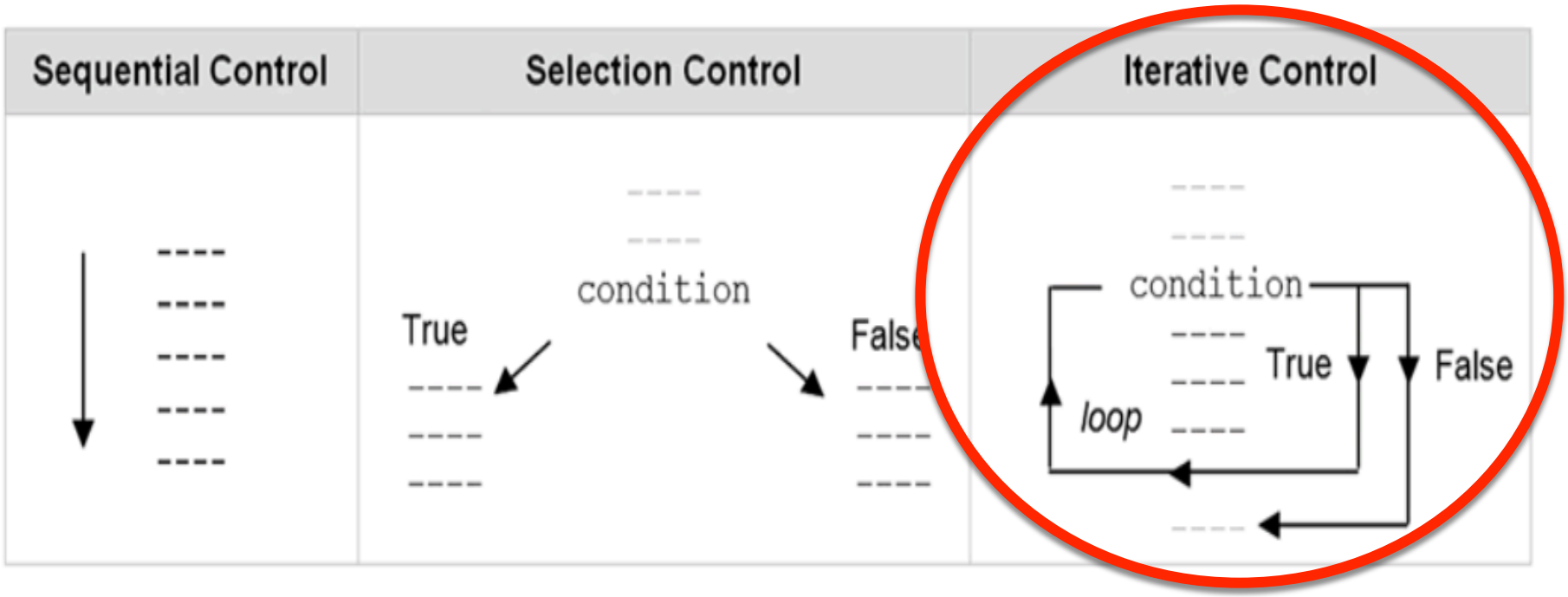
UNIVERSITAS SAM RATULANGI MANADO
FAKULTAS TEKNIK, JURUSAN TEKNIK ELEKTRO
Program Studi S-1 Teknik Informatika

Kendali Iteratif

Mata Kuliah: Algoritma & Logika Informatika (IFC3504)

Alwin M. Sambul, S.T., M.Eng., Ph.D.

Bentuk2 dasar kendali



11. Kendali Iteratif

Statemen Kendali Iteratif

- **Statemen kendali iteratif:** Statemen kendali yang digunakan untuk melakukan eksekusi **berulang** pada sekelompok statemen
- Ada dua statemen kendali iteratif dalam Python: **while** dan **for**.
  - Kita akan belajar dulu tentang **while, for** akan kita pelajari setelah kita belajar tentang **sequence**.

Statemen **while**

Statemen while: Statemen kendali iteratif yg mengeksekusi sekelompok statemen secara berulang tergantung pada suatu ***condition*** (ekspresi Boolean) tertentu.

while statement	Example use

while *condition:*
    suite
	sum = 0
	current = 1
	
	n = int(input('Enter value: '))
	
	while current <= n:
	    sum = sum + current
	    current = current + 1

Statemen **while**

Statemen while: Statemen kendali iteratif yg mengeksekusi sekelompok statemen secara berulang tergantung pada suatu ***condition*** (ekspresi Boolean) tertentu.

while statement	Example use

`while condition:`
`    suite`
	`sum = 0`
	`current = 1`
	
	`n = int(input('Enter value: '))`
	
	`while current <= n:`
	`    sum = sum + current`
	`    current = current + 1`

Loop

Apa yg dikerjakan program ini?

while statement	Example use

`while condition:`
`    suite`
	`sum = 0`
	`current = 1`
	
	`n = int(input('Enter value: '))`
	
	`while current <= n:`
	`    sum = sum + current`
	`    current = current + 1`

Katakanlah, user menginputkan angka **3**

Iteration	sum	current	current <= 3	sum = sum + current	current = current + 1
1	0	1	True	sum = 0 + 1 (1)	current = 1 + 1 (2)
2	1	2	True	sum = 1 + 2 (3)	current = 2 + 1 (3)
3	3	3	True	sum = 3 + 3 (6)	current = 3 + 1 (4)
4	6	4	False	loop termination	

Pengecekan Kesalahan Input

- Statemen **while** sangat cocok digunakan untuk pengecekan kesalahan output

test01.py - /Users/asambul/Programming/python/kuliah3/test01.py (3.4.3)

```
#tanpa input checking
sum = 0
current = 1
n = int(input("Enter value: "))
while current <= n:
    sum = sum + current
    current = current + 1
print("Sum of values from 1 to ",n," is ",sum)
```

Ln: 9 Col: 0

Pengecekan Kesalahan Input

- Statemen **while** sangat cocok digunakan untuk pengecekan kesalahan output

test02.py - /Users/asambul/Programming/python/kuliah3/test02.py (3.4.3)

```
#dengan input checking
sum = 0
current = 1
n = int(input("Enter value: "))
while n < 1 or n > 10:
    print("Must be between 1 and 10")
    n = int(input("Enter value: "))
while current <= n:
    sum = sum + current
    current = current + 1
print("Sum of values from 1 to ",n," is ",sum)
```

Ln: 13 Col: 0

Perulangan Infinitif

- **Perulangan infinitif**: struktur kendali iteratif yang tidak pernah selesai perulangannya, biasanya karena **kesalahan semantik**

test03.py - /Users/asambul/Programming/python/kuliah3/test03.py (3.4.3)

```
#mensimulasikan infinity loop
sum = 0
current = 1
n = int(input("Enter value: "))
while n < 1 or n > 10:
    print("Must be between 1 and 10")
    n = int(input("Enter value: "))
while current <= n:
    sum = sum + current
    #current = current + 1
    #current akan selamanya bernilai 1
print("Sum of values from 1 to ",n," is ",sum)
```

Ln: 4 Col: 31

Perulangan Tentu

- **Perulangan tentu:** perulangan dimana **jumlah** perulangannya diketahui sebelum perulangan dieksekusi

test01.py - /Users/asambul/Programming/python/kuliah3/test01.py (3.4.3)

```
#tanpa input checking
sum = 0
current = 1
n = int(input("Enter value: "))
while current <= n:
    sum = sum + current
    current = current + 1
print("Sum of values from 1 to ",n," is ",sum)
```

Perulangan tentu

Ln: 9 Col: 0

Perulangan tak tentu

- **Perulangan tentu:** perulangan dimana **jumlah** perulangannya tidak dapat diketahui sebelumnya

test02.py - /Users/asambul/Programming/python/kuliah3/test02.py (3.4.3)

```
#dengan input checking
sum = 0
current = 1
n = int(input("Enter value: "))
while n < 1 or n > 10:
    print("Must be between 1 and 10")
    n = int(input("Enter value: "))
while current <= n:
    sum = sum + current
    current = current + 1
print("Sum of values from 1 to ",n," is ",sum)
```

Perulangan tak tentu

Ln: 13 Col: 0

Boolean Flag

- **Boolean Flag**: Variabel Boolean yang mengendalikan perulangan

test04.py - /Users/asambul/Programming/python/kuliah3/test04.py (3.4.3)

```
#dengan Boolean flag
sum = 0
current = 1
n = int(input("Enter value: "))
ulangi = True
while ulangi:
    if n < 1 or n > 10:
        print("Must be between 1 and 10")
        n = int(input("Enter value: "))
    else:
        ulangi = False
while current <= n:
    sum = sum + current
    current = current + 1
print("Sum of values from 1 to ",n," is ",sum)
```

Ln: 18 Col: 0

Latihan

Latihan:

Bilangan2 Fibonacci:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Buatlah program untuk mencetak **n** bilangan Fibonacci

Fibonacci

Portrait by unknown artist

Born	c. 1170–75 Pisa[1]
Died	c. 1240–50 most likely Pisa
Nationality	Italian

fibonacci.py - /Users/asambul/Programming/python/kuliah3/fibonacci.py (3.4.3)

```
#mencetak n bilangan fibonacci
n = int(input("Masukkan n = "))
while n < 1:
    n = int(input("Masukkan n = "))
k = 0
f1 = 0
f2 = 1
while k < n:
    k = k + 1
    if k == 1:
        print("Bilangan ke-",k,": ",f1)
    elif k == 2:
        print("Bilangan ke-",k,": ",f2)
    else:
        temp = f2
        f2 = f1 + f2
        f1 = temp
        print("Bilangan ke-",k,": ",f2)
```

Ln: 9 Col: 13

Tips: **Print** tanpa **space** atau **newline**

- Statemen **print** tanpa space:
  - print(“Hello”,”World”,sep=“”)
- Statemen **print** tanpa newline:
  - print(“Hello World”,end=“”)
  - print(“Python is fun”)

fibonacci3.py - /Users/asambul/Programming/python/kuliah3/fibonacci3.py (3.4.3)

```
#mencetak n bilangan fibonacci
n = int(input("Masukkan n = "))
while n < 1:
    n = int(input("Masukkan n = "))
k = 0
f1 = 0
f2 = 1
while k < n:
    k = k + 1
    if k == 1:
        print(f1,end="")
    elif k == 2:
        print(f2,end="")
    else:
        temp = f2
        f2 = f1 + f2
        f1 = temp
        print(f2,end="")
    if n != k:
        print(",",end="")
```

Ln: 10 Col: 1