



Ekspresi Bernama dan Kondisional

Dasar Pemrograman
Program Studi S1 Informatika
Universitas Diponegoro
Semester Gasal 2020/2021

Outline



Ekspresi Bernama

Ekspresi Kondisional: if ... else ..

Ekspresi Kondisional: depend ... on ...

Ekspresi Bernama

- Ekspresi yang dituliskan untuk sementara di dalam fungsi dan diberi nama (→ disebut juga **nama antara**).
- Nama tersebut **bersifat lokal** (hanya berlaku di dalam fungsi,) tidak dapat digunakan di luar fungsi.
- Nama yang diberikan harus berbeda dengan nama fungsi.
- Tipe untuk nama antara tersebut tidak perlu dinyatakan secara eksplisit.
- Tipe dapat direduksi dari operator yang digunakan.
- Penggunaan nama antara menggunakan notasi **let .. in**.

Bentuk Umum

```
<NAMA-FUNGSI> :  
    let <Nama> = <Ekspresi> in  
        <Realisasi-Fungsi>
```

- Nama-fungsi: nama fungsi yang direalisasi
- Nama: nama antara yang bersifat lokal, untuk menyimpan hasil evaluasi Ekspresi.
- Ekspresi: ekspresi fungsional yang hasilnya disimpan di Nama.
- Realisasi-Fungsi: ekspresi fungsional, realisasi dari Nama-Fungsi.

Bentuk Umum (2)

```
<NAMA-FUNGSI > :  
  let   <Nama-1> = <Ekspresi-1>,  
        <Nama-2> = <Ekspresi-2>,  
  ...   <Nama-k> = <Ekspresi-k> in  
        <Realisasi-Fungsi>
```

- Dapat didefinisikan beberapa nama antara dalam satu let.
- Masing-masing <Ekspresi-*i*> akan dievaluasi secara independen dalam lingkup <NAMA-FUNGSI> dan bukan sekuensial.

Contoh 1

- Menghindari evaluasi ekspresi berulang-ulang.
- Contoh: $(1 + a * b) * (1 - 2 * a * b)$
- Terdapat ekspresi $(a * b)$ sebanyak 2 kali.
- Digunakan `let` untuk menghindari ekspresi berulang.
- `p` disebut nama antara, hanya dapat digunakan dalam fungsi `F`.

```
F(a, b) :
  let   p = a * b in
        (1+p) * (1- 2*p)
```

Contoh 2a

- Menjadikan program lebih mudah dibaca.
- Contoh: pada fungsi $MO(u,v,w,x)$
- Digunakan nama antara S

REALISASI

```
MO (u, v, w, x) :  
    let S = u+v+w+x in  
    (S - min4(u, v, w, x) - max4(u, v, w, x)) / 2
```


Contoh 2b

- Menjadikan program lebih mudah dibaca.
- Contoh: pada fungsi $MO(u,v,w,x)$
- Digunakan nama antara S , M , m
- Penggunaan nama M dan m sebaiknya dihindarkan (karena beberapa bahasa menganggap sama (tidak case-sensitive))

REALISASI

```
MO (u, v, w, x) :  
  let S = u+v+w+x  
      M = max2 (max2 (max2 (u, v), w), x)  
      m = min2 (min2 (min2 (u, v), w), x)  
  in (S - m - M) / 2
```


Let yang mengandung let

```
<Nama-Fungsi>  
  let Nama-1 = Ekspresi-1 in  
    let Nama-2 = Ekspresi-2 in  
      <Realisasi-Fungsi>
```

- Dalam hal ini konteks/ scope harus diperhatikan.
- Jangan gunakan nama yang membingungkan (misalnya: nama antara sama dengan nama parameter formal fungsi atau sama dengan nama antara yang lain).

Contoh3: let yang mengandung let

```
F (x, y) :  
  let x = 3 + 4 * 5 in  
    let y = x + 5 in  
      x + y
```

- Contoh tersebut membingungkan antara x dan y.
- Sebaiknya nama antara (lokal) tidak sama dengan nama parameter formal.

Ekspresi Bernama Vs Fungsi Antara

Apa perbedaan antara Ekspresi Bernama dengan Fungsi Antara?

Ekspresi Kondisional

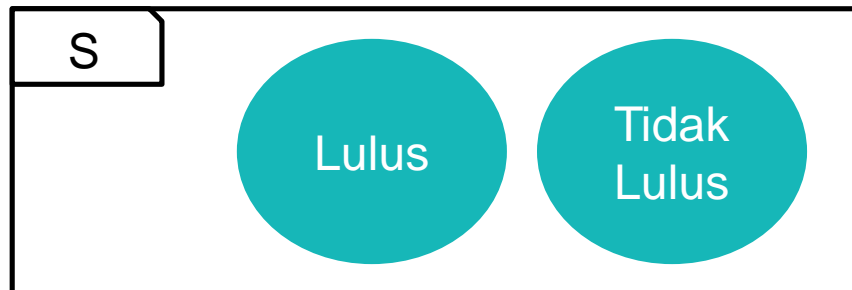
- Ekspresi kondisional adalah suatu ekspresi yang hasil evaluasinya tergantung pada hasil evaluasi beberapa kondisi.
- Ekspresi kondisional ditulis dengan melakukan analisa kasus.
- Analisa kasus → dekomposisi dari satu persoalan menjadi beberapa sub-persoalan yang ingin dipecahkan secara independen.
- Pada istilah pemrograman kasus disebut juga kondisi yang jika dievaluasi menghasilkan nilai Boolean.

Contoh Analisa Kasus

- Nilai akhir mahasiswa dapat memiliki rentang antara 0-100.
- Mahasiswa dinyatakan lulus atau tidak bergantung pada nilai akhirnya.

Contoh hasil analisa kasus:

- Jika nilai akhir ≥ 60 maka mahasiswa lulus.
- Jika nilai akhir < 60 maka mahasiswa tidak lulus.



Menentukan Kasus

- Setiap kasus harus disjoint (terpisah satu sama lain, tidak saling beririsan).
- Analisa kasus harus mencakup semua kasus yang mungkin.
- Kesalahan tipikal:
 - Ada kasus yang tidak tertulis.
 - Tidak disjoint.

Contoh Analisa Kasus yang Salah

Contoh hasil analisa kasus yang tidak lengkap:

- Jika nilai akhir ≥ 60 maka mahasiswa lulus.

Pertanyaan:

- Jika seorang mahasiswa mendapat nilai akhir 50, apakah dia lulus?

Contoh Analisa Kasus yang Salah

Contoh hasil analisa kasus yang beririsan:

- Jika nilai akhir ≥ 60 maka mahasiswa lulus.
- Jika nilai akhir ≤ 60 maka mahasiswa tidak lulus.

Pertanyaan:

- Jika seorang mahasiswa mendapat nilai akhir 60, apakah dia lulus?

Notasi Ekspresi Kondisional

Ada 2 macam

- if ... then ... Else
 - Bentuk ini digunakan untuk kasus yang komplementer (hanya ada 2 kondisi saja yang saling berlawanan)
- depend on ..
 - Bentuk ini digunakan jika terdapat lebih dari 2 kondisi yang akan dievaluasi.

Notasi depend on

depend on {deskripsi domain}

<Kondisi-1> : <Ekspresi-1>

<Kondisi-2> : <Ekspresi-2>

<Kondisi-3> : <Ekspresi-3>

Menggunakan else, artinya negasi selainnya.

depend on {deskripsi domain}

<Kondisi-1> : <Ekspresi-1>

<Kondisi-2> : <Ekspresi-2>

<Kondisi-3> : <Ekspresi-3>

else : *<Ekspresi-4>*

Notasi if ... then ... else ...

```
If <Kondisi-1> then  
    <Ekspresi-1>  
else  
    <Ekspresi-2>
```

Ekuivalen dengan:

```
depend on {deskripsi domain}  
    <Kondisi-1>      : <Ekspresi-1>  
not <Kondisi-1> : <Ekspresi-2>
```

Contoh Ekspresi Kondisional

Menentukan nilai maksimum dari dua buah nilai

MAKSIMUM 2 NILAI	max2(a,b)
<u>DEFINISI DAN SPESIFIKASI</u>	
$\text{max2} : 2 \text{ integer} \rightarrow \text{integer}$ <i>(max2 (a,b) menghasilkan maksimum dari 2 bilangan integer a dan b)</i>	
<u>REALISASI</u>	
{ Notasi if then else sebab hanya ada dua kasus komplementer... }	
$\text{max2} \ (a,b) :$ <u>if</u> $a \geq b$ <u>then</u> a <u>else</u> b	



Thank you

Selamat Belajar dan Berlatih !!!