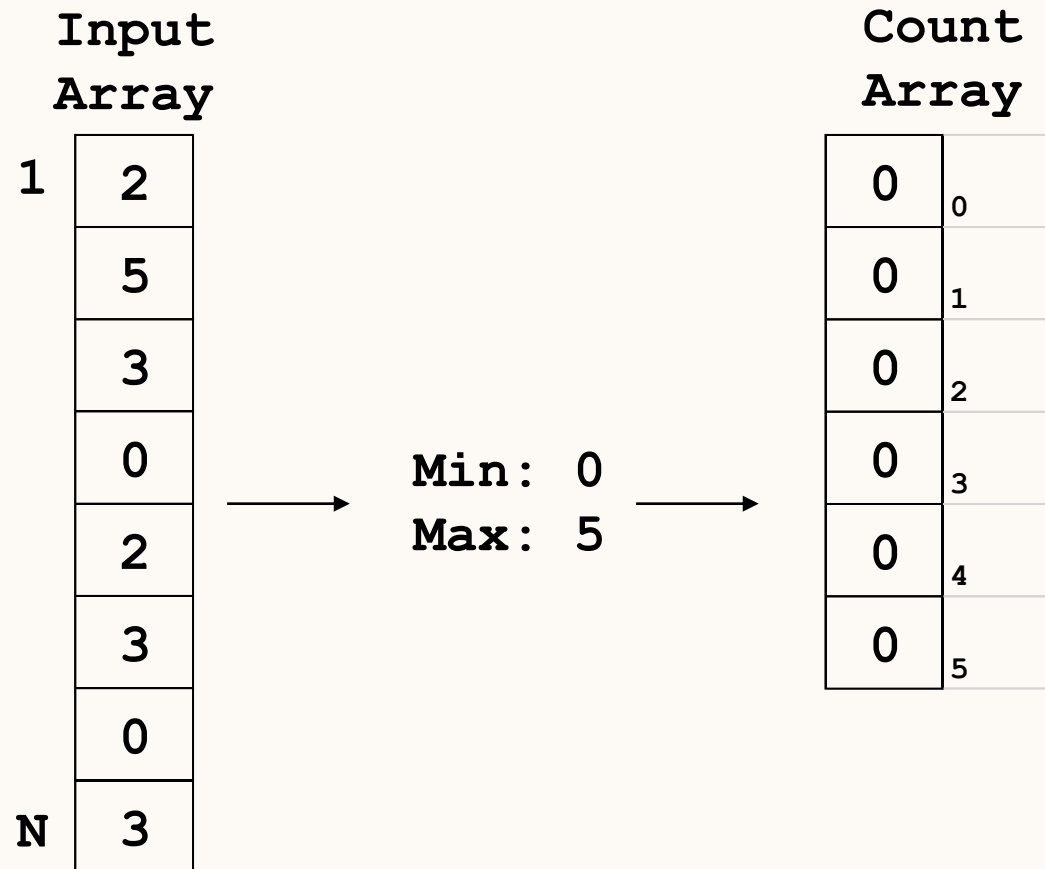


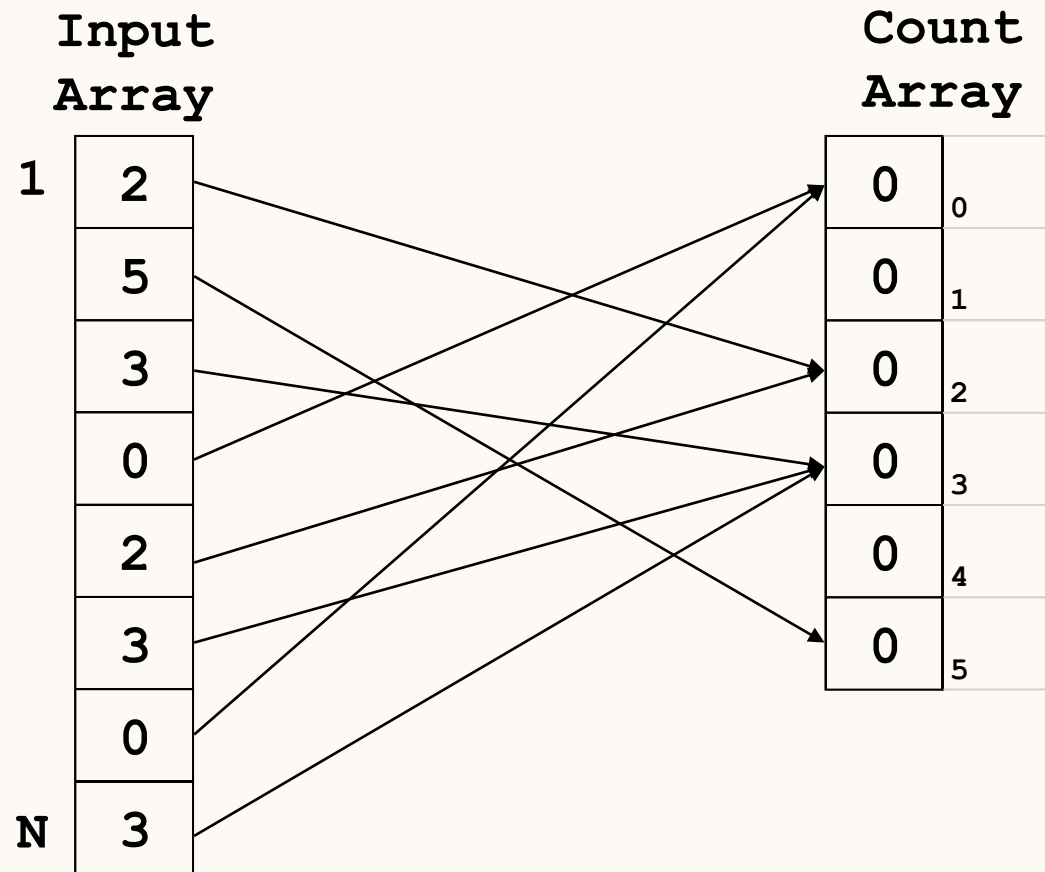
# **PERTEMUAN 9 PENGURUTAN I**

**PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN**

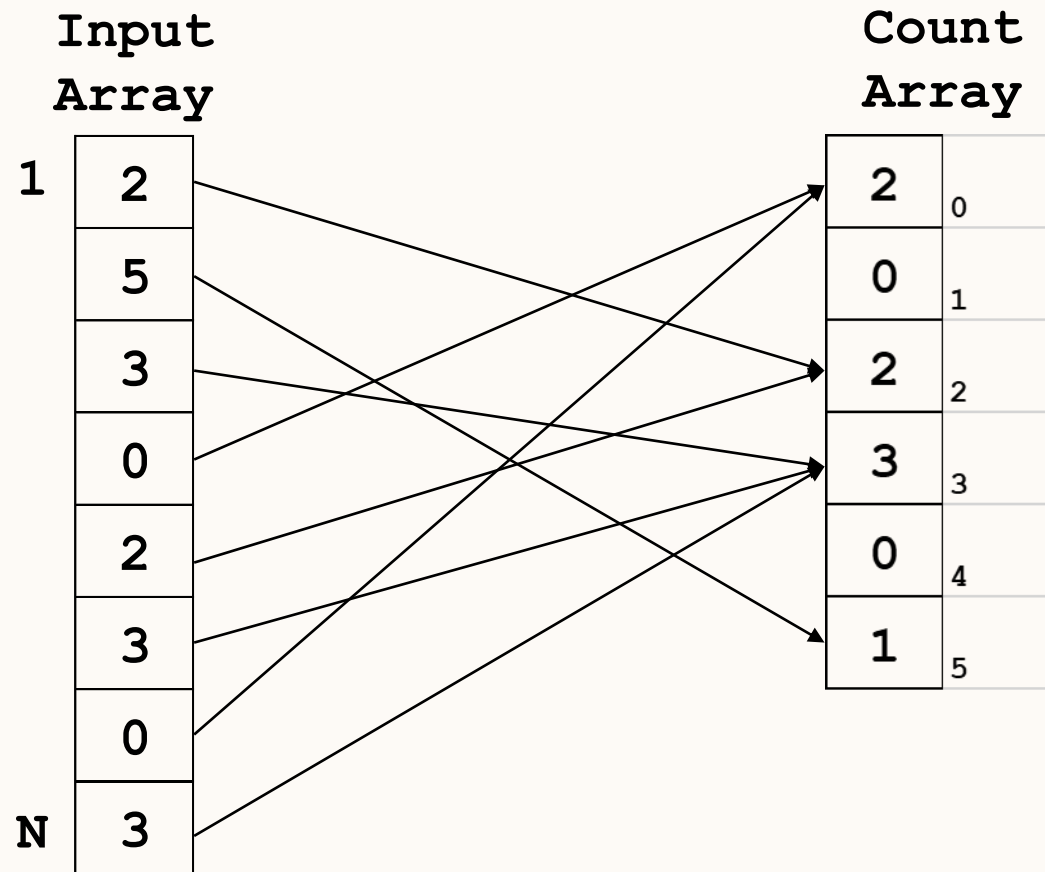
# PENGURUTAN DENGAN PENCACAHAN



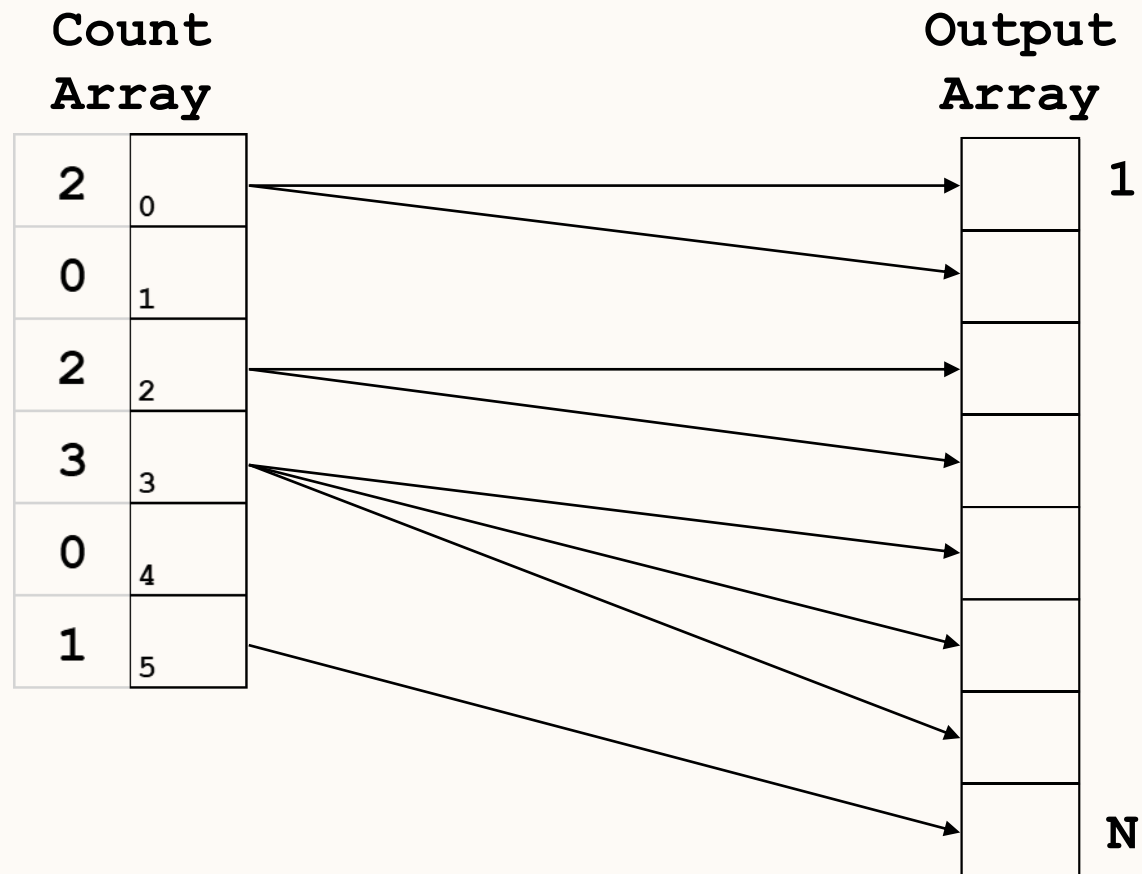
# PENGURUTAN DENGAN PENCACAHAN



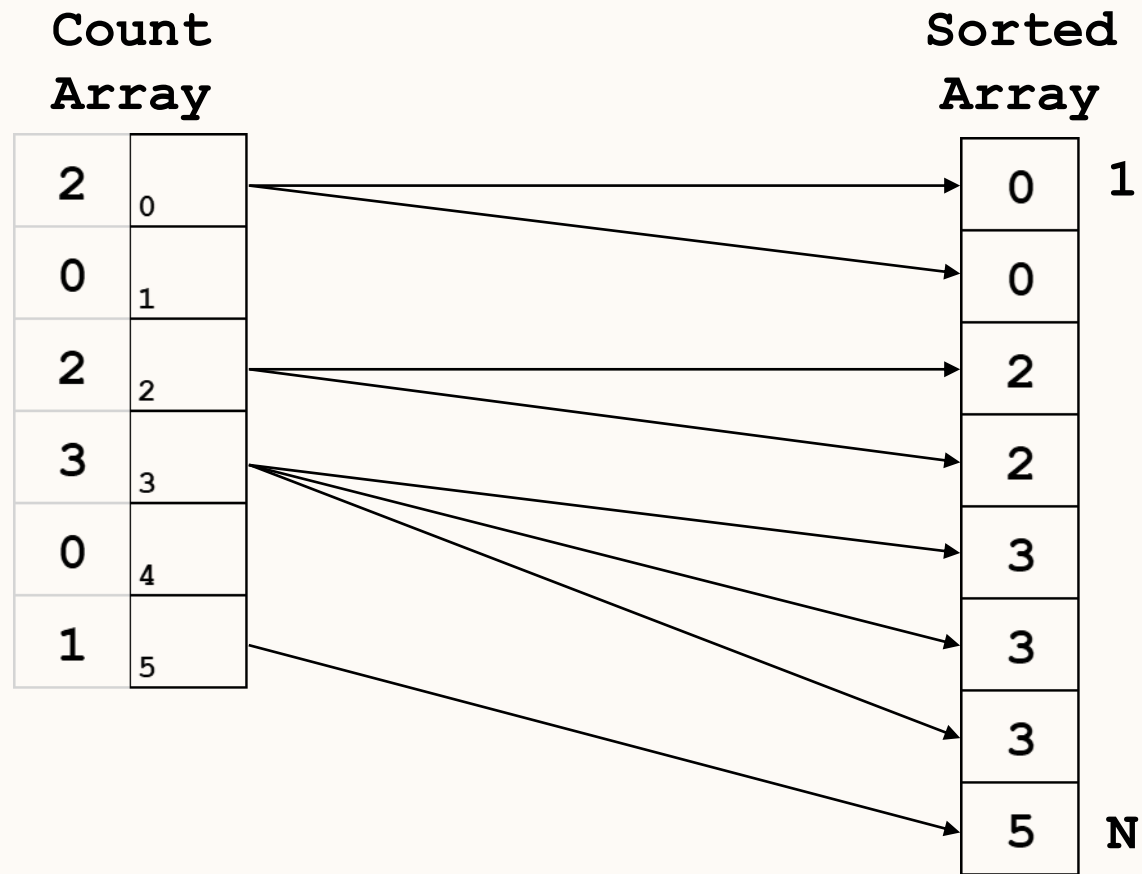
# PENGURUTAN DENGAN PENCACAHAN



# PENGURUTAN DENGAN PENCACAHAN



# PENGURUTAN DENGAN PENCACAHAN



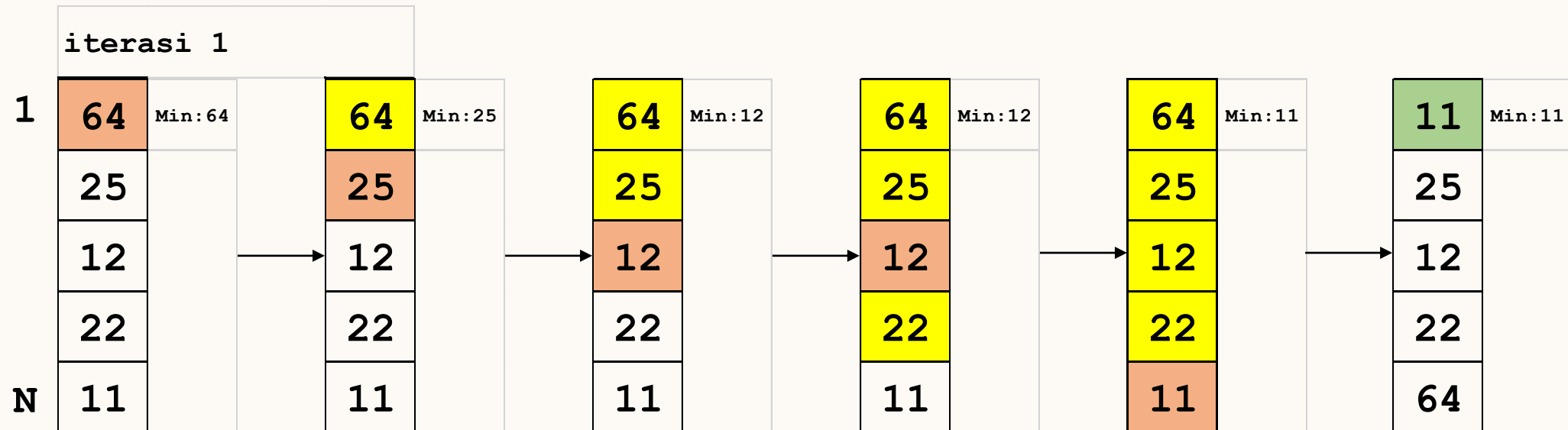
# PENGURUTAN DENGAN PENCACAHAN

<pre><b>procedure</b> CountSORT (<u>input/output</u> T : TabInt, <u>input</u> N : <u>integer</u>) { Mengurut tabel integer [1..N] dengan pencacahan }</pre>
<pre><b>KAMUS LOKAL</b> { ValMin dan ValMax adalah batas minimum dan Maximum harga yg tersimpan dalam T, harus diketahui }   TabCount : <u>array</u> [ValMin..Valmax] <u>of integer</u> [0..NMax]   i : <u>integer</u> { indeks untuk traversal tabel }   K : <u>integer</u> { jumlah elemen T yang sudah diisi pada proses pembentukan                kembali }</pre>
<pre><b>ALGORITMA</b> { Inisialisasi TabCount } i <u>traversal</u> [ValMin..ValMax]   TabCount<sub>i</sub> ← 0 { Counting } i <u>traversal</u> [1..N]   TabCount<sub>T<sub>i</sub></sub> ← TabCount<sub>T<sub>i</sub></sub> + 1 { Pengisian kembali : <math>T_1 \leq T_2 \dots \leq T_N</math> }  K ← 0 i <u>traversal</u> [ValMin..ValMax]   <u>if</u> (TabCount<sub>i</sub> ≠ 0) <u>then</u>     <u>repeat</u> TabCount<sub>i</sub> <u>times</u>       K ← K + 1       T<sub>K</sub> ← i</pre>

**Catatan :**

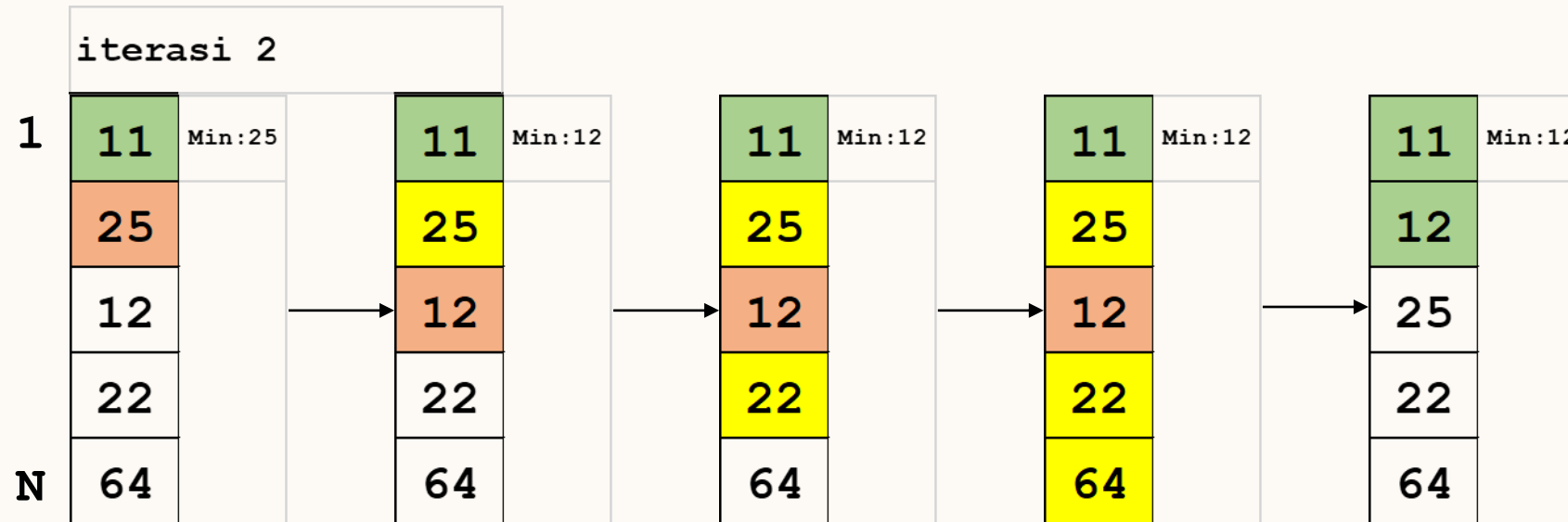
TabCount<sub>T<sub>i</sub></sub> dituliskan untuk menunjukkan bahwa indeks T adalah i, dan T<sub>i</sub> merupakan indeks dari TabCount.

# PENGURUTAN BERDASARKAN SELEKSI

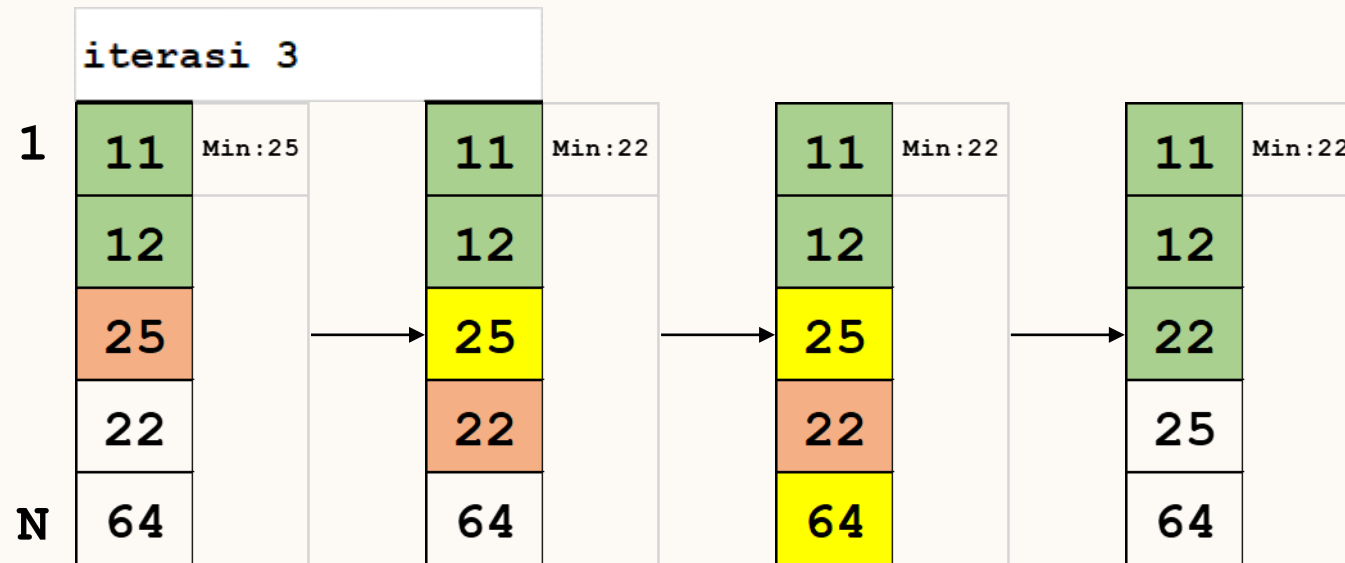




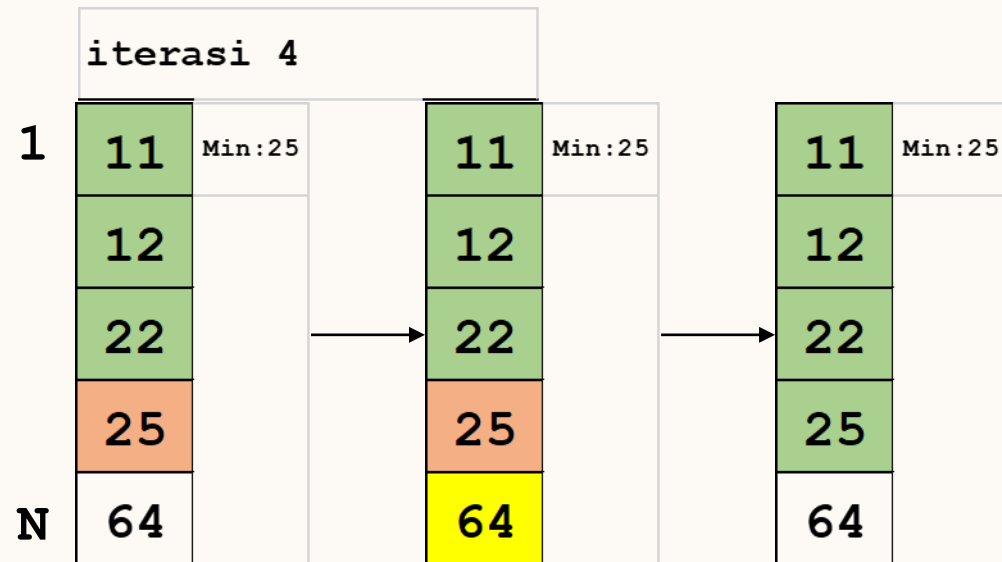
# PENGURUTAN BERDASARKAN SELEKSI



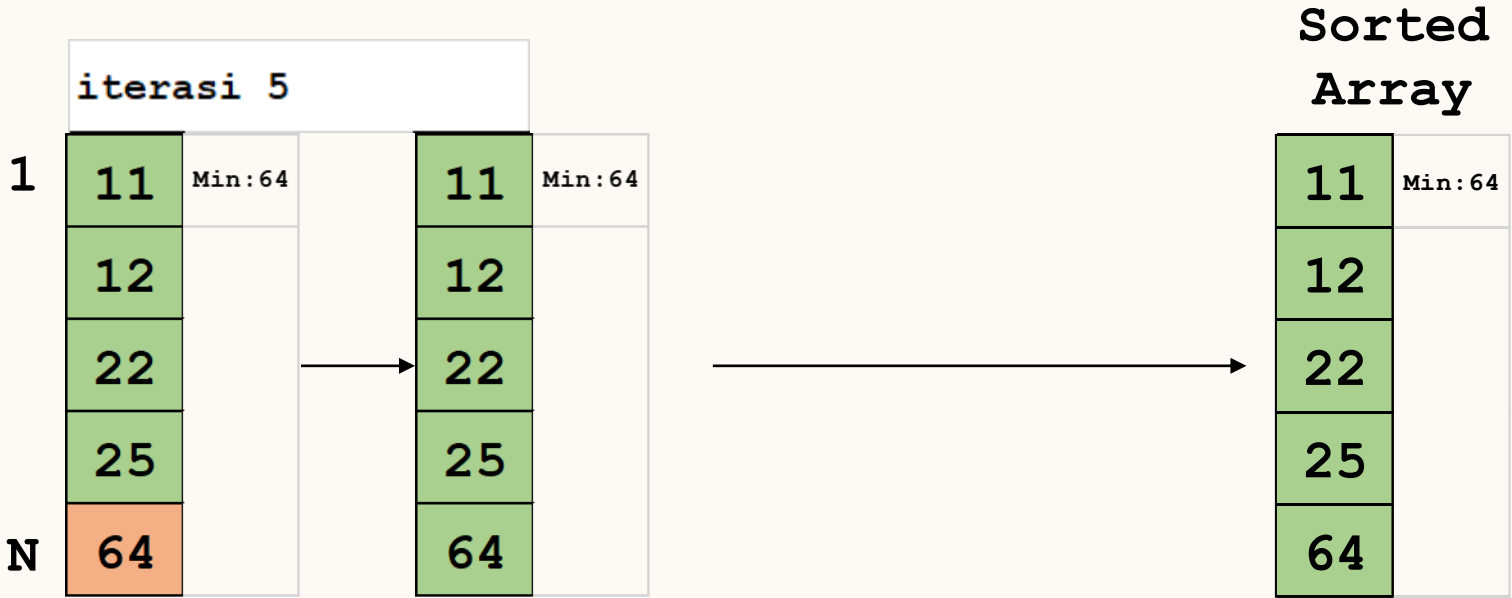
# PENGURUTAN BERDASARKAN SELEKSI



# PENGURUTAN BERDASARKAN SELEKSI



# PENGURUTAN BERDASARKAN SELEKSI



## PENGURUTAN BERDASARKAN SELEKSI (DESCENDING)

```
procedure MAXSORT (input/output T : TabInt, input N : integer)
{ Mengurut tabel integer [1..N] terurut mengecil dengan maksimum suksesif }
```

### KAMUS LOKAL

```
  i : integer      { indeks untuk traversal tabel }
  Pass : integer   { tahapan pengurutan }
  Temp : integer   { memorisasi harga untuk penukaran }
  IMax : integer   { indeks, di mana T [1..pass] bernilai maksimum }
```

### ALGORITMA

```
  Pass traversal [1..N-1]
  { Tentukan Maximum [Pass..N] }
  IMax ← pass
  i traversal [pass+1..N]
    if (TIMax < Ti ) then
      IMax ← i
  { TIMax adalah maximum T[pass..N] }
  { Tukar TIMax dengan TPass }
  Temp ← TIMax
  TIMax ← TPass
  TPass ← Temp
  { T [1..Pass] terurut: T1 ≥ T2 ≥ T3 ≥ ... ≥ TPass }
  { Seluruh tabel terurut, T1 ≥ T2 ≥ T3 ..... ≥ TN }
```

# CHALLENGE

1. Implementasikan algoritma Count Sort dan Selection Sort Descending ke dalam bahasa C
2. Modifikasi algoritma Selection Sort Descending sehingga dapat mengurutkan array secara menaik (Ascending)

# TUGAS

`T = [7, 1, a, 5, 3, b, 6]`

`N = 7`

`a = digit terakhir nim`

`b = digit kedua terakhir nim`

1. Urutkan tabel T dengan menggunakan algoritma Count Sort!
2. Urutkan tabel T dengan menggunakan algoritma Selection Sort secara descending!



# **THANK YOU**

Tim asprak alpro semoga diberi kesehatan  
mental