Tugas 1

Tags

Identitas

- Nama: Fahdii Ajmalal Fikrie
- NPM: 1906398370

Latihan Tugas 1

Pada setiap nomor, bacalah permintaan soal dan contoh input fungsi tersebut. Implementasikan fungsi tersebut dan carilah sumber-sumber yang dapat membantu dalam pengerjaan anda. Pada akhir tugas, tulislah apa yang anda dapatkan setelah mengerjakan soal-soal berikut.

Soal no. 1

Buatlah fungsi circleArea yaitu fungsi yang menerima parameter r Double dan mengembalikan luas dari lingkaran dengan jari-jari r dengan pi 3.14.

```
circleArea :: Floating r => (r) -> r
circleArea (r) = pi * r * r

main = do
    putStrLn "Insert the radius of the circle: "
    r <- readLn
    print $ circleArea (r)
```

Soal no. 2

Buatlah fungsi isTriangle yang menerima 3 buah parameter bertipe Integer yang merupakan sisi-sisi segitiga dan mengembalikan Boolean yang menyatakan apakah segitiga dapat dibuat berdasarkan 3 buah sisi tersebut.

```
{-
  Reference:
      > https://www.geeksforgeeks.org/check-whether-triangle-valid-not-sides-given/
-}

isTriangle :: Int -> Int -> Int -> [Char]
isTriangle s1 s2 s3
    | ((s1 + s2) <= s3) || ((s2 + s3) <= s1) || ((s3 + s1) <= s2)
```

```
        = "The 3 sides given is NOT a valid triangle"
    | otherwise
        = "The 3 sides given is a valid triangle"

main:: IO()
main = do
    putStrLn "Insert the 1st side of the triangle: "
    s1 <- readLn
    putStrLn "Insert the 2nd side of the triangle: "
    s2 <- readLn
    putStrLn "Insert the 3rd side of the triangle: "
    s3 <- readLn
    print $ isTriangle s1 s2 s3
```

Soal no. 3

Buatlah fungsi listSum yang menerima parameter List Integer dan mengembalikan total dari elemen didalam list tersebut.

```
{-
  References:
      > https://stackoverflow.com/questions/34817219/haskell-recursive-list-summation
      > https://stackoverflow.com/questions/43718678/read-strings-from-keyboard-in-haskell
      > https://stackoverflow.com/questions/8879391/how-do-i-convert-string-into-list-of-integers-in-haskell
-}

listSum :: [Int] -> Int
listSum [] = 0
listSum (x:t) = x + listSum t

seqOfStringToInt :: String -> [Int]
seqOfStringToInt seq = map read $ words seq

main:: IO()
main = do
    putStrLn "Insert the integers you want to sum (separated by whitespace): "
    seq <- getLine
    let listOfInt = seqOfStringToInt seq
    print $ listSum listOfInt
```

Soal no. 4

Buatlah fungsi listSumArea yang menerima parameter List circleArea dan mengembalikan total dari semua area dididalam list tersebut.

```
{-
  References:
      > https://stackoverflow.com/questions/34817219/haskell-recursive-list-summation
      > https://stackoverflow.com/questions/43718678/read-strings-from-keyboard-in-haskell
      > https://stackoverflow.com/questions/8879391/how-do-i-convert-string-into-list-of-integers-in-haskell
-}

listCircleAreasSum :: [Int] -> Int
listCircleAreasSum [] = 0
listCircleAreasSum (x:t) = x + listSum t
```

```
seqOfStringToInt :: String -> [Int]
seqOfStringToInt seq = map read $ words seq

main:: IO()
main = do
    putStrLn "Insert the circle Areas you want to sum (separated by whitespace): "
    seq <- getLine
    let listOfCircleAreas = seqOfStringToInt seq
    print $ listCircleAreasSum listOfCircleAreas
```

Soal no. 5

Buatlah implementasi reverseList.

```
{-
    References:
        > https://stackoverflow.com/questions/5952167/how-do-i-print-a-list-in-haskell
        > https://stackoverflow.com/questions/26847192/reverse-a-list-in-haskell/26847373
        > https://stackoverflow.com/questions/40934263/haskell-generic-type-to-generic-and-int
-}

-- Masih bingung bikin genericnya gimana

-- data IList =
--     I Int
--     | S String
--     | E String
--     deriving Show
-- reverseList :: [IList] -> [IList]

reverseList :: [Int] -> [Int]
reverseList [] = []
reverseList (x:xs) = reverseList xs ++ [x]

seqOfStringToInt :: String -> [Int]
seqOfStringToInt seq = map read $ words seq

main :: IO()
main = do
    putStrLn "Insert Int list items you want to reverse (separated by whitespace): "
    seq <- getLine
    let listItems = seqOfStringToInt seq
    print $ reverseList listItems
```

Soal no. 6

Buatlah implementasi algoritma quicksort descending.

```
{-
    Reference:
        > https://gist.github.com/thurloat/1752010
-}

quickSort :: Ord a => [a] -> [a]
quickSort [] = []
```

```
quickSort (p:xs) = (quickSort greater) ++ [p] ++ (quickSort lesser)
    where
        lesser  = filter (< p) xs
        greater = filter (>= p) xs

seqOfStringToInt :: String -> [Int]
seqOfStringToInt seq = map read $ words seq

main :: IO()
main = do
    putStrLn "Insert Int list items you want to sort (separated by whitespace): "
    seq <- getLine
    let listItems = seqOfStringToInt seq
    print $ quickSort listItems
```

Refleksi

Setelah mengerjakan latihan, saya jadi mempelajari beberapa konsep di haskell, seperti:
- Functional typing
- IO
- Rekursif
- Guards

Sedangkan, yang masih membuat saya bingung adalah cara menggunakan konsep generic di OO dalam haskell.