

SOAL TUBAY UTS DASPRO

Soal 1. Pengantar Dunia Pemrograman (Akademik STEI-K)

- Jelaskan apa itu paradigma pemrograman dan sebutkan 5 contohnya!
- Sebutkan dan jelaskan secara singkat 5 taksonomi bahasa pemrograman!

Soal 2.1 Ekspresi Fungsional (UTS 2016/2017)

- Tuliskan **definisi**, **spesifikasi**, **realisasi**, dan contoh **aplikasi** (berikut hasilnya) dari fungsi **prevSecond** yang menerima masukan 3 integer (j, m, d) merepresentasikan sebuah waktu dan mengembalikan sebuah tuple yang terdiri dari 3 integer (j1,m1,d1) merepresentasikan waktu 1 detik sebelumnya. Realisasi fungsi tidak boleh menggunakan konversi ke detik. Prekondisi: j, m, d dapat membentuk waktu yang valid, yaitu j bernilai [0..23], m bernilai [0..59], dan d bernilai [0..59].
- Tuliskan realisasi dari fungsi **prevNSeconds** berikut ini (definisi dan spesifikasi tidak perlu ditulis ulang). Realisasi fungsi harus menggunakan ekspresi rekursif.

```
prevNSeconds :: Int -> Int -> Int -> Int -> (Int,Int,Int)
{- prevNSeconds j m d n menghasilkan tuple (j1,m1,d1) yang merupakan waktu n detik sebelum
   pukul j:m:d.
   Prekondisi: 0<=j<=23, 0<=m<=59, 0<=d<=59, n>=0 -}
```

Contoh

Masukan				Hasil Fungsi	
j	M	d	n	prevSecond j m d	prevNSeconds j m d n
11	11	11	10	(11,11,10)	(11,11,1)
15	10	10	15	(15,10,9)	(15,9,55)
0	0	0	3600	(23,59,59)	(23,0,0)

Anda tidak diperkenankan membuat dan menggunakan fungsi antara baru (tapi boleh menggunakan fungsi yang tersedia dalam soal ini).

Soal 2.2 Analisis Rekurens (Akademik STEI-K)

Fibonacci adalah suatu barisan bilangan yang merupakan hasil penjumlahan dua bilangan sebelumnya yang diperkenalkan pertama kali oleh Leonardo da Pisa atau yang lebih dikenal dengan Fibonacci pada abad ke 13. Pola bilangan Fibonacci bisa dituliskan sebagai berikut:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Buatlah fungsi fibonacci yang menerima masukan sebuah buah integer n dan mengembalikan hasil berupa nilai suku ke- n dari barisan fibonacci tersebut. Masukan diasumsikan selalu valid, yaitu $n \geq 1$. Buatlah **definisi**, **spesifikasi**, **realisasi** dengan menggunakan ekspresi rekursif dan contoh aplikasi berikut hasilnya (minimum 2 buah) dari fungsi fibonacci.

Soal 3. List (UTS 2016/2017)

Diketahui definisi primitif tipe **list of integer** berikut ini. Semua primitif list dapat langsung digunakan (tidak perlu dibuat realisasinya).

```
-- DEFINISI DAN SPESIFIKASI PRIMITIF LIST
-- KONSTRUKTOR
konso :: Int -> [Int] -> [Int]
{- konso e li menghasilkan sebuah list of integer dari e dan li
    dengan e sebagai elemen pertama -}
konsDot :: [Int] -> Int -> [Int]
{- konsDot li e menghasilkan sebuah list of integer dari li dan e
    dengan e sebagai elemen terakhir -}
-- SELEKTOR
-- head :: [Int] -> Int
{- head l menghasilkan elemen pertama list l, l tidak kosong -}
-- tail :: [Int] -> [Int]
{- tail l menghasilkan list tanpa elemen pertama list l, l tidak kosong -}
-- last :: [Int] -> Int
{- last l menghasilkan elemen terakhir list l, l tidak kosong -}
-- init :: [Int] -> [Int]
{- init l menghasilkan list tanpa elemen terakhir list l, l tidak kosong -}
-- PREDIKAT DASAR
isEmpty :: [Int] -> Bool
{- isEmpty l true jika list of elemen l kosong, false jika tidak -}
isOneElmt :: [Int] -> Bool
{- isOneElmt l true jika list of integer l hanya mempunyai satu elemen, false jika tidak -}
```

Tuliskan **realisasi** dari keempat fungsi berikut (def inisi dan spesifikasi fungsi tidak perlu ditulis ulang)

```
isSortedDown :: [Int] -> Bool
{- isSortedDown l mengembalikan True apabila l terurut mengecil.
    Prekondisi: list tidak kosong
    Contoh: isSortedDown [8,5,2,4] = False; isSortedDown [5,2,0] = True -}

getElTengah :: [Int] -> Int
{- getElTengah l mengembalikan elemen l yang berada di tengah-tengah.
    Apabila elemen l berjumlah genap (misalnya n), maka yang dikembalikan adalah elemen pada
    posisi (n div 2).
    Prekondisi: list tidak kosong
    Contoh: getElTengah [3,2,6,5,8] = 6; getElTengah [1,2,3,4] = 2 -}

elPosGanjil :: [Int] -> [Int]
{- elPosGanjil l mengembalikan sebuah list yang berisi semua elemen l pada posisi ganjil.
    Contoh: elPosGanjil [2,4,6,8,1,3,5,7,9] = [2,6,1,5,9] -}

delAllX :: [Int] -> Int -> ([Int],Int)
{- delAllX l x mengembalikan sebuah list l1 dan integer n, dengan l1 memuat elemen-elemen l
    setelah elemen bernilai x dihapus dan n memuat jumlah kemunculan elemen bernilai x pada l.
    Contoh: delAllX [3,2,6,2,3] 3 = ([2,6,2],2); delAllX [] 3 = ([],0) -}
```

Untuk soal 3, tidak diperkenankan membuat dan menggunakan fungsi antara baru (tapi boleh menggunakan fungsi-fungsi yang tersedia dalam soal ini).

Soal 4. Aspek Fungsi dengan Parameter Fungsi (UTS 2015/2016)

- a. Buatlah definisi, spesifikasi, dan realisasi sebuah fungsi **sumInteger** yang menerima 2 (dua) buah integer positif (>0), misalnya **m** dan **n**, dan sebuah fungsi **f** dan menghasilkan penjumlahan dari semua integer antara m dan n (termasuk m dan n) yang memenuhi f. Jika dalam selang m dan n tidak ada yang memenuhi f, maka hasilnya adalah 0.

Contoh:

Masukan			sumInteger m n f	Keterangan
m	n	f		
2	9	isGenap :: Int -> Bool {- isGenap x menghasilkan true jika x adalah bilangan genap -}	20	Yang memenuhi isGenap dalam selang [2..9] = {2,4,6,8} sehingga $2+4+6+8 = 20$
2	9	gtThan5 :: Int -> Bool {- gtThan5 x menghasilkan true jika $x > 5$ -}	30	Yang memenuhi gtThan5 dalam selang [2..9] = {6,7,8,9} sehingga $6+7+8+9 = 30$
8	5	isGenap :: Int -> Bool {- isGenap x menghasilkan true jika x adalah bilangan genap -}	0	Selang [8..5] tidak terdefinisi

- b. Tuliskan aplikasi fungsi **sumInteger** pada butir a untuk 3 (tiga) masukan di bawah ini dalam bentuk **ekspresi lambda** dan tuliskan pula hasil aplikasinya.

m	n	f
		Menerima masukan sebuah integer x dan ...
1	100	menghasilkan true jika x dapat membagi habis 100.
1	100	menghasilkan true jika x adalah bilangan genap dan dapat dibagi habis oleh 10 atau jika x adalah bilangan ganjil dan dapat dibagi habis oleh 5. Selain itu, menghasilkan false.
25	25	menghasilkan true, jika x lebih kecil dari 10.

Untuk soal 4, tidak diperkenankan membuat dan menggunakan type baru atau fungsi antara baru.