



# Pseudocode

Logic Algorithm

Yunia Ikawati



# Objectives:

1. Mahasiswa mampu memahami pentingnya pseudocode sebagai alat bantu untuk mengubah algoritma menjadi kode program.
2. Mahasiswa mampu mengubah algoritma dan flowchart menjadi pseudocode



# Outline:

1. Dasar-dasar Pseudocode
2. Latihan



# Fundamental Pseudocode

# Let's Throwback to Algorithm

Algoritma adalah cara menjelaskan suatu tugas yang terdiri dari beberapa langkah, terutama ketika kita ingin menjelaskan kepada pihak lain (manusia maupun mesin) bagaimana melaksanakan langkah-langkah tersebut secara tepat.

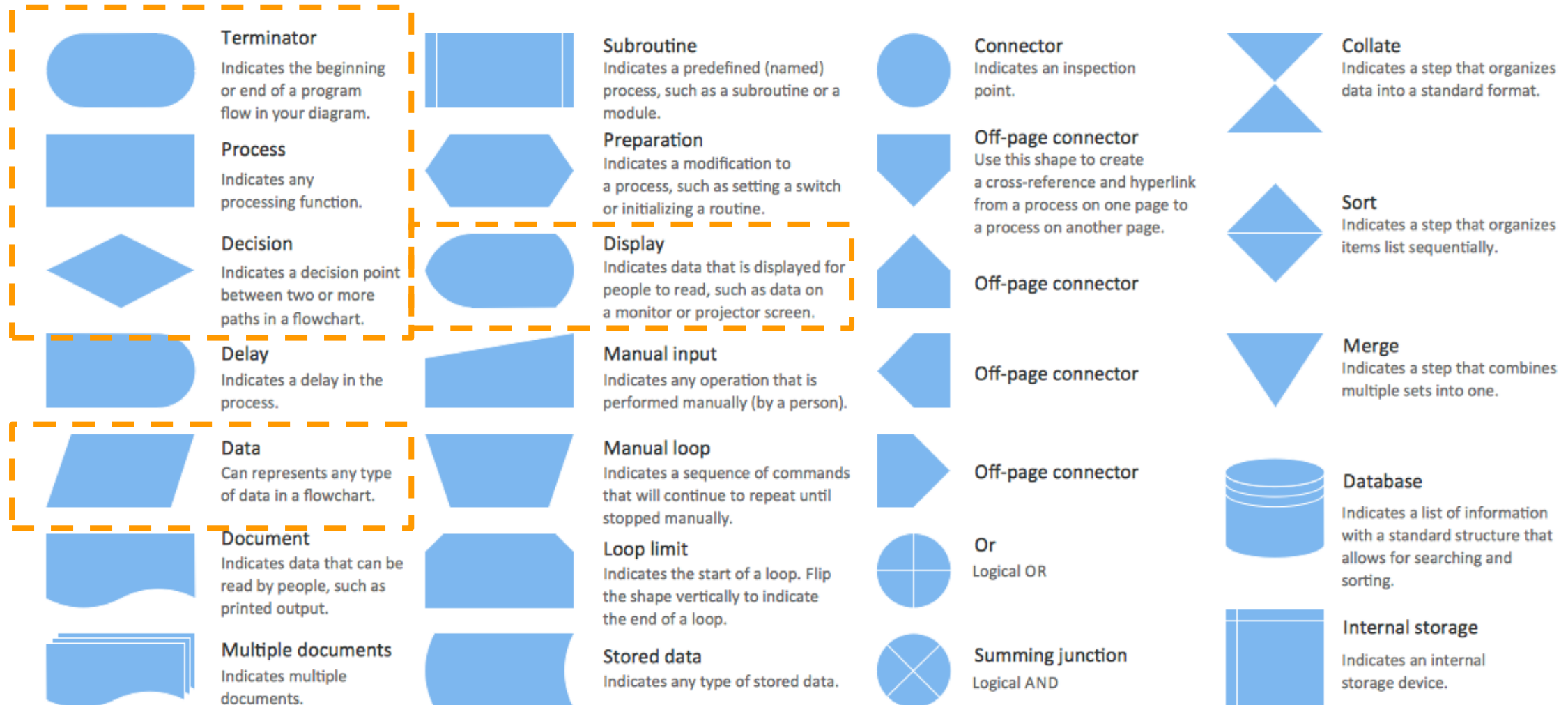
Karakteristik Algoritma:

1. Berhingga (Finiteness)
2. Jelas/Tidak ambigu (Definiteness)
3. Input
4. Output
5. Efektif (Effectiveness)

# Let's Throwback to Flowchart

- Flowchart merupakan representasi grafis dari algoritma.
- Flowchart lebih intuitif dan memudahkan pemahaman algoritma.
- Simbol dan bentuk standar digunakan untuk menunjukkan urutan langkah-langkah algoritma.

# Let's Throwback to Flowchart



# Algorithm vs Flowchart

Contoh 1: Menghitung Luas Persegi Panjang

Tujuan: Menghitung luas persegi panjang.

Algoritma:

1. Masukkan panjang persegi panjang.
2. Masukkan lebar persegi panjang.
3. Hitung luas = panjang  $\times$  lebar.
4. Tampilkan luas.

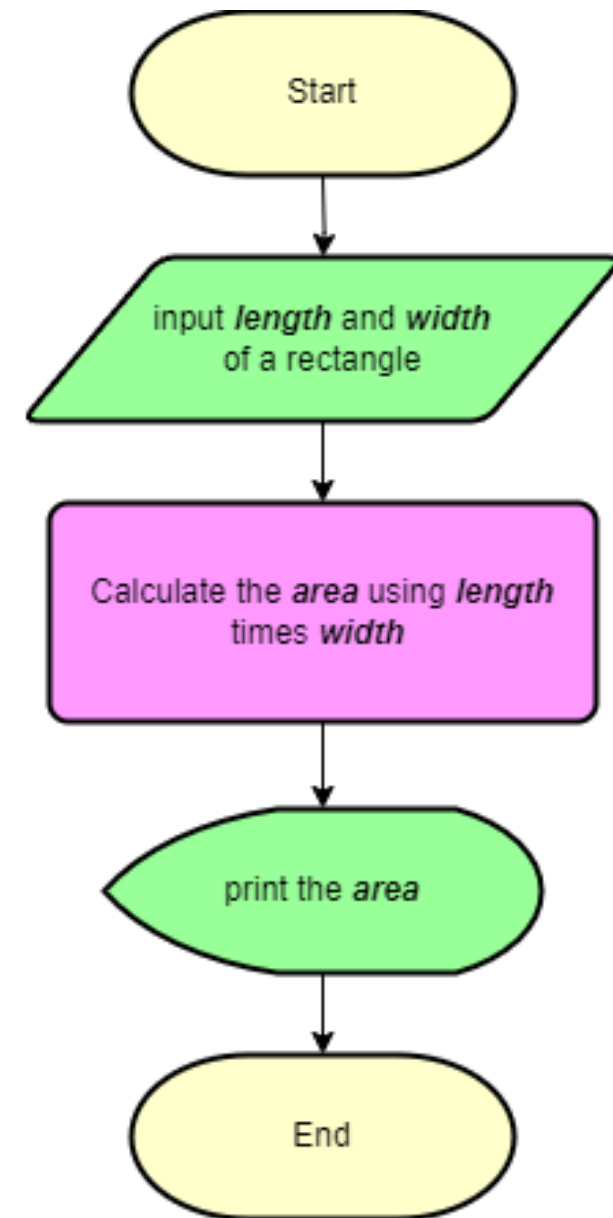
# Algorithm vs Flowchart

Contoh 1: Menghitung Luas Persegi Panjang

Tujuan: Menghitung luas persegi panjang.

Algoritma:

1. Masukkan panjang persegi panjang.
2. Masukkan lebar persegi panjang.
3. Hitung luas = panjang  $\times$  lebar.
4. Tampilkan luas.



# Algorithm vs Flowchart

Contoh 2: Menentukan Kelulusan Mata Kuliah Pemrograman 1

Tujuan: Menentukan apakah mahasiswa lulus mata kuliah Pemrograman.

Mahasiswa dinyatakan lulus jika nilai akhir  $\geq 56$ .

Algoritma:

1. Masukkan nilai akhir.
2. Periksa apakah nilai akhir  $\geq 56$ .
3. Jika ya, tampilkan "Lulus".
4. Jika tidak, tampilkan "Tidak Lulus".

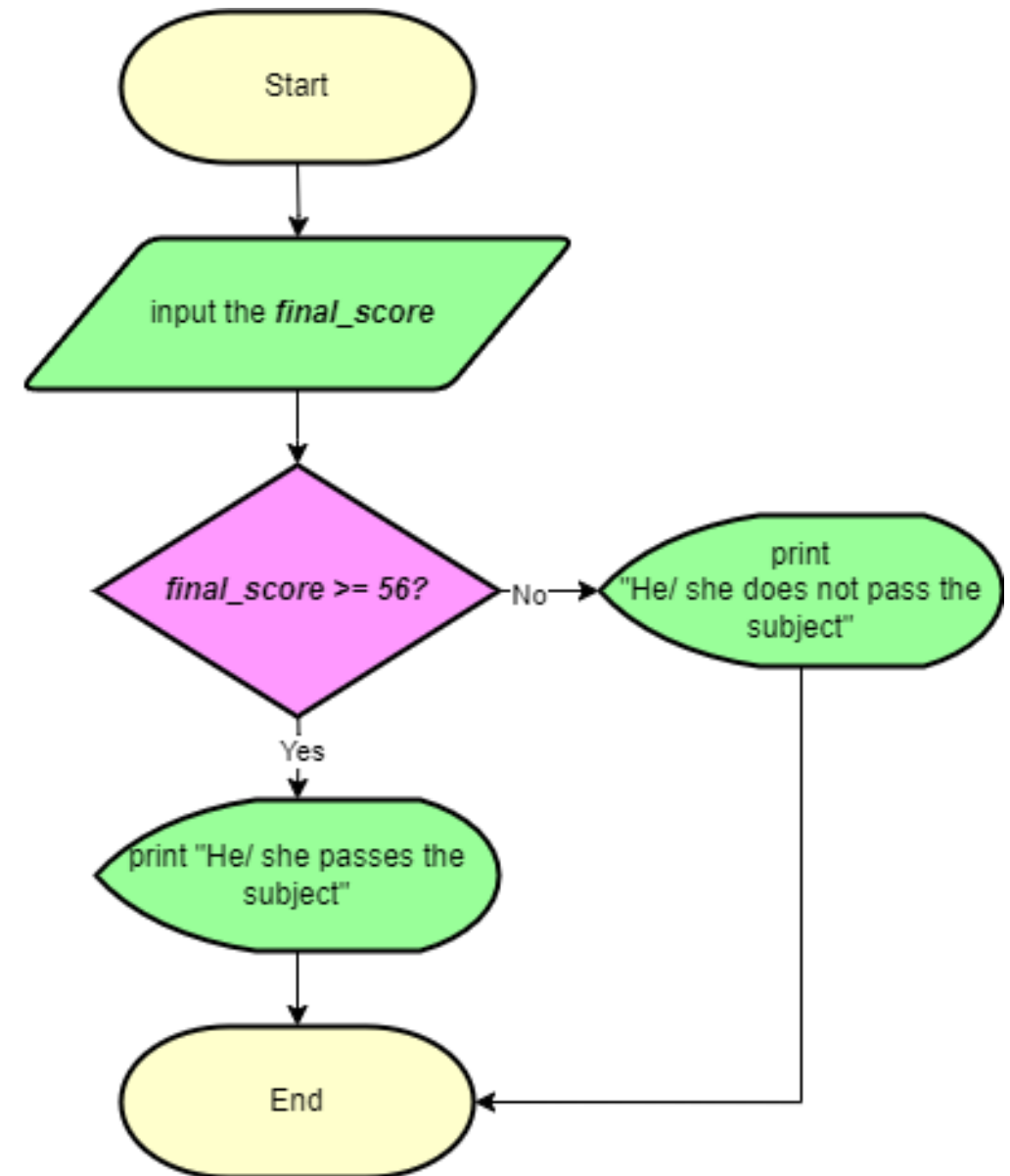
# Algorithm vs Flowchart

Contoh 2: Menentukan Kelulusan Mata Kuliah Pemrograman 1

Mahasiswa dinyatakan lulus jika nilai akhir  $\geq 56$ .

Algoritma:

1. Masukkan nilai akhir.
2. Periksa apakah nilai akhir  $\geq 56$ .
3. Jika ya, tampilkan "Lulus".
4. Jika tidak, tampilkan "Tidak Lulus".



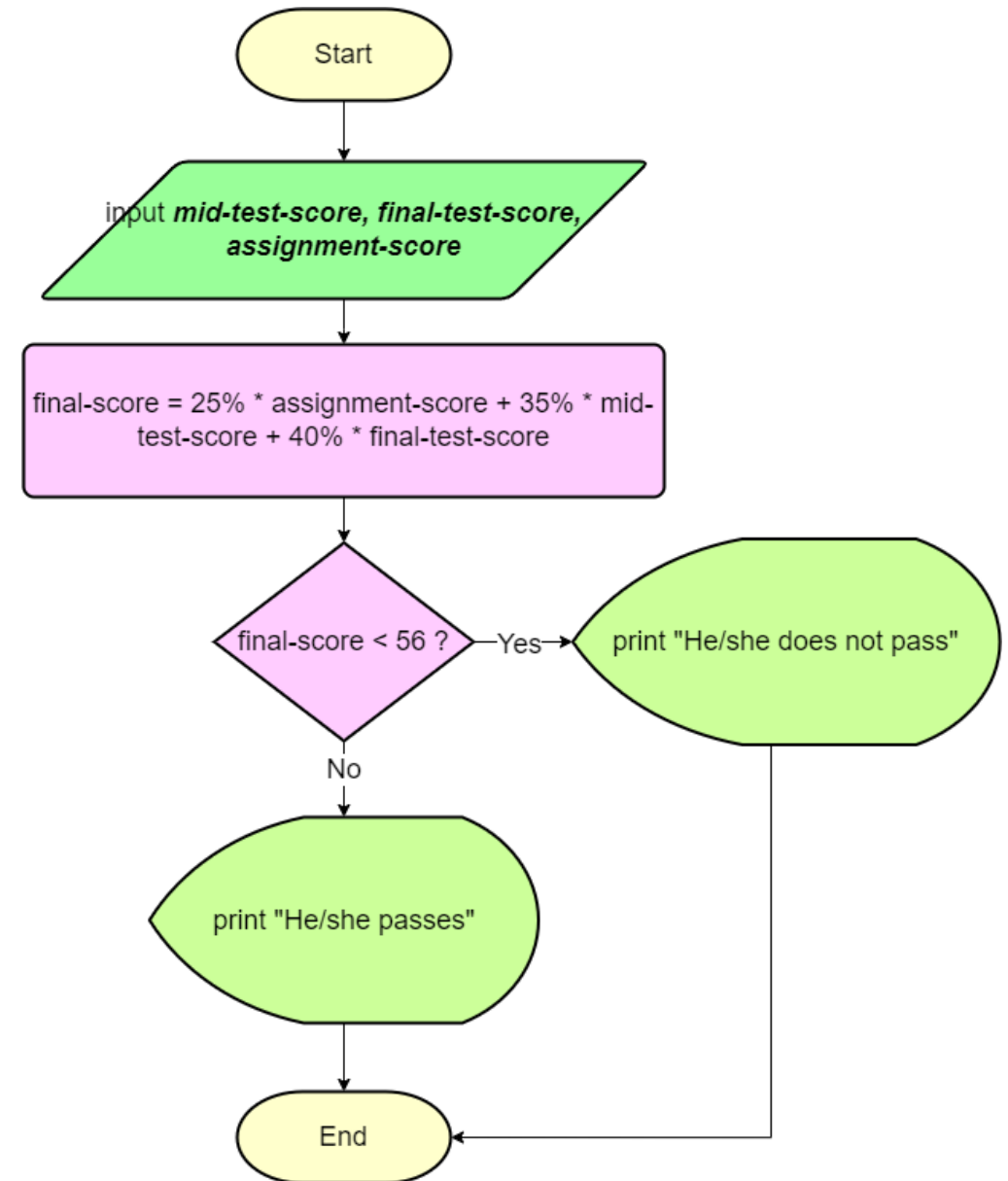
# Exercise

- Buat algoritma dan flowchart untuk menentukan kelulusan mahasiswa dengan ketentuan: Nilai Akhir = 25% Nilai Tugas 35% Nilai UTS 40% Nilai UAS
- Mahasiswa lulus jika Nilai Akhir  $\geq 56$ .

# Answer

## Algorithm:

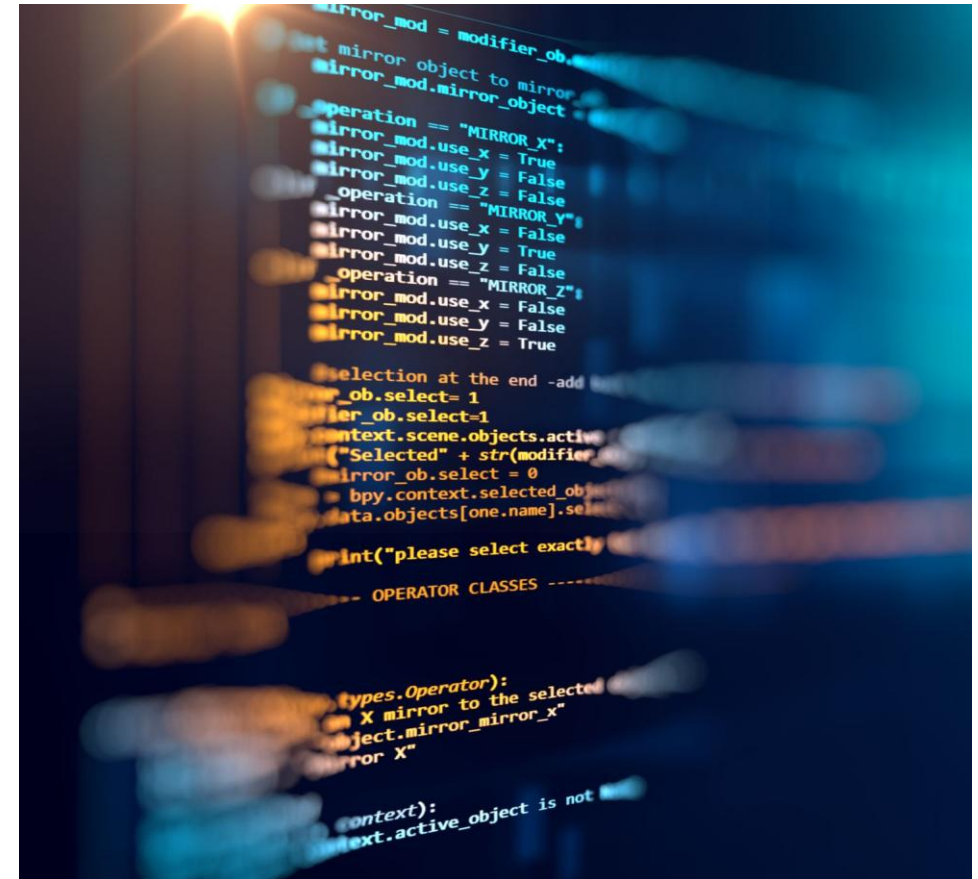
- Input nilai tugas, UTS, dan UAS.
- Hitung nilai akhir =  $(25\% \times \text{tugas}) + (35\% \times \text{UTS}) + (40\% \times \text{UAS})$ .
- Periksa apakah nilai akhir  $\geq 56$ .
- Jika ya, tampilkan "Lulus".
- Jika tidak, tampilkan "Tidak Lulus".



**Flowchart** membantu memvisualisasikan algoritma sehingga lebih mudah dipahami. Namun, flowchart tidak mudah langsung diterjemahkan menjadi kode program. Oleh karena itu, **pseudocode** digunakan sebagai alat bantu untuk mengubah algoritma atau flowchart menjadi kode program.

# What is Pseudocode?

- Pseudocode adalah kode semu yang menjelaskan cara menyelesaikan suatu masalah.
- Menggunakan bahasa yang menyerupai bahasa pemrograman.
- Lebih mudah dipahami dan lebih ringkas dibandingkan algoritma.
- Tidak memiliki aturan sintaks yang baku.
- Dapat diterapkan ke berbagai bahasa pemrograman.

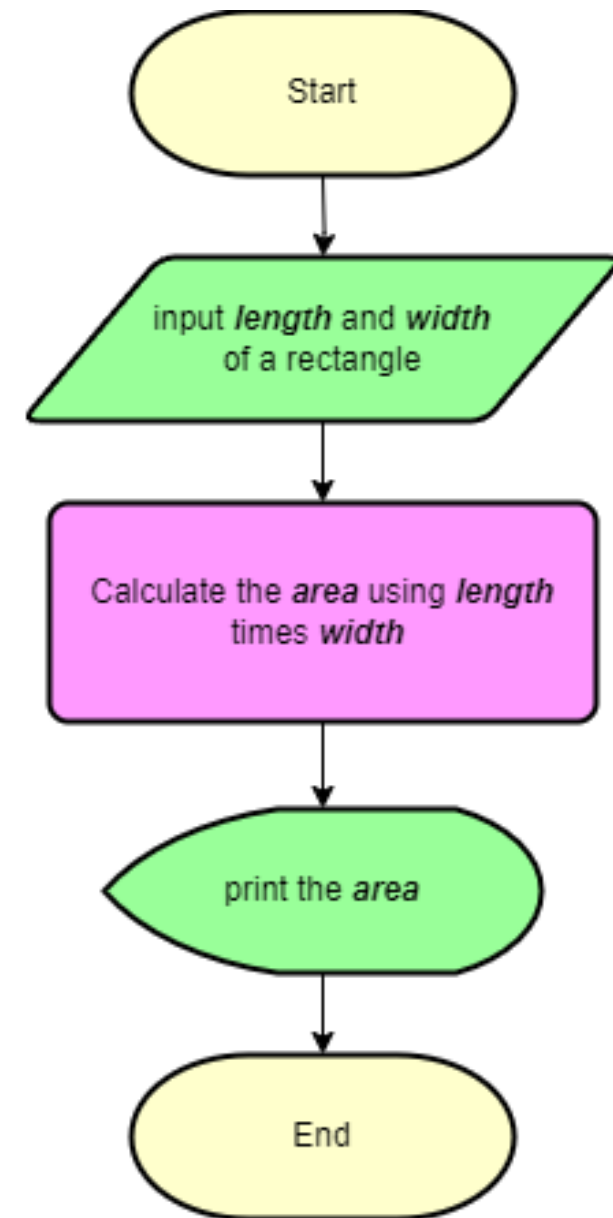


# Flowchart vs Pseudocode

Menghitung Luas Persegi Panjang:

## Pseudocode:

1. INPUT panjang
2. INPUT lebar
3.  $luas \leftarrow panjang * lebar$
4. CETAK luas



# Algorithm vs Pseudocode

Menghitung Luas Persegi Panjang:

## Algorithm:

- Masukkan panjang persegi panjang.
- Masukkan lebar persegi panjang.
- Hitung luas = panjang  $\times$  lebar.
- Tampilkan luas.

## Pseudocode:

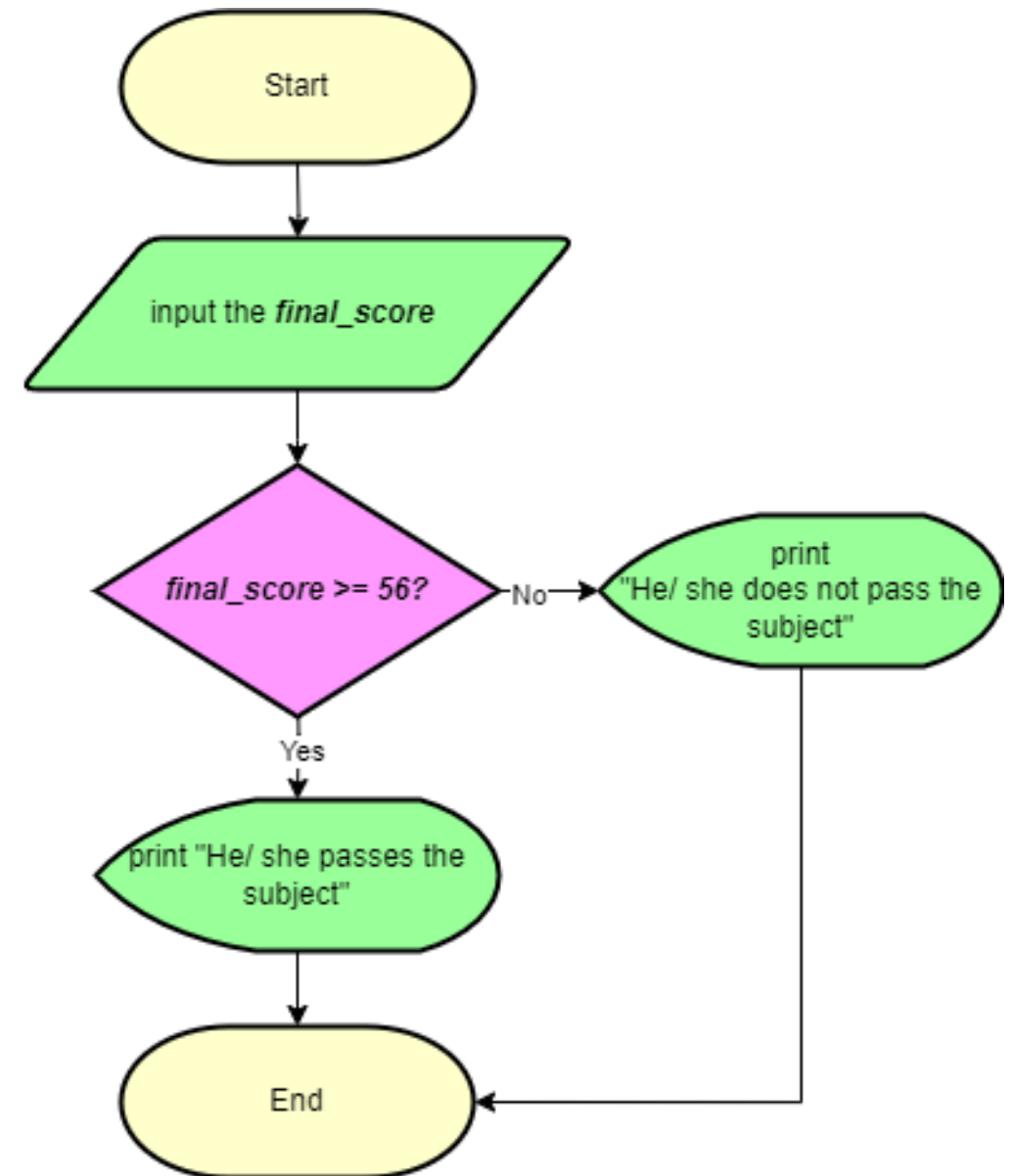
- INPUT panjang
- INPUT lebar
- luas  $\leftarrow$  panjang \* lebar
- CETAK luas

# Algorithm vs Flowchart

Menentukan Kelulusan Mahasiswa

## Pseudocode:

- INPUT nilai\_akhir
- IF nilai\_akhir  $\geq$  56 THEN
- CETAK "Mahasiswa Lulus"
- ELSE
- CETAK "Mahasiswa Tidak Lulus"
- ENDIF



# Algorithm vs Pseudocode

Objective: Check whether a student pass “Pemrograman 1” subject or not. He / She will pass if the final score is more than or equal to 56.

## Algorithm:

1. Masukkan nilai akhir.
2. Periksa apakah nilai akhir  $\geq 56$ .
3. Jika ya, tampilkan "Lulus".
4. Jika tidak, tampilkan "Tidak Lulus".

## Pseudocode:

- INPUT nilai\_akhir
- IF nilai\_akhir  $\geq 56$  THEN
- CETAK "Mahasiswa Lulus"
- ELSE
- CETAK "Mahasiswa Tidak Lulus"
- ENDIF

# Pseudocode Functionality

## Variables

Variabel adalah tempat menyimpan nilai.

*quotient, decimalNumber, newBase*

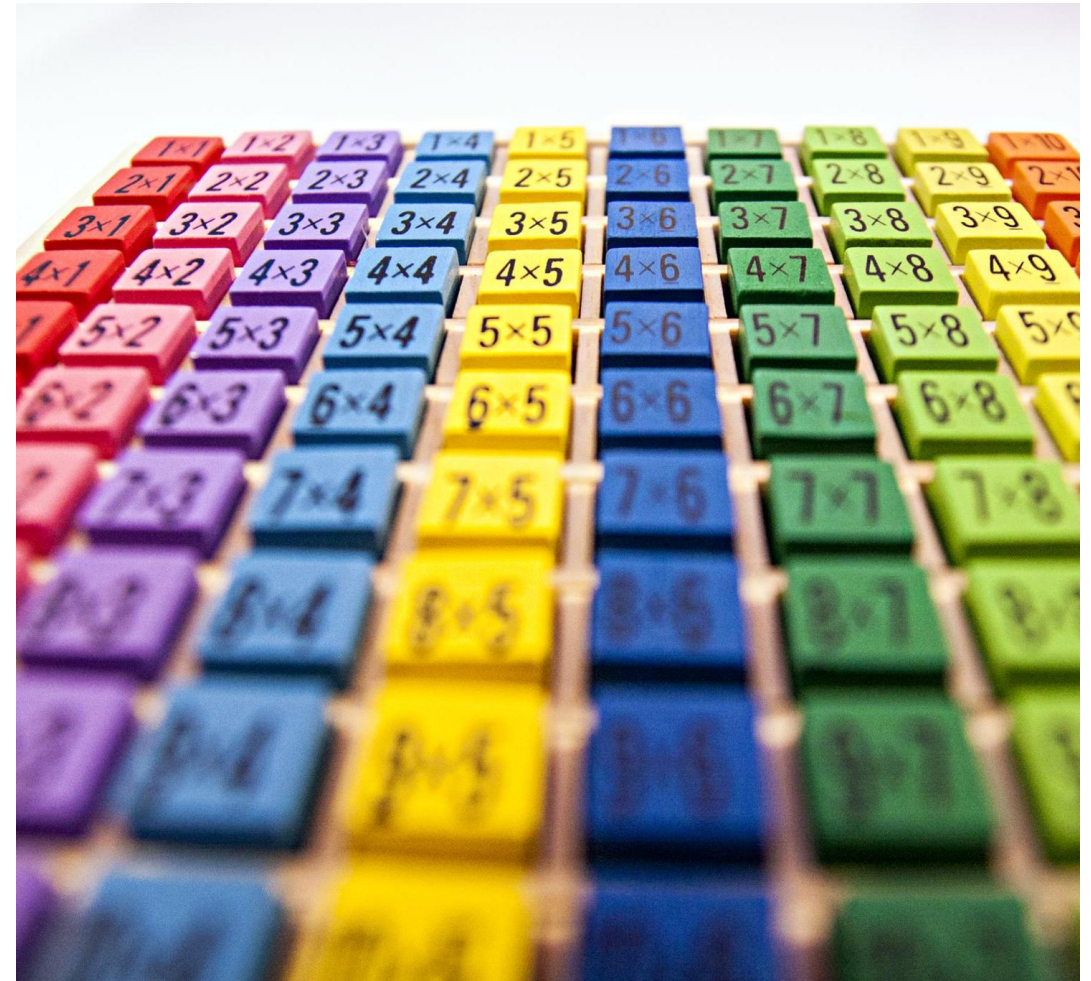
## Assignment

Menyimpan hasil ekspresi ke dalam variabel.

*quotient*  $\leftarrow$  64

*quotient*  $\leftarrow$  6 \* 10 + 4

SET *quotient* to 189



# Pseudocode Functionality

## Output

Menampilkan nilai ke perangkat keluaran.

WRITE, PRINT

## Input

Membaca nilai dari luar dan menyimpannya ke variabel.

GET, READ, INPUT



# Pseudocode Functionality

## Sequence/Urutan

Instruksi dieksekusi satu per satu sesuai urutannya..

**Example:** Menukar nilai dua bilangan

1. DECLARE A, B, C sebagai integer
2. INPUT A dan B
3.  $C \leftarrow A$
4.  $A \leftarrow B$
5.  $B \leftarrow C$

| 1 | A  | B  | C  |
|---|----|----|----|
| 2 | 10 | 5  | -  |
| 3 | -  | 5  | 10 |
| 4 | 5  | -  | 10 |
| 5 | 5  | 10 | -  |

# Pseudocode Functionality

## Selection / Decision (Percabangan)

Menentukan pilihan berdasarkan kondisi.

**Example:** Check if a certain number is a negative number or not

1. READ number
2. IF number < 0 THEN
3. WRITE "Bilangan negatif"
4. ELSE
5. WRITE "Bilangan nol atau positif"
6. ENDIF

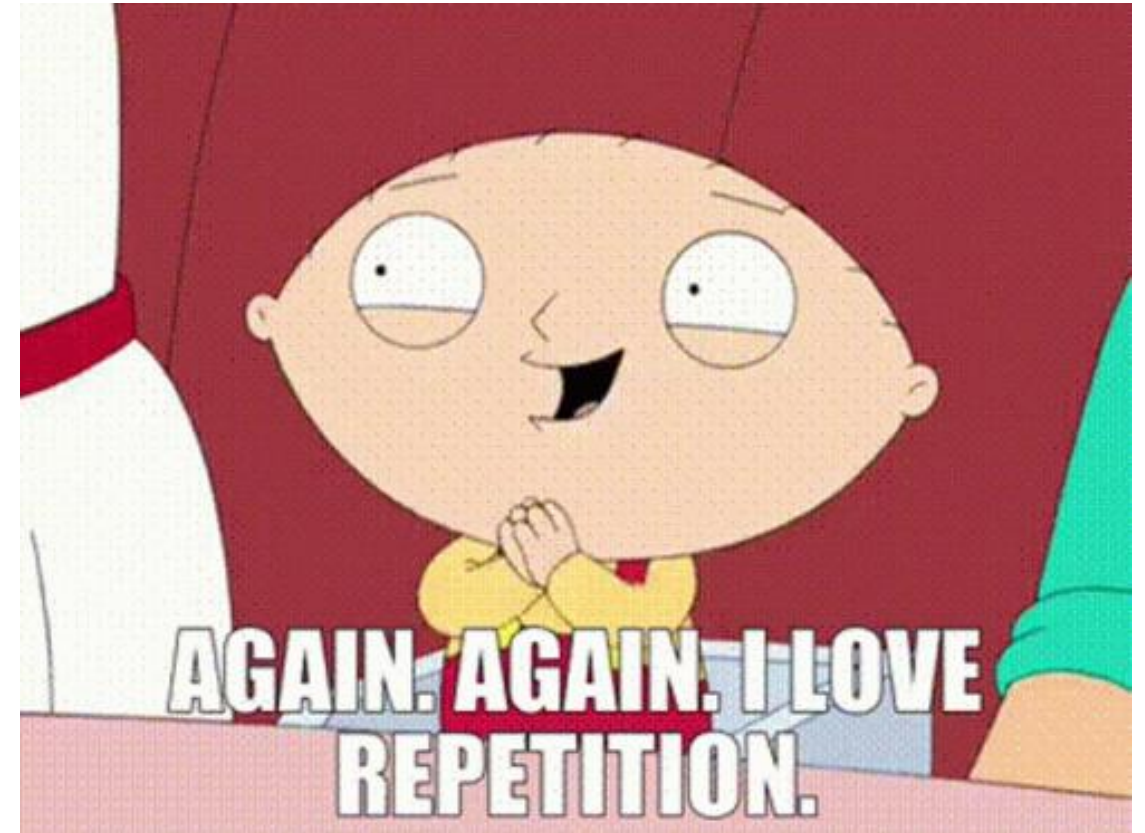
# Pseudocode Functionality

## Repetition / Perulangan

Mengulang serangkaian instruksi.

**Example:** Input and print 10 integer number sequentially

1. SET count = 1
2. WHILE count < 10
3. WRITE "Masukkan bilangan"
4. READ aNumber
5. WRITE "Anda memasukkan " + aNumber
6. count  $\leftarrow$  count + 1
7. ENDWHILE



# Three Constructs

```
do action 1  
do action 2  
...  
...  
do action  $n$ 
```

a. Sequence

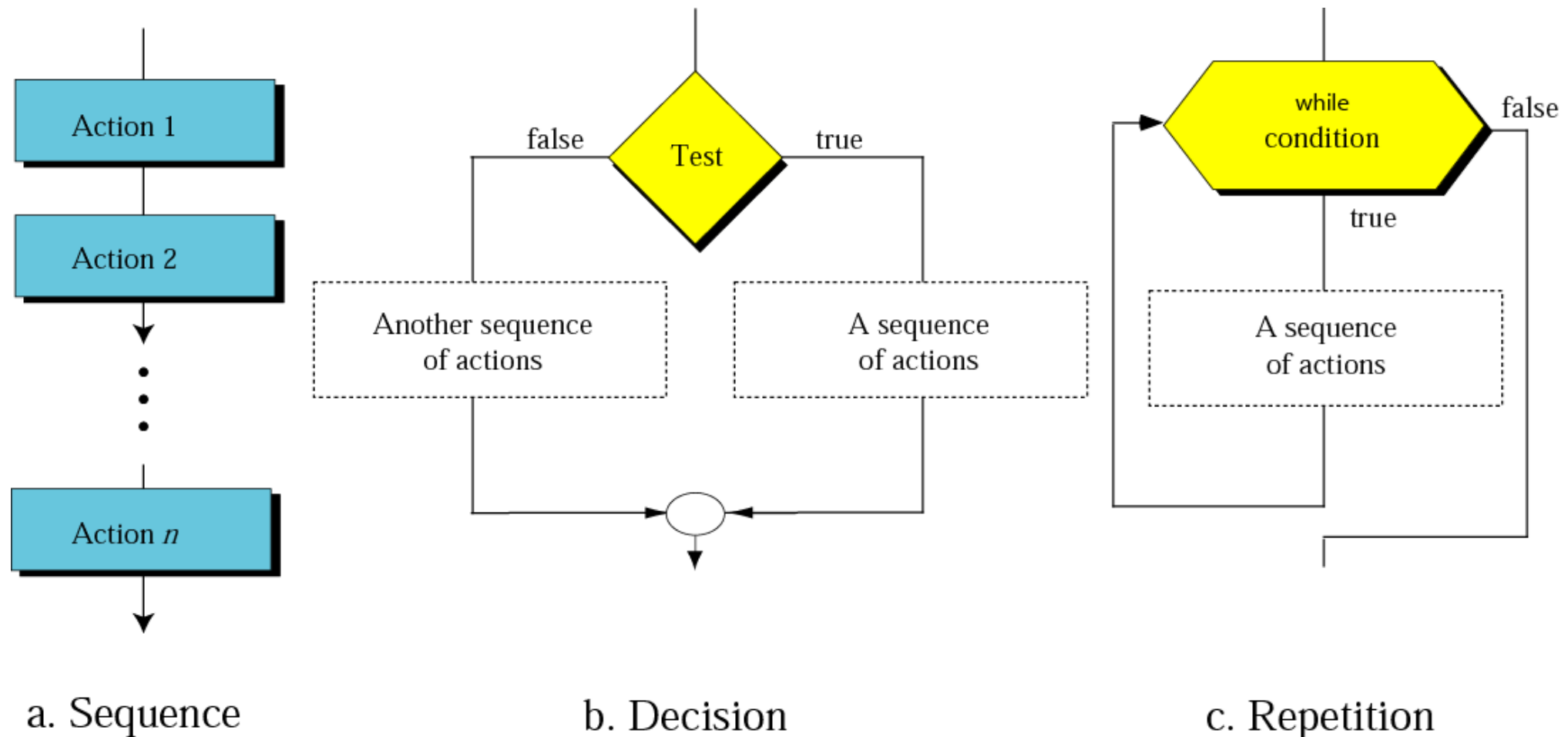
```
if a condition is true,  
then  
    do a series of actions  
else  
    do another series of actions
```

b. Decision

```
while a condition is true,  
    do action 1  
    do action 2  
    ...  
    ...  
    do action  $n$ 
```

c. Repetition

# Three Constructs (Flowchart)



# Three Constructs (Pseudocode)

```
action 1  
action 2  
⋮  
action n
```

a. Sequence

```
if (condition)  
  then  
    action  
    action  
    ...  
  else  
    action  
    action  
    ...  
End if
```

b. Decision

```
while (condition)  
  action  
  action  
  ...  
End while
```

c. Repetition

# Which one to Choose between Pseudocode or Flowchart?

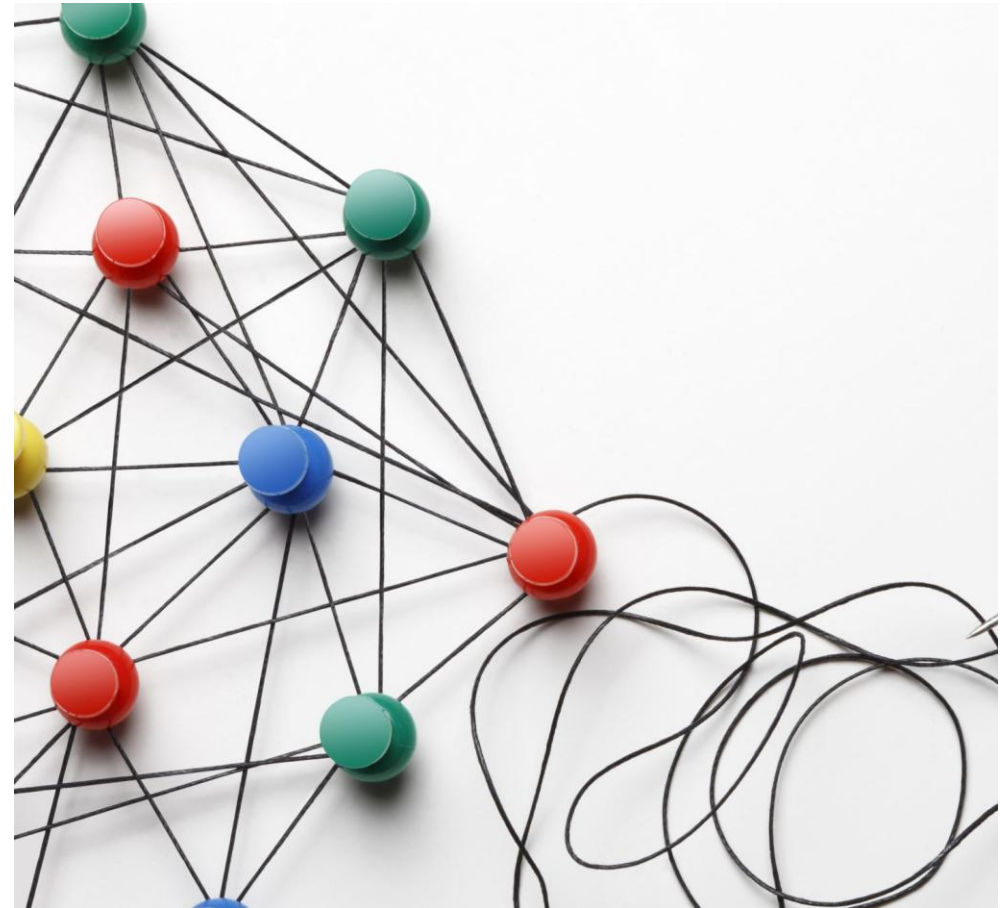
## Flowchart

Menggunakan simbol-simbol untuk menggambarkan tindakan.

## Pseudocode

Gabungan antara kode program dan bahasa sehari-hari.

Gunakan yang paling membantu Anda dalam menyusun kode program.



# Summary

- Pseudocode adalah gabungan antara kode program dan bahasa manusia untuk menjelaskan penyelesaian masalah.
- Pseudocode menggunakan bentuk yang mirip bahasa pemrograman.
- Terdapat tiga struktur dasar:
  - Sequence (Urutan)
  - Decision (Percabangan)
  - Repetition (Perulangan)

# Task

1. Buatlah flowchart dan Pseudocode untuk menghitung keliling dan luas lingkaran
  - Keliling =  $2 * \text{PI} * \text{radius}$
  - Luas =  $0.5 * \text{PI} * \text{radius} * \text{radius}$

# Task

2. Buatlah flowchart dan Pseudocode untuk mengkonversi suhu dari Celcius ke Fahrenheit dengan rumus

$$F = C * 1.8 + 32$$

# Task

3. Buatlah flowchart dan Pseudocode untuk menampilkan sederetan bilangan ganjil dari 10 sampai 30 kecuali 21 dan 27

Contoh:

Output : 11 13 15 17 19 23 25 29

# Reference

1. Karl, Beecher. "Computational Thinking: A Beginner's Guide to Problem-Solving and Programming." Swindon, UK: BCS, The Chartered Institute for IT (2017).
2. Dale, Nell B., and John Lewis. Computer science illuminated. Jones & Bartlett Learning, 2007.



# THANK YOU!