

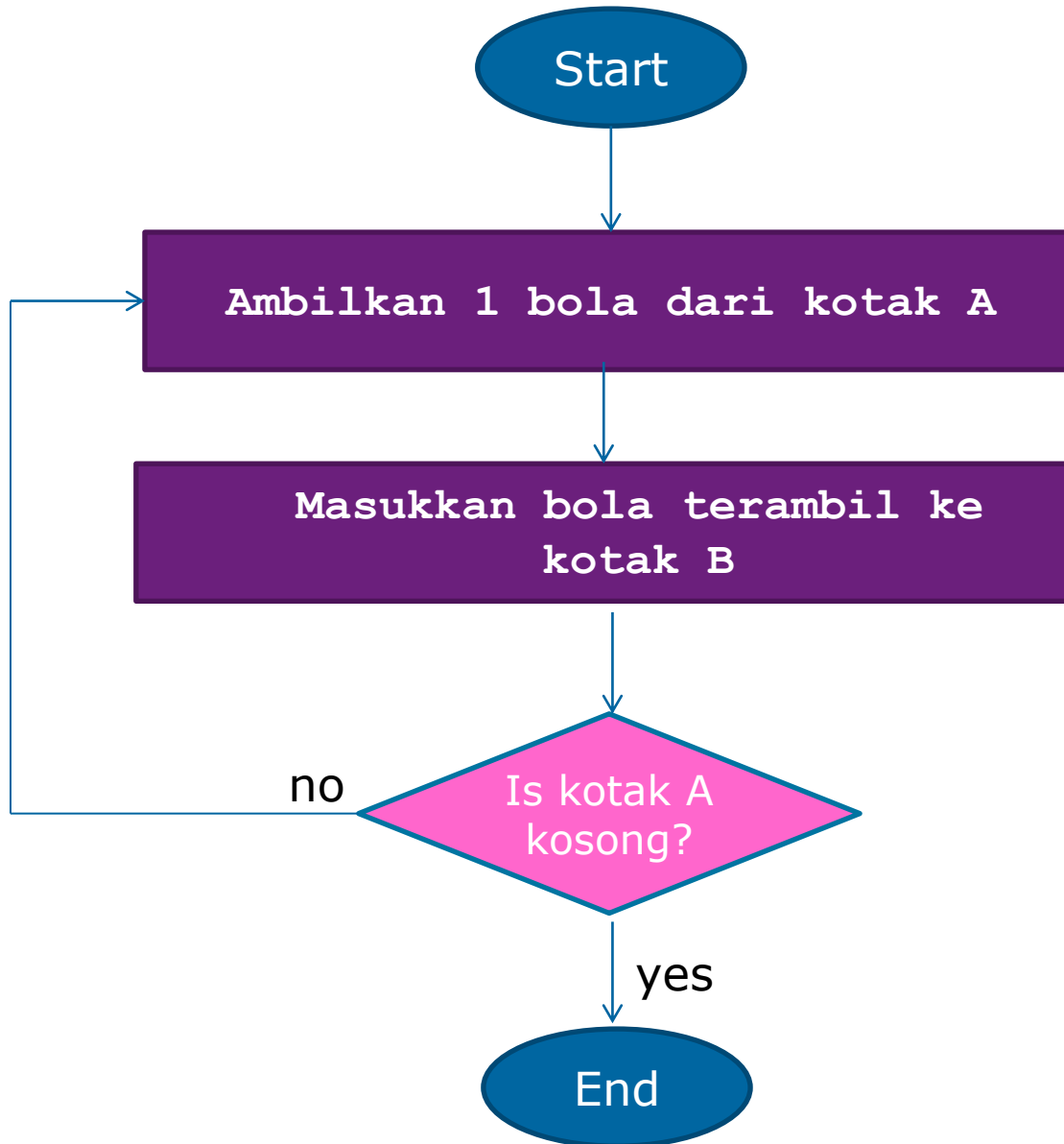
KUG1C3

Dasar Algoritma dan Pemrograman



Loop

Remember..



Loop Introduction

- ▶ Computer is able to repeat an instruction, even action, over and over again with the same performance
- ▶ In contrast to humans who tend to make mistakes when doing the same thing (because of fatigue or boredom),



Loop Introduction

- ▶ Loop must stop
- ▶ An endless loop algorithm is a wrong algorithm
- ▶ An endless loop must be detected by programmer even before the program executed



Loop Structure

2 parts :

- ▶ A conditions that lead to the repetition of a moment to stop, which is expressed by a logical expression either explicitly or implicitly
- ▶ body of the loop, i.e. the action that should be repeated as long as the specified condition is met for repetition

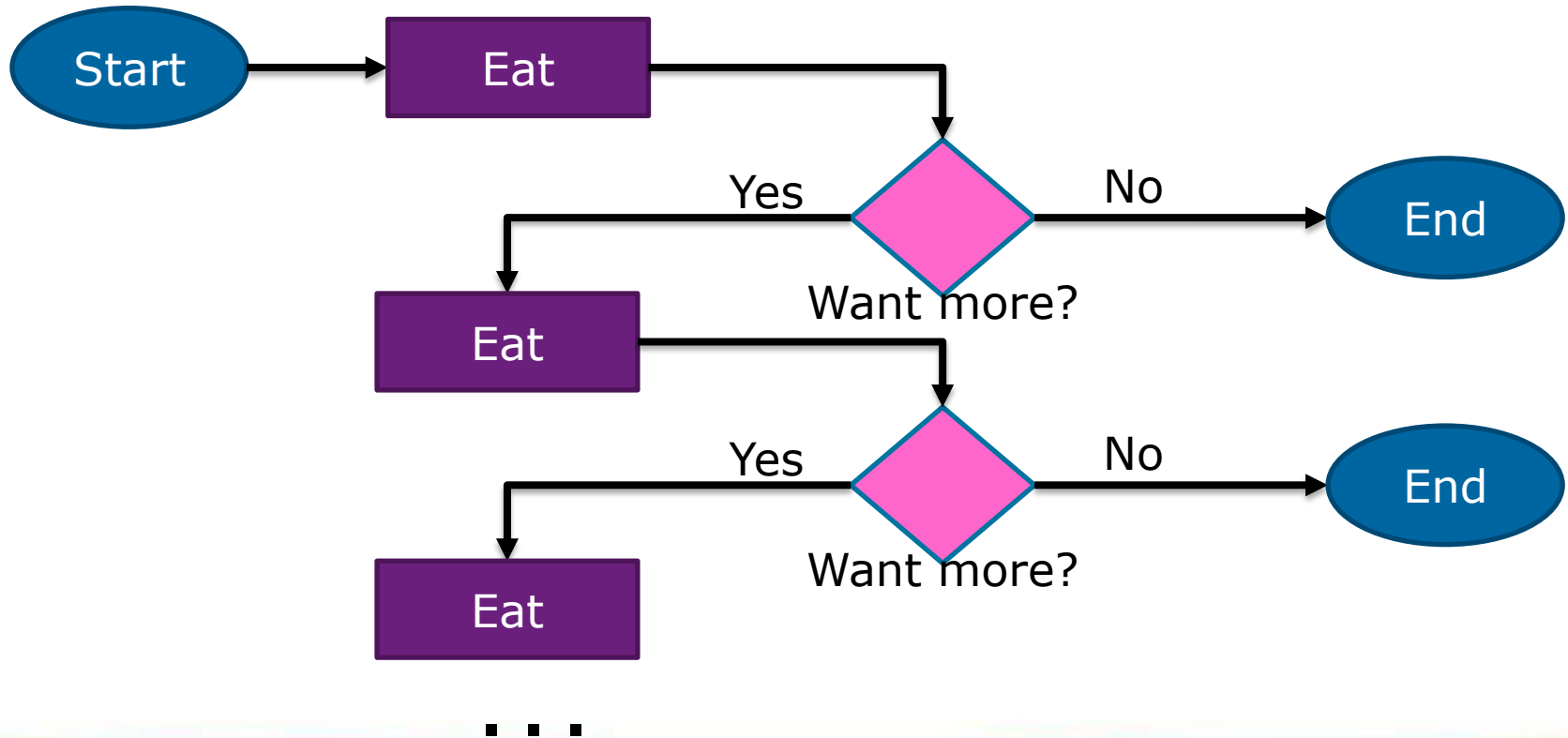
Loop Structure

In addition, the loop structure is usually accompanied by parts:

- ▶ initialization, the action is performed before the loop was first performed.
- ▶ termination, i.e. the actions taken after the repetitions completed

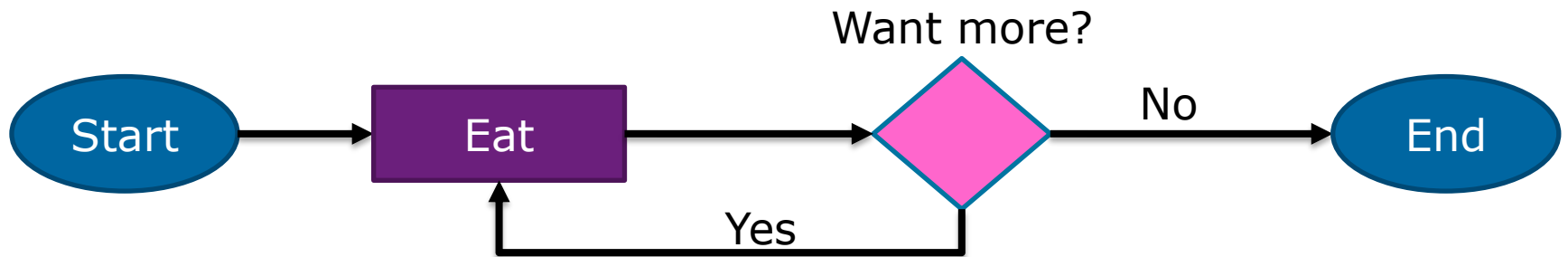
Loop

- ▶ I want to eat until I can't eat anymore



Loop : Repeat-Until, Iterate-Stop

- ▶ I want to eat until I can't eat anymore

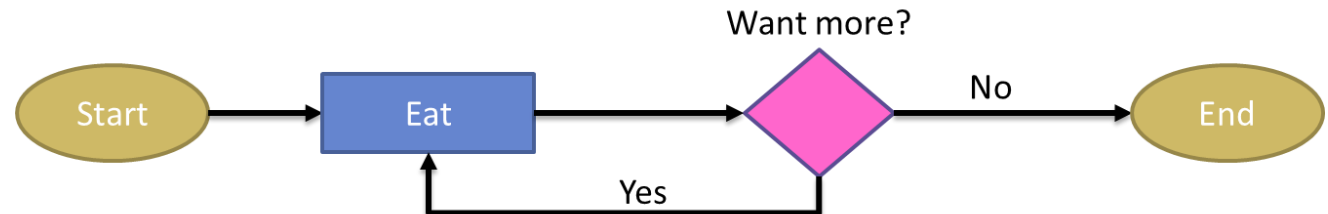


repeat-until

repeat

action

until (condition)



repeat-until

Program eat_food

Kamus

more: boolean

Algoritma

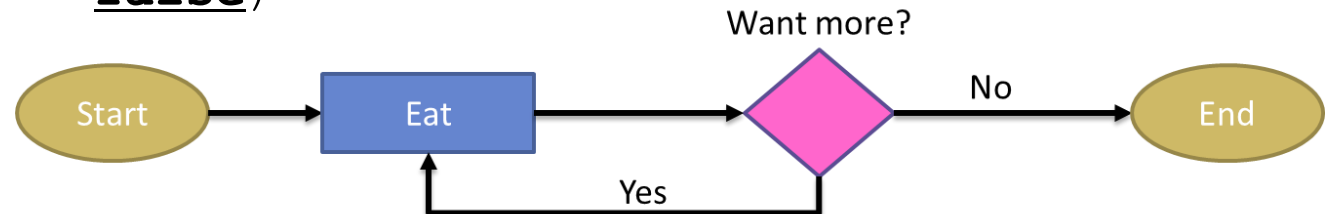
repeat

output ('eat')

output ('want more?')

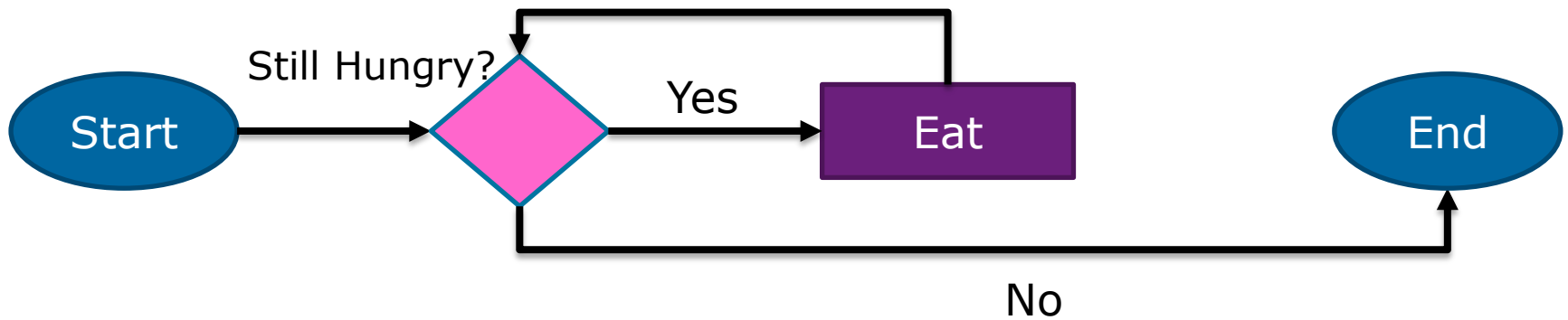
input (more)

until (more = false)



Loop : While-do

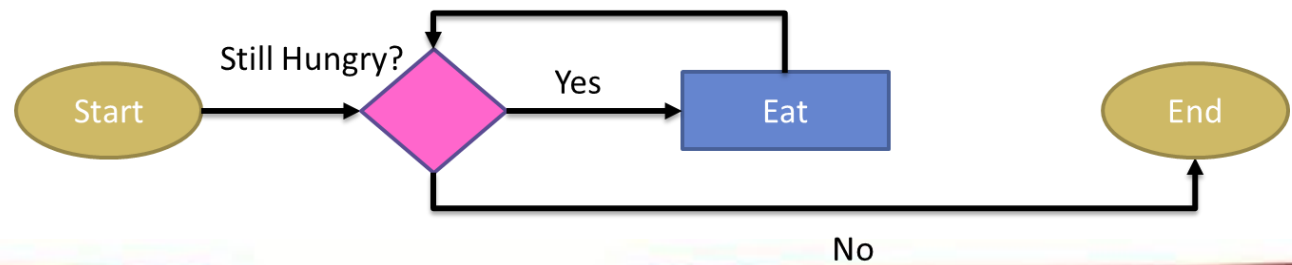
- ▶ While I'm hungry, I'm gonna eat



while-do

while condition do

action



while-do

Program eat_food

Kamus

hungry : boolean

Algoritma

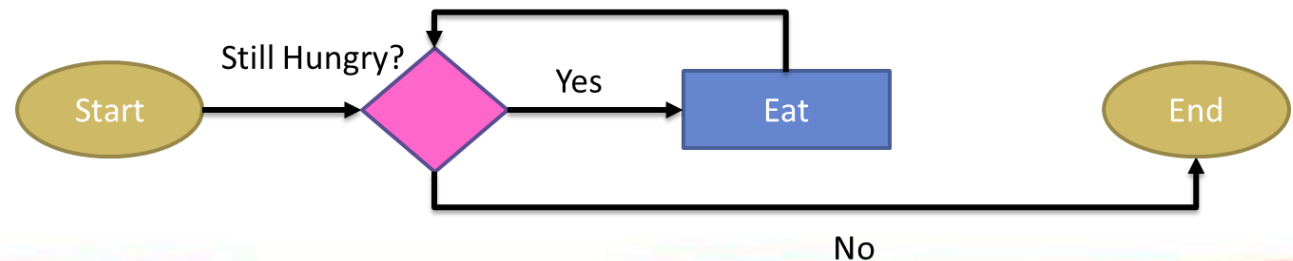
hungry ← true

while hungry do

output('eat')

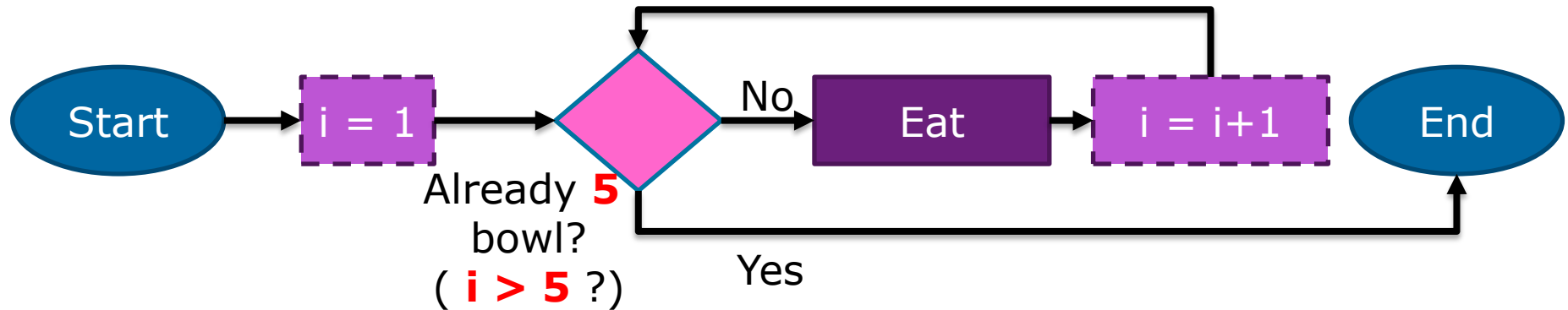
output(still hungry?)

input(hungry)



Loop : Repeat-n-times

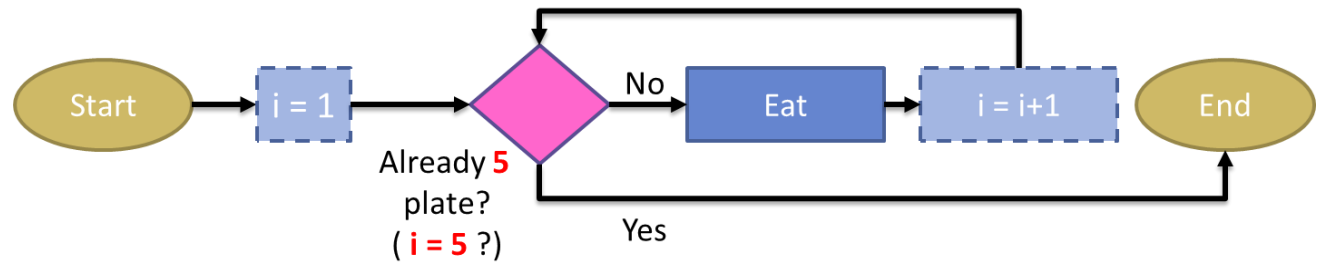
- ▶ I'm going to eat **5** bowls of meatball



repeat-n-times

repeat n times

action



repeat-n-times

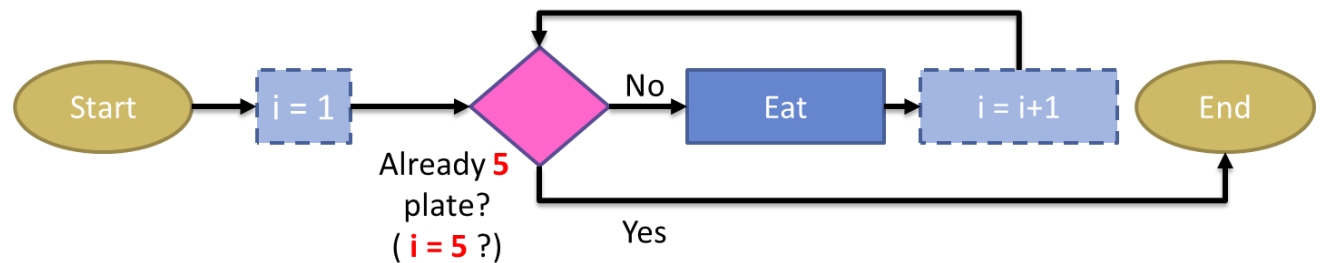
Program eat_food

Kamus

Algoritma

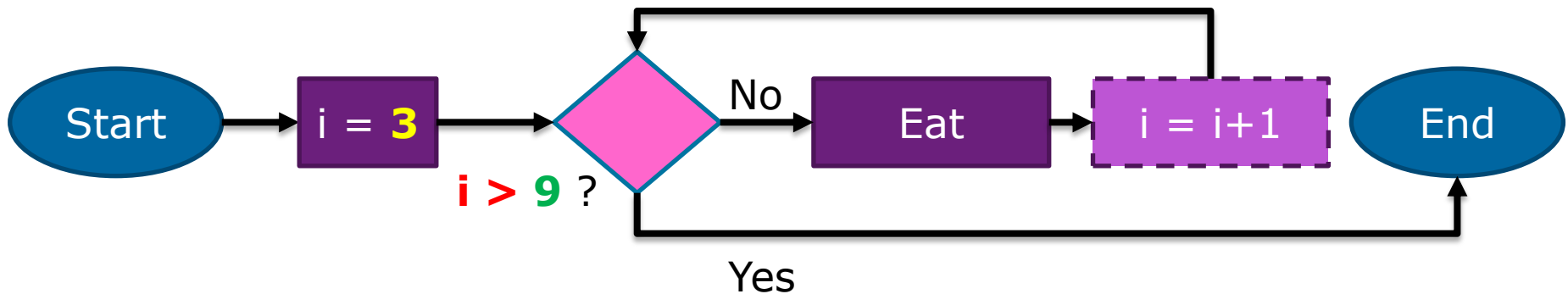
repeat 5 times

output ('eat')



Loop : Repeat-n-times, Traversal

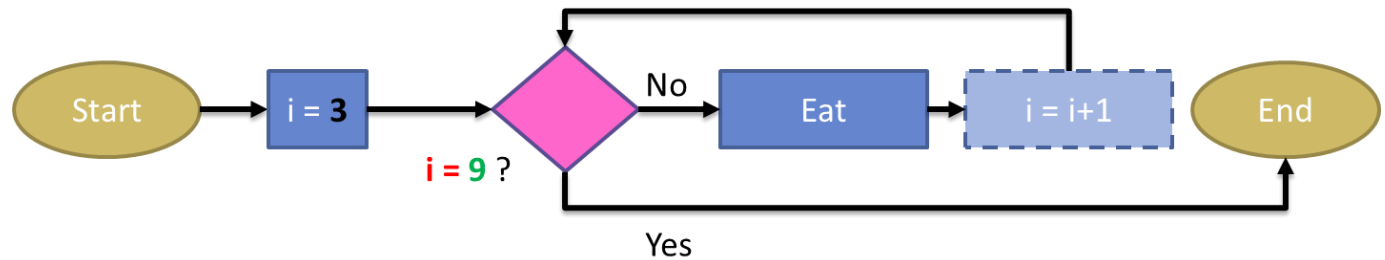
- ▶ There're some plates of food,
- ▶ I'm going to eat plate number 3 to 9



Traversal

i traversal [n..m]

action



Traversal

Program eat_food

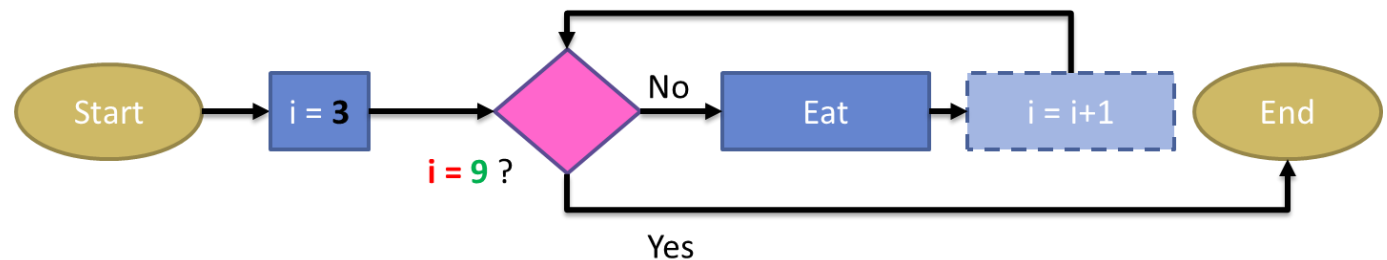
Kamus

i : integer

Algoritma

i traversal [3..9]

output ('eat')



Looping Notation

There are four kinds of repetition notation:

- ▶ based on the number of repetitions (repeat times)
- ▶ by stop condition (repeat until)
- ▶ based on the condition of repetition (while do)
- ▶ based enumerator (traversal)

Example-1

- ▶ Buat algoritma untuk menampilkan bilangan dari 1 sampai n , dimana n adalah berupa masukan dari user!
- ▶ Buat algoritma untuk menampilkan bilangan dari m sampai n , dimana m dan n berupa masukan dari user, $m < n$!

Example-2

- ▶ Buat algoritma untuk menjumlahkan sejumlah bilangan! Bilangan akan terus diinputkan sampai user menginputkan bilangan -999.

Exercise 1

- ▶ Buat algoritma untuk menghitung m^n , dimana m dan n didapatkan dari masukan user!
- ▶ Contoh
 - 2 {input m }
 - 3 {input n }
 - 8 {output 2^3 }

Exercise 2

- ▶ Buat algoritma untuk melakukan tebak angka sampai benar! Angka yang harus ditebak didefinisikan melalui masukan user.

Contoh:

5 → {input angka yang harus ditebak}

3 → {input tebakan}

4 → {input tebakan}

8 → {input tebakan}

5 → {input tebakan}

{selesai}

Exercise 3

- ▶ Buatlah program untuk menampilkan deret n bilangan ganjil pertama; $n > 0!$

- ▶ Contoh:

5 {input n }

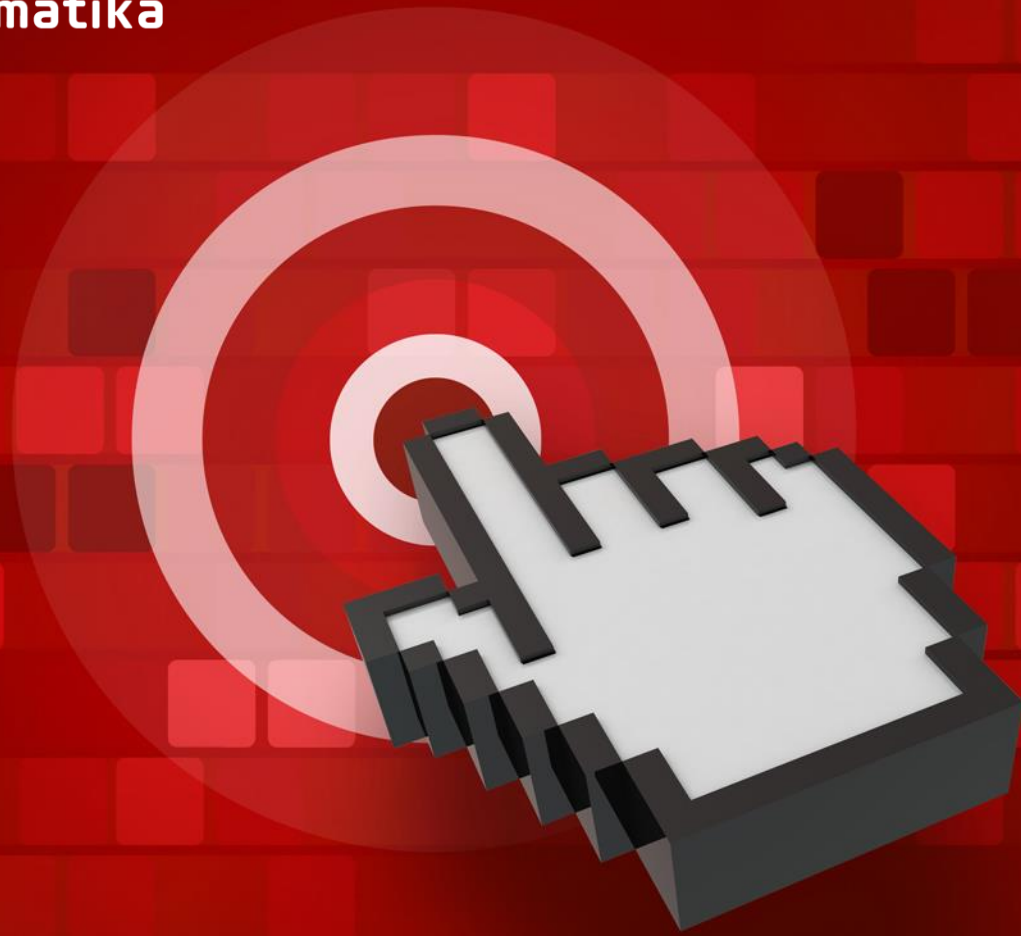
1 3 5 7 9 {output deret}

Exercise 4

- ▶ Buat fungsi ***pangkat (m, n: integer) → integer*** untuk menghitung m^n !
- ▶ Dengan memanfaatkan fungsi ***pangkat (m, n: integer) → integer***, buat algoritma untuk menampilkan deret bilangan x^a sampai x^b !
- ▶ Contoh:
 - 3 {input x}
 - 2 {input a}
 - 6 {input b}
 - 9 27 81 243 729 {output deret $3^2 .. 3^6$ }



Fakultas Informatika
School of Computing
Telkom University



THANK YOU