



Tugas 3

Tags

Identitas

- Nama: Fahdii Ajmalal Fikrie
- NPM: 1906398370

Soal 1

Buatlah fungsi myCurry dan myUncurry yang memiliki sifat seperti fungsi curry dan uncurry pada library Haskell

```
-- No. 1 – myCurry & myUncurry
myCurry :: ((a, b) -> c) -> a -> b -> c
myCurry f = \x y -> f (x, y)

myUncurry :: (a -> b -> c) -> ((a, b) -> c)
myUncurry f = \ (x,y) -> f x y
```

Soal 2

Deret bilangan fibonacci adalah serangkaian deret angka sederhana yang susunan angkanya merupakan penjumlahan dari dua angka sebelumnya. Buatlah fungsi fibonacci yang menerima bilangan bulat positif n dan mengembalikan list yang berisi bilangan fibonacci dari 0 sampai n

```
-- No. 2 – Fibonacci
fibonacci :: Integer -> Integer
fibonacci 0 = 1
fibonacci 1 = 0
fibonacci n = fibonacci(n-1) + fibonacci(n-2)

fibonaccis :: Integer -> [Integer]
fibonaccis n = map fibonacci [1..n]
```

Soal 3

Buatlah fungsi power namun hanya dengan menggunakan operasi penjumlahan (+)

```
-- No. 3 – Power function w/ only addition
multiply :: Integer -> Integer -> Integer
multiply a 0 = 0
multiply a 1 = a
multiply a b = a + multiply a (b-1)

power :: Integer -> Integer -> Integer
power a 0 = 1
power a 1 = a
power a b = multiply a (power a (b-1))
```

Soal 4

Buatlah fungsi sumEven yang menerima list dari bilangan asli dan menjumlahkan bilangan-bilangan genap saja.

```
-- No. 4 – sumEven
-- Kalo tipenya Real malah compile error :(
sumEven :: [Integer] -> Integer
sumEven [] = 0
sumEven (x:xs)
  | (x `mod` 2 == 0) = x + sumEven xs
  | otherwise      = 0 + sumEven xs
```

Soal 5

Buatlah sebuah kalkulator investasi sederhana berupa fungsi invest yang menerima 3 buah parameter, yaitu:

- nominal tiap bulan
- return investasi tiap bulan (%)
- durasi (bulan)

```
-- No. 5 – Kalkulator Investasi
invest :: Double -> Double -> Double -> Double
invest nom inv dur = interest nom inv dur

interest :: Double -> Double -> Double -> Double
```

```
interest nom inv 0 = nom
interest nom inv 1 = interest (nom * (1+inv)) inv 0
interest nom inv dur = nom + interest (nom * (1+inv)) inv (dur-1)
```

Refleksi



Untuk materi minggu ini, saya pribadi masih belum paham betul materi `curry` dan `uncurry` (terlepas dari definisinya saja). Saya juga masih belum berhasil mengimplementasikan `fibonacci` & `power function w/ addition` tanpa helper method (meskipun yang saya baca dari lecture notes upenn kalau haskell semakin banyak mengcombine fungsi—yang kecil dan hanya melakukan satu hal/fungsi—semakin baik.