Fungsi dan Ekspresi dalam Haskell

Tim Pengajar
IF1210 Dasar Pemrograman
Update: 10 Februari 2015

Keterangan

- Seluruh kode ditulis pada file **<nama-file>.hs**
- Gunakan **:load <nama-file>** untuk me-load script
- Setelah di-load, fungsi pada script dapat diaplikasikan

Template Notasi Fungsional

JUDUL **Nama-Fungsi (list-parameter-formal)**

DEFINISI DAN SPESIFIKASI

Nama-Fungsi : domain → range

{Tuliskan spesifikasi fungsi dengan nama, domain, dan range yang disebutkan di atas.}

REALISASI

Nama-Fungsi (list-parameter) : <ekspresi-fungsional>

APLIKASI

⇒Nama-Fungsi (list-parameter-aktual)

⇒Nama-Fungsi (list-parameter-aktual)

⇒Nama-Fungsi (list-parameter-aktual)

Template dalam Bahasa Haskell (dalam script <nama-file>.hs)

```
-- Judul          Nama-Fungsi (list-parameter-formal)

-- Definisi dan Spesifikasi
<Nama-Fungsi> :: <domain> → <range>
     -- Tuliskan spesifikasi fungsi dengan nama, domain,
     -- dan range yang disebutkan di atas.


-- Realisasi

<Nama-Fungsi> <list parameter> = <ekspresi-fungsional>


-- Contoh aplikasi

-- <Nama-Fungsi> <list-parameter>
```

Contoh (1) pangkat.hs

```
-- Pangkat2 - fx2(x)
-- Definisi dan Spesifikasi
fx2 :: Int -> Int
     -- fx2(x) menghitung pangkat dua dari x,
     -- sebuah bilangan integer
-- Realisasi
fx2 x = x * x

-- Contoh aplikasi
-- fx2 2   ATAU  fx2(2)
```

Contoh (2) (tambahkan pada pangkat.hs)

```
-- Pangkat3 - fx3(x)
-- Definisi dan Spesifikasi
fx3 :: Int -> Int
     -- fx3(x) menghitung pangkat tiga dari x,
     -- sebuah bilangan integer
-- Realisasi
fx3 x = (fx2 x) * x

-- Contoh aplikasi
-- fx3 5    ATAU   fx3(5)
```

Fungsi dengan lebih dari 1 parameter (masukan dianggap sebagai tuple)

```
-- PangkatN - fN(x,y)
-- Definisi dan Spesifikasi
fN :: (Int, Int) -> Int
     -- fN(x,y) menghitung pangkat x pangkat y,
     -- x dan y bilangan integer
-- Realisasi
fN(x,y) = x^y

-- Contoh aplikasi
-- fN(2,3)
```

Fungsi dengan lebih dari 1 parameter – versi 2

```
-- PangkatN - fNv2(x,y)
-- Definisi dan Spesifikasi
fNv2 :: Int -> Int -> Int
     -- fNv2(x,y) menghitung pangkat x pangkat y,
     -- x dan y bilangan integer
-- Realisasi
fNv2 x y = x^y


-- Contoh aplikasi
-- fNv2 2 3
```

IsOrigin? (x,y)

```
-- isOrigin - isOrigin(x,y)

-- Definisi dan Spesifikasi

isOrigin :: Int -> Int -> Bool

     -- isOrigin(x,y) bernilai true jika x=0 dan y=0

-- Realisasi

isOrigin x y = (x==0) && (y==0)


-- Contoh aplikasi

-- isOrigin(2,3)
```

Latihan 0

Buatlah fungsi-fungsi berikut seperti yang telah diberikan di kelas:

1. isPositif(x) bernilai true jika x >= 0
2. isAnA(c) bernilai true jika c='A'
3. isValid(x) bernilai true jika x < 5 atau x > 500

Jawaban

```
-- POSITIF                                          isPositif(x)

-- DEFINISI DAN SPESIFIKASI

isPositif :: Int -> Bool

  -- IsPositif(x) benar jika x positif

-- REALISASI

isPositif x = x >= 0

-- APLIKASI

-- isPositif 1
```

Jawaban

```
-- APAKAH HURUF A                                isAnA(c)

-- DEFINISI DAN SPESIFIKASI

isAnA :: Char -> Bool

-- isAnA (c) benar jika c adalah karakter (huruf) 'A'

-- REALISASI

isAnA c  =  c == 'A' || c == 'a'

-- APLIKASI

-- isAnA 'A'
```

Jawaban

```
-- APAKAH VALID                                    isValid(x)

-- DEFINISI DAN SPESIFIKASI

isValid :: Int -> Bool

-- IsValid (x) benar jika x bernilai lebih kecil dari 5

-- atau lebih besar dari 500

-- REALISASI

isValid x =  x < 5 || x > 500

-- APLIKASI

-- isValid 0
```

Ekspresi Bernama dengan LET Contoh (1)

```
tambah2x :: Int -> Int

-- tambah2x(x) menghasilkan x + 2

tambah2x (x) =

   let a = 2

   in

       (x + a)
```

Ekspresi Bernama dengan LET (Contoh 2)

```
luasLingkaran :: Float -> Float
-- luasLingkaran(r) menghasilkan luas lingkaran
-- dengan rumus = pi * r * r
luasLingkaran (r) =
   let pi = 3.14
   in
       pi*r*r
```

Ekspresi Bernama dengan LET (2)
Mean Olympique (V1)

```
-- Mean Olympique                                    mo(a,b,c,d)
-- DEFINISI DAN SPESIFIKASI
mo :: Float -> Float -> Float -> Float -> Float
  {-  mo(a,b,c,d) menghasilkan harga rata-rata dari dua di
      antara a, b, c, d, dengan mengabaikan nilai terbesar dan
      nilai  terkecil  -}
-- REALISASI -- versi tanpa "abstraksi"
mo a b c d =
     let maxab = (a+b + abs(a-b))/2
         maxcd = (c+d + abs(c-d))/2
         minab = (a+b - abs(a-b))/2
         mincd = (c+d - abs(c-d))/2
     in
          let maks = (maxab+maxcd + abs(maxab-maxcd))/2
              min = (minab+mincd - abs(minab-mincd))/2
          in
               (a+b+c+d-maks-min)/2
```

Ekspresi Bernama dengan LET (2)
Mean Olympique (V2) - 1

```
-- Mean Olympique                                    mo(a,b,c,d)
-- DEFINISI DAN SPESIFIKASI
mo2 :: Float -> Float -> Float -> Float -> Float
-- mo2(a,b,c,d) menghasilkan harga rata-rata dari dua di
-- antara a, b, c, d, dengan mengabaikan nilai terbesar dan
-- nilai  terkecil
max4 :: Float -> Float -> Float -> Float  -> Float
-- max4(a,b,c,d) menghasilkan maksimum dari a, b, c, d
min4 :: Float -> Float -> Float -> Float  -> Float
-- min4(a,b,c,d) menghasilkan minimum dari a, b, c, d
max2 :: Float -> Float -> Float
-- max2(a,b) menghasilkan maksimum dari a dan b
min2 :: Float -> Float -> Float
-- min2(a,b) menghasilkan minimum dari a dan b
...
```

Ekspresi Bernama dengan LET (2)
Mean Olympique (V2) - 2

```
...

-- REALISASI { versi dengan "abstraksi" }
max2 a b = (a+b+abs(a-b))/2
min2 a b = (a+b-abs(a-b))/2
max4 a b c d = max2 (max2 a b) (max2 c d)
min4 a b c d = min2 (min2 a b) (min2 c d)
mo2 a b c d = (a+b+c+d - (max4 a b c d) - (min4 a b c d))/2
```

Ekspresi Bernama dengan LET (2)
Mean Olympique (V3) - 1

```
-- Mean Olympique                                    mo(a,b,c,d)
-- DEFINISI DAN SPESIFIKASI
mo2 :: Int -> Int -> Int -> Int -> Float
{-  mo2(a,b,c,d) menghasilkan harga rata-rata dari dua di
    antara a, b, c, d, dengan mengabaikan nilai terbesar dan
    nilai  terkecil  -}
max4 :: Int -> Int -> Int -> Int -> Int
-- max4(a,b,c,d) menghasilkan maksimum dari a, b, c, d
min4 :: Int -> Int -> Int -> Int -> Int
-- min4(a,b,c,d) menghasilkan minimum dari a, b, c, d
max2 :: Int -> Int -> Int
-- max2(a,b) menghasilkan maksimum dari a, b
min2 :: Int -> Int -> Int
-- min2(a,b) menghasilkan minimum dari a, b
...
```

Ekspresi Bernama dengan LET (2)
Mean Olympique (V3) - 2

```
...

-- REALISASI  {versi dengan "abstraksi"}
max2 a b = div (a+b+abs(a-b)) 2
min2 a b = div (a+b-abs(a-b)) 2
max4 a b c d = max2 (max2 a b) (max2 c d)
min4 a b c d = min2 (min2 a b) (min2 c d)
mo2 a b c d = fromIntegral (a+b+c+d- (max4 a b c d)-
                                    (min4 a b c d))/2
```

Ekspresi Kondisional – (if-then-else)

```
-- Maksimum 2 bilangan                          max2(x,y)

-- Definisi dan Spesifikasi

max2 :: Int -> Int  -> Int

   -- max2(x,y) menerima dua nilai integer x dan y

   -- menghasilkan x jika x >= y; atau y jika x <= y

-- Realisasi

max2 x y = if (x >= y) then x else y
```

Ekspresi Kondisional – (depend on)

```
-- Maksimum 2 bilangan                          max2(a,b)

-- Definisi dan Spesifikasi

max2 :: Int -> Int  -> Int

   -- max2(a,b) menerima dua nilai integer a dan b

   -- menghasilkan a jika a >= b; atau b jika a <= b

-- Realisasi

max2 a b  | a >= b    = a

          | otherwise = b
```

Ekspresi Kondisional – (depend on)

```
-- Maksimum 3 bilangan                          max3(a,b,c)

-- Definisi dan Spesifikasi

max3 :: Int -> Int -> Int -> Int

   -- max3(a,b,c) menghasilkan nilai terbesar di

   -- antara 3 integer a, b, dan c

-- Realisasi

max3 a b c  | (a>=b && a>=c) = a

            | (b>=a && b>=c) = b

            | (c>=a && c>=b) = c
```

Perkenalan
Ekspresi Rekursif – Faktorial (v1)

```
-- Faktorial2 (definisi-2)                          fac2(n)

-- DEFINISI DAN SPESIFIKASI
fac2 :: Int -> Int
  {- fac2(n) = n! sesuai dengan definisi rekursif
     factorial, dengan basis 1 -}

-- REALISASI menggunakan if-then-else
fac2 n = if n == 1 then     -- Basis-1
             1
         else               -- Rekurens
             fac2 (n-1) * n
```

Perkenalan
Ekspresi Rekursif – Faktorial (v2)

```
-- Faktorial2 (definisi-2)                          fac2(n)

-- DEFINISI DAN SPESIFIKASI
fac2 :: Int -> Int
  {- fac2(n) = n! sesuai dengan definisi rekursif
     factorial, dengan basis 1 -}

-- REALISASI menggunakan if-then-else
fac2 n = if n == 1 then     -- Basis-1
             1
         else               -- Rekurens
             fac2 (n-1) * n
```

LATIHAN MANDIRI

Buatlah latihan-latihan berikut ini dan submitlah hasilnya di Oddysseus

Latihan 1

- Buatlah realisasi dari fungsi di bawah ini berdasarkan definisi dan spesifikasi yang diberikan.

```
-- APAKAH JAM VALID?                    isJamValid(j,m,d)
-- DEFINISI DAN SPESIFIKASI
isJamValid :: Int -> Int -> Int -> Bool
  {- IsJamValid(j,m,d) menghasilkan nilai true jika
     j, m, d menyusun jam yang valid. Definisi jam yang
     valid adalah jika elemen jam (j) bernilai antara 0
     dan 23, elemen menit (m) bernilai antara 0 dan 59,
     dan elemen detik (d) bernilai antara 0 dan 59. -}
```

Latihan 2

- Diberikan 3 buah integer j, m, d dengan j adalah integer [0..23], m adalah integer [0..59], d adalah integer [0..59], yang artinya adalah jam (j), menit (m), dan detik (d) pada suatu tanggal tertentu.
- Hitunglah jumlah detik dari jam tersebut terhitung mulai jam 0:0:0 pada tanggal ybs.

Latihan 3

- Buatlah realisasi dari fungsi di bawah ini berdasarkan definisi dan spesifikasi yang diberikan. Jika perlu membuat fungsi antara, buatlah definisi, spesifikasi dan realisasinya.

```
-- APAKAH DATE VALID?          isDateValid(d,m,y)

-- DEFINISI DAN SPESIFIKASI
isDateValid ::  Int -> Int -> Int -> Bool
{- isDateValid(d,m,y) mengembalikan nilai true jika
   d, m, y membentuk date yang valid. Definisi date yang
   valid adalah jika elemen hari (d) bernilai antara 1
   dan 31, tergantung pada bulan dan apakah tahun kabisat
   atau bukan, elemen bulan (m) bernilai antara 1 dan 12,
   dan elemen tahun (y) bernilai lebih dari antara 0 dan
   99 -}
```

Latihan 4

Buatlah definisi, spesifikasi, dan realisasi dari fungsi **NilaiTengah** yang menerima masukan 3 buah integer yang berlainan nilainya yang urutannya bisa acak dan mengembalikan sebuah integer yang merupakan salah satu dari ke-3 nilai tersebut yang jika diurutkan berada di tengah.

Contoh aplikasi fungsi:

```
*Main> nilaiTengah 1 2 3
2
*Main> nilaiTengah (-6) 1 5
5
*Main> nilaiTengah (-1) (-4) 10
-1
```

Petunjuk: Buatlah fungsi min3 dan max3 dalam realisasinya.

Latihan 5

- Mata uang US adalah *dollar*. 1 dollar = 100 sen. Dalam mata uang di US dikenal beberapa jenis koin yang masing-masing diberi nama yaitu *quarter* (1 quarter = 25 sen = 0,25 dollar), *dime* (1 dime = 10 sen = 0,1 dollar), *nickel* (1 nickel = 5 sen = 0,05 dollar), dan *penny* (1 penny = 1 sen = 0,01 dollar). Buatlah sebuah program yang menerima input sejumlah koin quarter, dime, nickel, dan penny dan menghasilkan berapa dollar dan berapa sen yang senilai dengan total koin-koin tersebut.
- Contoh: 8 quarter, 20 dime, 30 nickel, dan 77 penny adalah sama dengan 6 dollar dan 27 sen.
- Perhatikan bahwa output yang dihasilkan adalah pasangan nilai <dollar, sen>.