

## Soal 1. Pengantar Pemrograman

1. Sebutkan dan jelaskan ciri khas dari 3 (tiga) paradigma pemrograman yang Anda ketahui. Sebutkan pula masing-masing 1 (satu) contoh bahasa pemrograman yang mengimplementasikannya.
2. Jelaskan apa yang disebut sebagai bad programmer dan bagaimana cara agar Anda tidak menjadi seorang bad programmer.
3. Jelaskan mengapa membangun software diibaratkan seperti membangun pencakar langit sedangkan membangun program kecil seperti membangun rumah kecil.

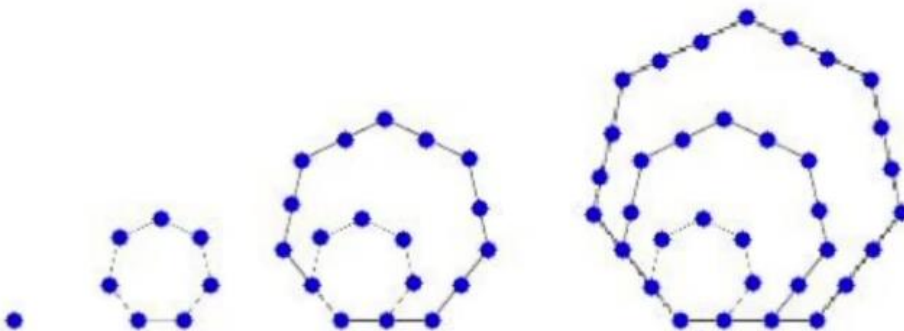
## Soal 2. Analisis Rekurens dan List

Menurut definisi, pola bilangan adalah suatu bentuk bilangan yang tersusun dengan bilangan lain yang membentuk suatu pola. Jika kita mengikuti terbentuknya suatu bidang 2D, bidang paling sederhana yang dapat membentuk bidang lainnya adalah segitiga, sehingga pada aritmatika kita mengenal deret-segitiga. Turunan dari deret-segitiga tersebut kita bisa membangun deret-segiempat, -segilima, -segienam, dan seterusnya.

Anda diminta untuk membuat sebuah fungsi bernama **fDeretTujuh** yang digunakan untuk menghasilkan urutan bilangan berpola bidang segitujuh. Secara matematika pola deret segitujuh didefinisikan sbb.

Pola Bilangan	Rumus	Deret bilangan ( $U_n$ )
Deret-segitujuh	$U_n = \frac{5n^2 - 3n}{2}$	1, 7, 18, 34, 55, 81, 112, ...

Berikut adalah ilustrasi elemen 1 s.d. 4 dari deret segitujuh.



Fungsi **fDeretTujuh** menerima masukan sebuah integer **m** lebih besar dari nol ( $m > 0$ ). Fungsi **fDeretTujuh** menghasilkan bilangan integer yang menyatakan elemen ke-**m** pada pola bilangan deret segitujuh.

### Soal a

Buatlah definisi, spesifikasi, dan realisasi dari fungsi **fDeretTujuh** dengan menggunakan **pendekatan rekursif** (wajib). **Dilarang** menggunakan ekspresi lambda atau fungsi sebagai parameter.

Contoh aplikasi fungsi fDeretTujuh:

```
> fDeretTujuh 5
```

```
55
```

```
> fDeretTujuh 4
```

```
34
```

### Soal b

Dengan memanfaatkan fungsi **fDeretTujuh** (tanpa modifikasi apapun), buatlah definisi, spesifikasi, dan realisasi fungsi **fDeretTujuhList** yang menerima input berupa dua buah bilangan integer **d** dan **e** (asumsikan  $d > 0$  dan  $e > 0$ ). Hasil dari fungsi **fDeretTujuhList** adalah sebuah list bilangan integer yang berisi elemen ke-d hingga ke-e dari deret segitujuh. Jika  $e < d$ , dihasilkan list kosong.

Contoh aplikasi fungsi fDeretTujuhList:

```
> fDeretTujuh 2 4
```

```
[7,18,34]
```

```
> fDeretTujuh 3 3
```

```
[18]
```

```
> fDeretTujuh 4 3
```

```
[]
```

Gunakan tipe dan operasi **list of integer** yang diberikan definisi dan spesifikasinya berikut ini. Semua fungsi primitif 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
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 -}

```

### Soal 3. List dan Operasinya

Diketahui definisi primitif tipe **list of integer** berikut ini. Semua fungsi dapat langsung digunakan (tidak perlu dibuat realisasinya, kecuali diminta pada soal).

```

-- 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
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 -}

```



## -- FUNGSI LAIN

**nbX :: [Int] -> Int -> Int**

*{- nbX l x menghasilkan banyaknya kemunculan x di l -}*

**isMember :: Int -> [Int] -> Bool**

*{- isMember x l menghasilkan true jika x adalah elemen l -}*

Untuk semua soal berikut, **tidak diperbolehkan menambahkan fungsi antara**, tetapi boleh menggunakan fungsi-fungsi lain yang tersedia di soal ini.

### Soal a

Pilihlah salah satu dari fungsi **nbX** atau **isMember** dan buatlah realisasinya sesuai definisi dan spesifikasi di atas (definisi dan spesifikasi fungsi tidak perlu ditulis ulang).

### Soal b

Tuliskan **realisasi** dari fungsi-fungsi berikut (definisi dan spesifikasi fungsi tidak perlu ditulis ulang).

**removeElmt :: [Int] -> Int -> [Int]**

*{- removeElmt l x menghasilkan list l tanpa elemen bernilai x paling awal/depan, jika x ada di l.*

**listCountN :: [Int] -> Int -> ([Int],Int)**

*{- listCountN l n menghasilkan pasangan nilai (l1,m) dengan l1 adalah list unik yang berisi elemen-elemen l yang muncul sebanyak n, dan m adalah banyaknya elemen l1 alias banyaknya elemen l yang muncul sebanyak n.*

*Prekondisi: n > 0 -}*

Contoh aplikasi fungsi dan hasilnya:

<b>l</b>	<b>x</b>	<b>n</b>	<b>removeElmt l x</b>	<b>listCountN l n</b>
[1,2,2,4,5,6,1,5,5,6,5,6]	6	2	[1,2,2,4,5,1,5,5,6,5,6]	([1,2],2)
[1,2,2,4,5,6,1,5,5,6,5,6]	7	5	[1,2,2,4,5,6,1,5,5,6,5,6]	([],0)
[7]	7	1	[]	([1],1)
[]	2	2	[]	([],0)

#### Soal 4. Ekspresi Lambda dan Fungsi dengan Parameter Fungsi

Diketahui fungsi **aturParkir** yang digunakan untuk menentukan slot (tempat) parkir mobil, yang menerima tiga masukan:

- **li**: sekumpulan (list) nomor polisi kendaraan (nomor polisi adalah integer  $> 0$ ),
- **n**: banyaknya slot parkir yang tersedia ( $n > 0$ ), dan
- **f**: fungsi penentuan slot parkir.

Fungsi **f** menerima masukan sebuah nomor polisi kendaraan (integer), dan mengeluarkan nomor slot parkir untuk kendaraan tersebut. Nomor slot parkir adalah integer dimulai dari 0 hingga  $n-1$ .

Fungsi **aturParkir** menghasilkan sebuah tuple yang terdiri atas dua nilai:

- List yang berisi nomor slot parkir yang digunakan, dan
- Banyaknya kendaraan dari list masukan yang tidak mendapat slot parkir.

Sebuah kendaraan tidak mendapat slot parkir jika hasil dari fungsi **f** untuk kendaraan tersebut:

- berupa nomor slot yang sudah digunakan oleh kendaraan lain (slot parkir diprioritaskan bagi nomor kendaraan yang lebih depan pada **li**), atau
- berupa nomor slot  $< 0$ , atau
- berupa nomor slot  $\geq n$ .

Contoh masukan dan keluaran dari fungsi **aturParkir**.

Masukan			Hasil aturParkir li n f	Keterangan
li	n	F		
[34, 123, 8265, 217, 856]	20	Menerima sebuah integer x dan mengembalikan sisa hasil pembagian x dengan 20.	[(14,3,5,17,16),0]	Semua mobil mendapatkan slot parkir mobil.
[4232, 654, 1712, 2310, 4124, 1102, 724]	30	Menerima sebuah integer x dan jika x genap maka mengembalikan nilai x div 400 dan kemudian ditambah 10; jika x ganjil maka mengembalikan nilai x div 55 dan ditambah 6.	[(20,11,14,10,12,19),1]	Nomor polisi 4124 dialokasikan ke slot parkir 20, namun slot sudah dipakai nomor polisi 4232, jadi ada 1 kendaraan yang tidak mendapat slot parkir. Catatan: perhatikan urutan pemrosesan elemen list li.

[ ]	77	Menerima sebuah integer x dan mengembalikan nilai x div 11.	([ ],0)	Tidak ada mobil yang perlu dialokasikan.
[101, 12345, 44, 567, 99, 808]	100	Menerima sebuah integer x dan jika x kurang dari atau sama dengan 100 maka mengembalikan nilai x dikurangi 1; jika x lebih dari 100 maka mengembalikan sisa hasil pembagian x dengan 100 lalu dikurangi 10.	([35,43,57,98],2)	Nomor polisi 101 tidak mendapat slot parkir karena hasil fungsi penentuan slot parkir bernilai < 0; demikian juga untuk nomor polisi 808.
[999]	50	Menerima sebuah integer x dan mengembalikan nilai x div 11.	([ ],1)	Nomor polisi 999 tidak mendapat slot parkir karena hasil fungsi penentuan slot parkir bernilai >= n.

### Soal a

Buatlah **definisi**, **spesifikasi**, dan **realisasi** dari fungsi **aturParkir**. Fungsi **aturParkir** memanfaatkan fungsi **isMember** (definisi dan spesifikasi fungsi **isMember** yang disediakan di bawah, tidak perlu direalisasikan dan tinggal digunakan). Untuk soal ini **tidak diperbolehkan menambahkan fungsi antara**, tetapi boleh menggunakan fungsi-fungsi lain yang tersedia di soal ini.

### Soal b

Buatlah **contoh aplikasi fungsi aturParkir** dengan masukan seperti pada tabel berikut. Gunakan **ekspresi lambda** pada pemanggilan fungsi **aturParkir** tersebut, dan **tuliskan juga hasilnya**.

li	n	f
[52, 2, 11, 262, 107, 33]	26	Menerima sebuah integer x dan mengembalikan sisa hasil pembagian (x - 3) dengan 26.
[225, 24, 400, 666, 345]	15	Menerima sebuah integer x dan jika x habis dibagi 4 tetapi tidak habis dibagi 10 maka menembalikan nilai x - 8; jika x habis dibagi 40 maka mengembalikan nilai x div 40 lalu ditambah 7; selain itu mengembalikan nilai sisa hasil pembagian x dengan 15.

Gunakan type list dan operasinya sesuai yang diajarkan di kelas. Berikut definisi primitif tipe **list of integer** (termasuk fungsi **isMember**) untuk digunakan dalam soal 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  
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 -}  
  
-- Fungsi Lain  
isMember :: Int -> [Int] -> Bool  
{- isMember x l menghasilkan true jika x adalah elemen l -}
```

---