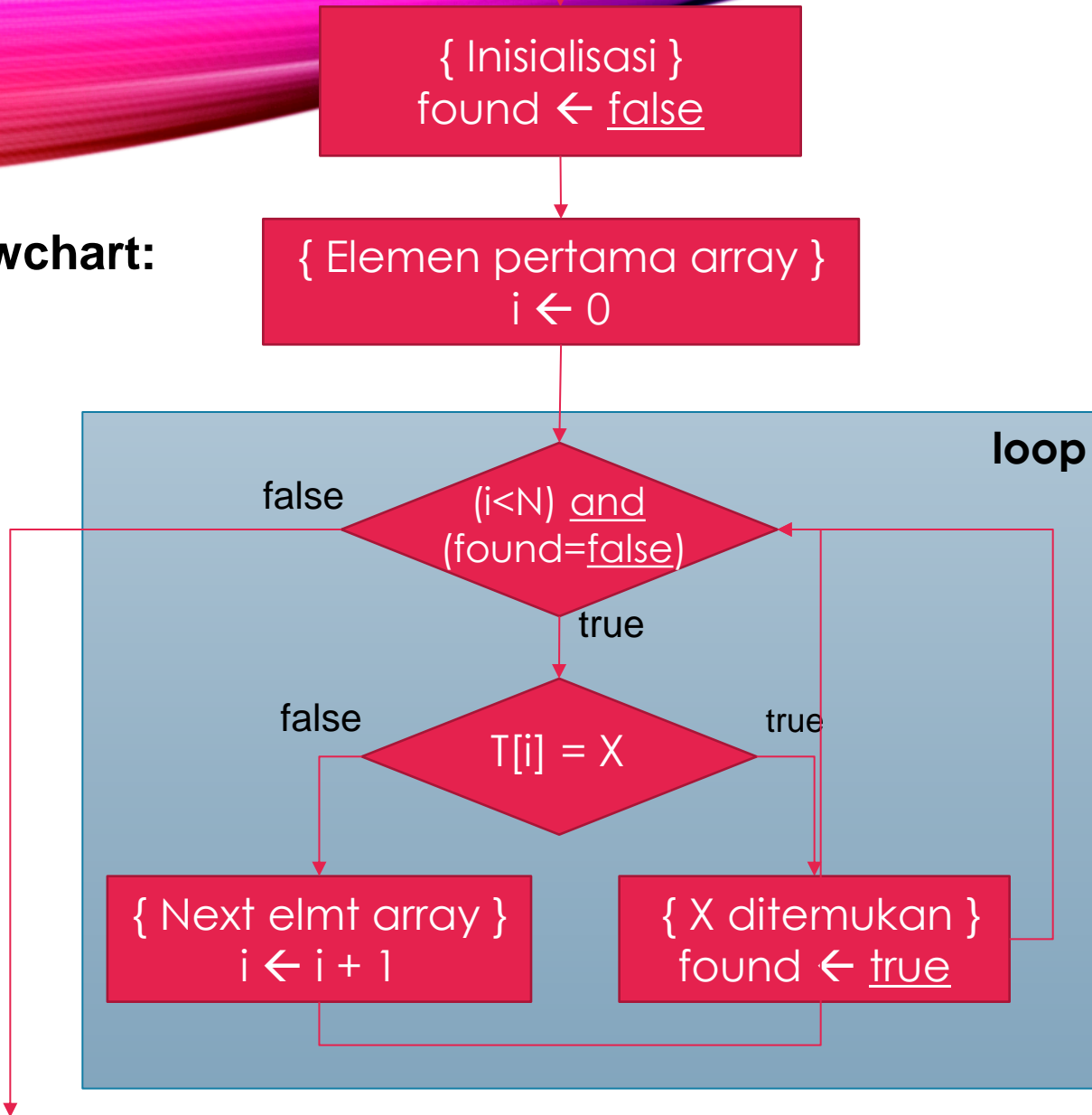
# SEARCHING (1)

## MENCARI INDEKS PERTAMA KEMUNCULAN NILAI

- **Searching** adalah proses yang penting dalam pemrosesan tabel karena sering dilakukan terhadap sekumpulan data yang disimpan dalam tabel
- Diketahui:
  - Sebuah array T dengan ukuran N elemen
  - Nilai X (bertipe sama dengan elemen T)
- Buatlah program untuk menuliskan ke layar **indeks pertama** di T di mana X ditemukan
- Asumsikan: Array tidak kosong (minimum 1 elemen,  $N > 0$ )
- Contoh:
  - $N = 10$ ; T berisi: [9,12,30,-1,0,4,-1,3,30,14];  $X = -1$  maka X ditemukan pertama kali di indeks 3
  - $N = 8$ ; T berisi: [1, 3, 5, 8, -12, 90, 3, 5];  $X = 0$  maka X tidak ditemukan di T

# SEARCHING (2)

## Flowchart:



## Pseudocode:

```

...
i ← 0
found ← false
while (i < N) and (found = false) do
    if (T[i] = X) then
        found ← true
    else { T[i] ≠ X }
        i ← i + 1
    { i ≥ N or found = true }
...

```

```
# Program SearchArray
# Mencari indeks di mana X ditemukan pertama kali di T
# KAMUS
# N : int; ukuran T
# T : array [0..N-1] of int
# i, X : int
# found : bool; menentukan X sdh ditemukan/belum

# ALGORITMA
# Asumsi: input array sudah dibuat; N terdefinisi

# Membaca nilai yang dicari, yaitu X
X = int(input())

# Pencarian dimulai dari elemen ke-2
i = 0
found = False # found = False; X belum ditemukan
while (i < N and found == False):
    if (T[i] == X):
        found = True # found = True; X sudah ditemukan
    else:
        i = i + 1 # hanya increment jika X belum
ditemukan
# i = N atau found = True

# Cetak Hasil
if (found == True): # X ditemukan di T
    print (str(X) + " ditemukan di indeks ke-" + str(i))
else: # found = False; X tidak ditemukan di T
    print (str(X) + " tidak ditemukan")
```

# SEARCHING (3) PROGRAM PYTHON

# LATIHAN-4

- Berdasarkan contoh sebelumnya, buatlah program untuk mencari indeks **terakhir** di mana X ditemukan di T.
- Petunjuk: proses pencarian dilakukan “mundur”, yaitu dari indeks elemen terakhir ke elemen pertama.
  - Modifikasi apa yang harus dilakukan terhadap algoritma sebelumnya?

# LATIHAN-5

- Sebuah vektor  $v = (v_0, v_1, v_2, v_3, v_4)$  direpresentasikan sebagai suatu array of integer dengan 5 buah elemen.
- Diketahui dua buah vektor, masing-masing terdiri atas 5 elemen, misalnya  $V$  dan  $U$ .
- Tuliskan hasil penjumlahan kedua vektor.
- Penjumlahan dua vektor menghasilkan vektor lain,  $W$ , dengan elemen ke- $i$  adalah:  $W_i = V_i + U_i$

$$W = U + V = (v_0, v_1, v_2, v_3, v_4) + (u_0, u_1, u_2, u_3, u_4)$$

$$W = (v_0+u_0, v_1+u_1, v_2+u_2, v_3+u_3, v_4+u_4)$$