

# Notasi Algoritmik dan Padanan dalam Bahasa Python 3 - Analisis Kasus, Pengulangan, Subprogram

**Tim Pengajar IF1210**

Sekolah Teknik Elektro dan Informatika

# Analisis Kasus

# Analisis Kasus - Satu Kasus

## Notasi Algoritmik

```
if (<kondisi>) then  
    <aksi>  
{ jika kondisi=false  
  tidak terdefinisi aksi }
```

## Python

```
if (<kondisi>):  
    <aksi>  
# jika kondisi=false  
# tidak terdefinisi aksi
```

# Analisis Kasus - Dua Kasus Komplementer

## Notasi Algoritmik

```
if (<kondisi>) then  
    <aksi-1>  
else { kondisi=false }  
    <aksi-2>
```

## Python

```
if (<kondisi>):  
    <aksi-1>  
else: # kondisi=false  
    <aksi-2>
```

# Analisis Kasus - Banyak Kasus

| Notasi Algoritmik   | Python  |
|---|---|
| <pre><u>depend on</u> (&lt;var1&gt;,...)<br/>    &lt;kondisi-1&gt; : &lt;aksi-1&gt;<br/>    &lt;kondisi-2&gt; : &lt;aksi-2&gt;<br/>    ...<br/><u>else</u> { &lt;kondisi-n&gt; }<br/>    &lt;aksi-n&gt;</pre> | <pre><b>if</b> (&lt;kondisi-1&gt;):<br/>    &lt;aksi-1&gt;<br/><b>elif</b> (&lt;kondisi-2&gt;):<br/>    &lt;aksi-2&gt;<br/>...<br/><b>else</b>: # <i>kondisi-n</i><br/>    &lt;aksi-n&gt;</pre> |

# Pengulangan

# Konsep Pengulangan

- Di notasi algoritmik ada 5 jenis pengulangan:
  - Pengulangan berdasarkan jumlah pengulangan
  - Pengulangan berdasarkan kondisi berhenti [di akhir]
  - Pengulangan berdasarkan kondisi mengulang [di awal]
  - Pengulangan berdasarkan dua aksi
  - Pengulangan berdasarkan pencacah

# Pengulangan – Berdasarkan jumlah pengulangan

| Notasi Algoritmik                      | Python                                       |
|--|--|
| <u>repeat</u> n <u>times</u><br><aksi> | <b>for</b> i <b>in</b> range (n) :<br><aksi> |

Penggunaan:

- Diketahui secara persis **berapa kali** aksi harus dilakukan dan aksi terdefinisi



# Pengulangan – Berdasarkan kondisi berhenti

| Notasi Algoritmik  | Python   |
|--|--|
| <u><b>repeat</b></u><br><aksi><br><u><b>until</b></u> <kondisi-berhenti> | <b>while True:</b><br><aksi><br><b>if</b> (<kondisi-berhenti>) :<br><b>break</b> |

Penggunaan:

- Aksi minimum dilakukan 1x (karena kondisi berhenti baru dicek setelah aksi dieksekusi)
- Kondisi berhenti berupa ekspresi boolean (lebih luas dari sekedar harus mengulang berapa kali)

# Pengulangan – Kondisi mengulang di awal

## Notasi Algoritmik

```
while <kondisi-mengulang> do  
    <aksi>  
{ Kondisi berhenti dicapai di sini }
```

## Python

```
while (<kondisi-mengulang>) :  
    <aksi>  
# Kondisi berhenti dicapai di sini
```

### Penggunaan:

- Dimungkinkan aksi tidak pernah dilakukan (karena kondisi mengulang dicek sebelum aksi dilakukan)  
→ kasus yang menyebabkan aksi tidak pernah dilakukan disebut sbg. **kasus kosong**

# Pengulangan – Berdasarkan dua aksi (1)

## Notasi Algoritmik

### iterate

*<aksi-1>*

### stop *<kondisi-berhenti>*

*<aksi-2>*

{ Kondisi berhenti dicapai di sini }

## Python

### **while True:**

*<aksi-1>*

**if** (*<kondisi-berhenti>*) :

**break**

**else:**

*<aksi-2>*

# Kondisi berhenti dicapai di sini

### Penggunaan:

- Merupakan gabungan antara repeat-until dan while-do
- aksi-1 minimum dilakukan 1x dan aksi-2 bisa tidak dilakukan sama sekali

# Pengulangan – Berdasarkan pencacah

## Penggunaan

- Pencacah harus bertipe ordinal, contoh: integer
- Diketahui dengan pasti nilai awal dan nilai akhir pencacah

## Notasi Algoritmik

```
{ Mencacah maju }
<pencacah> traversal [<min>..<maks>]
    <aksi>
{ Mencacah mundur }
<pencacah> traversal [<maks>..<min>]
    <aksi>
```

## Python

```
# Mencacah maju
for <pencacah> in range (<min>, <maks+1>) :
    <aksi>
# Mencacah mundur
for <pencacah> in range (<maks>, <min-1>, -1) :
```

# Subprogram

## Subprogram

*A set of instructions designed to perform a frequently used operation within a program*

### **Function:**

pemetaan suatu nilai domain (input) ke range (output)

### **Procedure:**

deretan instruksi yang jelas initial state dan final state-nya

# Fungsi dalam Notasi Algoritmik

**function** *nama\_fungsi* (*param1: type1*, *param2: type2*, ..., *paramn: typen*) → *type\_hasil*  
 { Spesifikasi fungsi }

## KAMUS LOKAL

{ Konstanta, variabel, fungsi/prosedur yang dideklarasikan local di function }

## ALGORITMA

{ Langkah-langkah algoritmik untuk fungsi }  
 { Return hasil menggunakan notasi : → *hasil* }

## Contoh

**function** pangkat2 (*x* : integer) → integer  
 { Menghasilkan pangkat2 dari *x* }

## KAMUS LOKAL

## ALGORITMA

→  $x * x$

# Prosedur dalam Notasi Algoritmik

**procedure** *nama\_proc* ([input/output] *param1:type1*, ..., [input/output] *paramn:typen*)  
 { Spesifikasi prosedur }

## KAMUS LOKAL

{ Konstanta, variabel, fungsi/prosedur yang dideklarasikan local di prosedur }

## ALGORITMA

{ Langkah-langkah algoritmik untuk prosedur, tidak ada return hasil }

## Contoh

**procedure** CetakNama (input nama : string)  
 { I.S.: nama terdefinisi; F.S.: tercetak "Hello + nama" di layar }

## KAMUS LOKAL

## ALGORITMA

output ("Hello ", nama)



# Notasi Algoritmik – Prosedur

- Jenis parameter prosedur
  - input: sebelum prosedur dipanggil, nilai harus sudah terdefinisi
  - output: sebelum prosedur dipanggil nilai bebas/sembarang, setelah prosedur dijalankan nilai terdefinisi
  - input/output: sebelum prosedur dipanggil nilai sudah terdefinisi dan nilai bisa berubah setelah prosedur dijalankan

Pemanggilan  
Fungsi/Prosedur  
dalam  
Notasi Algoritmik

Program *Judul*

{ Spesifikasi Program }

**KAMUS**

{ Variabel, konstanta, definisi fungsi/prosedur }

{ Definisi dan spesifikasi fungsi/prosedur }

function ...

{ spesifikasi function }

procedure ...

{ spesifikasi procedure }

**ALGORITMA PROGRAM UTAMA**

{ Algoritma program mengandung pemanggilan  
fungsi/prosedur }

{ **REALISASI FUNGSI/PROSEDUR** }  
{ lihat slide berikutnya }

Pemanggilan  
Fungsi/Prosedur  
dalam  
Notasi Algoritmik

```
{ REALISASI FUNGSI/PROSEDUR}  
function ...  
{ spesifikasi function }  
KAMUS LOKAL  
...  
ALGORITMA  
...  
procedure ...  
{ spesifikasi procedure }  
KAMUS LOKAL  
...  
ALGORITMA  
...
```

**Program** CetakKuadrat

```
{ Input: masukan integer dan
  mencetak nilai kuadratnya ke layar }
```

**KAMUS**

bil : integer

function Kuadrat (x : integer) → integer  
 { menghasilkan kuadrat x }

procedure CetakInt (input x : integer)  
 { I.S. x terdefinisi; F.S. x tercetak ke layar }

**ALGORITMA PROGRAM UTAMA**

```
input(bil)   { baca dari keyboard }
CetakInt (Kuadrat(bil))
```

**{ REALISASI FUNGSI/PROSEDUR }**

function Kuadrat (x : integer) → integer  
 { menghasilkan kuadrat x }

**KAMUS LOKAL****ALGORITMA**

→ x \* x

procedure CetakInt (input x : integer)  
 { I.S. x terdefinisi; F.S. x tercetak ke layar }

**KAMUS LOKAL****ALGORITMA**

output(x)

# Contoh Pemanggilan Fungsi/Prosedur di Notasi Algoritmik

# Fungsi di Python

```
def nama_fungsi (param1, param2, ..., paramn)  
# Spesifikasi fungsi, type hasil berdasarkan nilai return  
  
# KAMUS LOKAL  
# Konstanta, variabel, fungsi/prosedur yang dideklarasikan lokal di function  
  
# ALGORITMA  
# Langkah-langkah algoritmik untuk fungsi  
# Return hasil menggunakan notasi : return hasil
```

## Contoh

```
def pangkat2 (x)  
# Menghasilkan pangkat2 dari x  
  
# KAMUS LOKAL  
  
# ALGORITMA  
    return x * x
```

# Prosedur di Python

Di Python, semua parameter berjenis input.

Jika butuh parameter output atau input/output, harus dibuat sebagai fungsi.

```
def nama_proc (param1, param2, ..., paramn)
# Spesifikasi prosedur
# KAMUS LOKAL
# Konstanta, variabel, fungsi/prosedur yang dideklarasikan lokal di prosedur
# ALGORITMA
# Langkah-langkah algoritmik untuk prosedur, return tanpa memberikan hasil
```

## Contoh

```
def CetakNama (nama)
# I.S.: nama terdefinisi; F.S.: tercetak "Hello + nama" di layar
# KAMUS LOKAL
# ALGORITMA
    print ("Hello " + nama)
    return
```

```
# Program CetakKuadrat
# Input: masukan integer dan mencetak
# nilai kuadratnya ke layar

# KAMUS
# bil : int

def Kuadrat (x)
# menghasilkan kuadrat x
    return x * x

def CetakInt (x)
# I.S. x terdefinisi; F.S. x tercetak ke
# layar
    print (x)
    return

# ALGORITMA PROGRAM UTAMA
bil = int(input()) # baca dari keyboard
CetakInt(Kuadrat(bil))
```

# Contoh Pemanggilan Fungsi/Prosedur di Python

# Bahan Bacaan

- Diktat “Dasar Pemrograman Bagian Pemrograman Prosedural” → dengan notasi algoritmik
- Slides “Kumpulan Materi Dasar Pemrograman dengan Python 3” – Bahan Kuliah KU1102/Pengenalan Komputasi