



Responsi - 1

Oleh: Tim Asisten IF1210 Dasar Pemrograman 2021-2






Agenda Responsi

1. Penjelasan dari Asisten
2. Mengerjakan soal praktikum yang sama pada olympia yang berdurasi 2 jam.


Kerjakan saja soal yang kalian dapat minggu lalu atau bebas pilih 1 soal 1 opsi, yang lainnya kosongkan (*Maks nilai 600 + 100 dari bonus*).





Disclaimer

Sangat disarankan untuk tidak melakukan *copy-paste* untuk menjawab responsi. Karena jawaban bisa beragam untuk beberapa pertanyaan. **Jadikan SLIDE ini referensi ide kalian untuk menjawab soal dan bahan belajar.** Supaya kalian bisa paham dan mengerjakan praktikum-praktikum berikutnya yang mungkin saja tingkat kesulitannya bertambah dari praktikum ini. Penyalahgunaan **SLIDE Responsi** merupakan tanggung jawab masing-masing.





Soal 1



Deret Aritmatika

Opsi Jawaban

Nama File : DeretAritmatika.hs

Nama modul : DeretAritmatika

Buatlah definisi, spesifikasi, dan realisasi fungsi

deretAritmatika yang menerima masukan 3 nilai integer, yaitu n , a , dan b . Parameter a merupakan suku pertama dari suatu barisan aritmatika yang memiliki beda b .

Realisasi fungsi **deretAritmatika** akan menghasilkan jumlah n suku pertama dari barisan aritmatika tersebut.

Contoh:

```
> deretAritmatika 4 4 5  
46.0
```

Keterangan:

$$S_n = (n(2a + (n-1)b))/2$$

```
module DeretAritmatika where
```

```
-- DEFINISI DAN SPESIFIKASI
```

```
deretAritmatika :: Int -> Int -> Int -> Float  
{- Menghitung jumlah n suku pertama dari deret  
aritmatika dengan suku pertama a dan beda b-}
```

```
-- REALISASI
```

```
deretAritmatika n a b = fromIntegral((n * (2 * a +  
(n - 1) * b))) / fromIntegral(2)
```

```
-- Aplikasi
```

```
-- deretAritmatika 4 4 5
```

Luas Layang-Layang

Jawaban

module LuasLayang **where**

-- DEFINISI DAN SPESIFIKASI

layang :: Int -> Int -> Float

{- Menghitung luas layang-layang berdasarkan diagonal 1 dan diagonal 2 -}

-- REALISASI

layang d1 d2 = 0.5 * fromIntegral (d1 * d2)

-- APLIKASI

-- layang 3 7

Nama file : LuasLayang.hs

Nama modul : LuasLayang

Buatlah definisi, spesifikasi, dan realisasi fungsi **layang** yang menerima masukan 2 nilai integer, yaitu kedua diagonal sebuah layang-layang misalnya d1 dan d2.

Realisasi fungsi layang akan menghasilkan luas layang-layang tersebut dengan rumus: $\frac{1}{2} * d1 * d2$

Contoh:

```
> layang 3 7
10.5
```

Luas Lingkaran

Nama File: LuasLingkaran.hs

Header: module LuasLingkaran where

Buatlah definisi, spesifikasi, dan realisasi fungsi **luasLingkaran** yang menerima masukan sebuah bilangan real (float) r yang merepresentasikan jari-jari sebuah lingkaran (asumsikan $r > 0$) dan menghasilkan luas lingkaran berdasarkan rumus: $luas = 3.1415 * r * r$.

Contoh aplikasi dan hasil:

No	Aplikasi	Hasil	Keterangan
1.	luasLingkaran 2	12.566	$3.1415 * 2 * 2$
2.	luasLingkaran 3.5	38.483376	$3.1415 * 3.5 * 3.5$

Jawaban

module LuasLingkaran **where**

-- DEFINISI DAN SPESIFIKASI

luasLingkaran :: **Float** -> **Float**

{- Menghitung luas lingkaran berdasarkan input jari-jari (r) -}

-- REALISASI

luasLingkaran r = **3.145** * r * r

-- APLIKASI

-- luasLingkaran 7 7

Luas Trapesium

Nama File: LuasTrapesium.hs

Header: module LuasTrapesium where

Buatlah definisi, spesifikasi, dan realisasi fungsi **luasTrapesium** yang menerima masukan 3 buah bilangan real (float) t , $s1$, $s2$ dengan t = tinggi trapesium, $s1$ = panjang sisi sejajar 1, dan $s2$ = panjang sisi sejajar 2 (asumsikan: $t > 0$, $s1 > 0$, $s2 > 0$, dan $s1$ tidak sama dengan $s2$) dan menghasilkan luas trapesium berdasarkan rumus: $\text{luas} = \frac{1}{2} * t * (s1 + s2)$

Contoh aplikasi fungsi dan hasilnya:

```
> luasTrapesium 2 4 3  
7.0
```


Luas Trapesium

Jawaban

module LuasTrapesium **where**

-- DEFINISI DAN SPESIFIKASI

luasTrapesium :: **Float** -> **Float** -> **Float** -> **Float**

{- Menghitung luas trapesium berdasarkan input tinggi (t) serta dua sisi yang sejajar (a dan b) -}

-- REALISASI

luasTrapesium t a b = t * (a + b)/2

-- APLIKASI

-- luasTrapesium 5 12 13

Luas Segitiga

Opsi Jawaban

module LuasSegitiga **where**

-- DEFINISI DAN SPESIFIKASI

luasSegitiga :: **Float** -> **Float** -> **Float**

{- Menghitung luas segitiga berdasarkan input panjang alas dan tinggi -}

-- REALISASI

luasSegitiga a t = **1/2** * a * t

luasSegitiga a t = **0.5** * a * t

} Dapat dipilih salah satu

-- APLIKASI

-- luasSegitiga 10 10

Nama File: LuasSegitiga.hs

Header: module LuasSegitiga where

Buatlah definisi, spesifikasi, dan realisasi fungsi

luasSegitiga yang menerima masukan 2 buah bilangan real (float) a dan t dengan a = alas segitiga dan t = tinggi segitiga (asumsikan: $a > 0$, $t > 0$) dan menghasilkan luas segitiga berdasarkan rumus: $\text{luas} = \frac{1}{2} * a * t$

Contoh aplikasi fungsi dan hasilnya:

```
> luasSegitiga 3 4  
6.0
```

Persamaan Kuadrat

Opsi Jawaban

module PersamaanKuadrat where

Nama File: PersamaanKuadrat.hs

Header: module PersamaanKuadrat where

Buatlah definisi, spesifikasi, dan realisasi fungsi **persamaanKuadrat** yang menerima 3 buah bilangan integer (a,b,c) dan sebuah nilai integer x, menghasilkan nilai persamaan kuadrat yang dibentuk oleh ketiga bilangan tersebut untuk nilai x yang diberikan dengan rumus: $ax^2 + bx + c$

Contoh aplikasi fungsi dan hasilnya:

```
> persamaanKuadrat 1 2 1 (-1)
0
```

```
-- DEFINISI DAN SPESIFIKASI
persamaanKuadrat :: Int -> Int -> Int -> Int ->
Int
{- Menghitung hasil persamaan kuadrat  $ax^2 + bx + c$  -}

-- REALISASI
persamaanKuadrat a b c x = a * x ^ 2 + b * x + c

-- APLIKASI
-- persamaanKuadrat 1 2 1 (-1)
```



Soal 2



Konversi Suhu

Nama File: KonversiSuhu.hs

Header: module KonversiSuhu where

Buatlah sebuah fungsi **konversiSuhu** (definisi, spesifikasi, dan realisasi) yang digunakan untuk mengkonversi suhu dari satu satuan Celcius ke satuan suhu yang lain, yaitu Fahrenheit, Reamur, atau Kelvin. Berikut adalah rumus untuk melakukan konversi jika suhu dalam derajat Celcius adalah C:

Suhu Tujuan	Rumus Konversi
Reamur	$4/5 * C$
Fahrenheit	$(9/5 * C) + 32$
Kelvin	$C + 273.15$

Fungsi ini menerima masukan:

- 1 buah nilai bertipe real (float), misalnya t, yang merupakan besaran suhu dalam derajat Celcius.
- 1 buah kode satuan suhu konversi, bertipe karakter, misalnya k, yang diasumsikan bernilai 'R' (Reamur), 'F' (Fahrenheit), atau 'K' (Kelvin).

Fungsi menghasilkan suhu dalam satuan k yang merupakan konversi suhu t derajat Celcius.

Contoh:

konversiSuhu (25, 'R') artinya adalah konversi suhu 25 derajat Celcius ke suhu dalam derajat Reamur.

Contoh aplikasi fungsi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	konversiSuhu 25 'R'	20	$25^{\circ}\text{C} = 4/5 * 25^{\circ}\text{R}$ (=) $25^{\circ}\text{C} = 20^{\circ}\text{R}$
2.	konversiSuhu 37 'F'	98.6	$37^{\circ}\text{C} = (9/5 * 37^{\circ}\text{F}) + 32^{\circ}\text{F}$ (=) $37^{\circ}\text{C} = 98.6^{\circ}\text{F}$
3.	konversiSuhu (-30) 'K'	243.15	$-30^{\circ}\text{C} = -30\text{K} + 273.15\text{K}$ (=) $0^{\circ}\text{C} = 243.15\text{K}$

Opsi Jawaban

module KonversiSuhu where

-- DEFINISI DAN SPESIFIKASI

konversiSuhu :: Float -> Char -> Float

-- REALISASI

konversiSuhu t k

| k == 'R' = $4/5 * t$

| k == 'F' = $(9/5 * t) + 32$

| k == 'K' = $t + 273.15$

-- APLIKASI

-- konversiSuhu 25 'R'

Max 3 Bilangan

Nama File: Max3.hs

Header: module Max3 where

Tuliskanlah fungsi **max3** yang menerima 3 buah bilangan integer yang berbeda, dan menuliskan nilai terbesar di antara ketiganya. Tidak boleh membuat fungsi antara. Berikut adalah definisi dan spesifikasi fungsi dalam notasi fungsional:

`max3 : 3 integer -> integer`

{ max3 (a,b,c) mengirimkan nilai yang paling besar di antara a, b, dan c.
Asumsi: a, b, c bilangan berbeda }

Contoh aplikasi dan hasil:

No	Aplikasi	Hasil	Keterangan
1.	max3 1 3 5	5	Bilangan terbesar adalah 5
2.	max3 (-10) 0 (-5)	0	Bilangan terbesar adalah 0
3.	max3 20 34 33	34	Bilangan terbesar adalah 34

Opsi Jawaban (dengan Guard)

module Max3 where

-- DEFINISI DAN SPESIFIKASI

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

{- max3 (a,b,c) mengirimkan nilai yang paling besar di antara a, b, dan c. Asumsi: a, b, c bilangan berbeda -}

-- REALISASI

max3 a b c

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

} Hati-hati dengan kondisinya!

-- APLIKASI

-- max3 1 2 3



Soal 3



Klasifikasi Komputer (1)

Setiap komputer dapat masuk ke salah satu dari 5 kelompok, yaitu 1, 2, 3, 4, dan 5.

Komputer diklasifikasikan berdasarkan CPU, GPU, dan harddisk. Ketentuannya:

1. CPU > 7, GPU > 7, dan harddisk > 7, maka kelompok 5
2. Jika setidaknya satu kemampuan bernilai kurang dari sama dengan 7, maka kelompok 4
3. Jika CPU, GPU, dan harddisk ≤ 7 , maka kelompok 3
4. Jika CPU atau GPU kurang dari 5, maka kelompok 2
5. Jika salah satu nilai < 2, maka kelompok 1 (tidak peduli dengan nilai kemampuan lainnya)

Prioritas kelompok terurut dari nomor kelompok terkecil hingga terbesar.

Misalnya, apabila masukan cocok dengan kelompok 2 dan 4, maka yang dianggap benar adalah kelompok 2.

Ketentuan setiap kelompok setelah diterjemahkan

1. **CPU < 2 atau GPU < 2 atau HDD < 2**
2. **CPU < 5 atau GPU < 5**
3. **CPU ≤ 7 dan GPU ≤ 7 dan harddisk ≤ 7**
4. **...**
5. **CPU > 7 dan GPU > 7 dan harddisk > 7**

Klasifikasi Komputer (2)

Kesalahan umum:

1. Kurang teliti dalam membaca soal dan mengolah informasi dari soal.
2. Urutan if-then-else atau | yang kurang tepat
3. Kurang tanda kurung
4. Tidak mengikuti contoh dan instruksi pada soal
 - a. Tidak menuliskan `module KlasifikasiKomputer where`
 - b. Salah nama file
 - c. Salah nama fungsi

Klasifikasi Komputer (3)

```
module KlasifikasiKomputer where
```

```
-- KLASIFIKASI KOMPUTER
```

```
-- klasifikasi(c, g, h)
```

```
-- DEFINISI DAN SPESIFIKASI
```

```
klasifikasi :: Int -> Int -> Int -> Int
```

```
{- klasifikasi(c, g, h) menerima integer c (kemampuan CPU), g (kemampuan GPU), dan h (kemampuan harddisk) untuk  
diklasifikasikan -}
```

```
-- REALISASI
```

```
klasifikasi c g h
```

```
| (c < 2 || g < 2 || h < 2) = 1
```

```
| (c < 5 || g < 5) = 2
```

```
| (c <= 7 && g <= 7 && h <= 7) = 3
```

```
| (c > 7 && g > 7 && h > 7) = 5
```

```
| otherwise = 4
```

Menghitung Indeks Mahasiswa

Untuk kemudahan penentuan indeks mahasiswa, tim dosen membuat sebuah program yang akan melakukan klasifikasi indeks mahasiswa berbasis aturan.

Hanya ada 5 kemungkinan indeks dari seorang mahasiswa yaitu A, B, C, D, dan E. Setiap indeks dinotasikan dengan bilangan, A = 4, B = 3, dst sampai E = 0. Berikut adalah aturan penilaian :

- Apabila setiap nilai (3 nilai) lebih dari sama dengan 75, maka mendapat indeks A
- Apabila hanya 1 atau 2 nilai yang lebih dari sama dengan 75, maka mendapat indeks B
- Apabila nilai tubes yang di bawah 40, tapi uts dan uas di atas 40 (biarpun di atas 75 sekalipun) maka mendapat indeks C
- Apabila semua nilai di bawah 75 dan tidak ada yang di bawah 40, maka juga mendapat indeks C
- Apabila nilai uts atau uas di bawah 40, maka mendapat indeks D tidak peduli nilai tubes
- Apabila ada satu nilai yang 0, maka indeks otomatis E

Buatlah definisi, spesifikasi, dan realisasi sebuah fungsi yang bernama **hitungIndeks** yang menerima 3 input bertipe float yaitu (berturut-turut) nilaiUTS, nilaiUAS, dan nilaiTubes. Fungsi akan mengeluarkan indeks mahasiswa (dinotasikan dengan bilangan bulat).

Cat : asumsikan saja bahwa input benar [0..100]

Contoh :

```
> hitungIndeks 100 100 0
```

```
0
```

```
> hitungIndeks 100 100 20
```

```
2
```

```
> hitungIndeks 100 20 100
```

```
1
```

Menghitung Indeks Mahasiswa

Opsi Jawaban

`module HitungIndeks where`

`-- DEFINISI DAN SPESIFIKASI`

`hitungIndeks :: Float -> Float -> Float -> Int`

`{- Menghitung indeks mahasiswa berdasarkan nilai UTS, nilai UAS, dan nilai Tubes
-}`

`-- REALISASI`

`hitungIndeks uts uas tubes`

`| (uts == 0) || (uas == 0) || (tubes == 0) = 0
| (uts < 40) || (uas < 40) = 1
| (uts >= 75) && (uas >= 75) && (tubes >= 75) = 4
| (tubes < 40) && (uas >= 40) && (uts >= 40) = 2
| (uts < 75) && (uas < 75) && (tubes < 75) = 2
| otherwise = 3`

`-- APLIKASI`

`-- hitungIndeks 100 100 0`

Menghitung Indeks Mahasiswa

Kesalahan Umum:

1. Kurang teliti dalam membaca serta menganalisis pernyataan dalam soal sehingga kasus percabangan salah
2. Urutan if-then-else atau '|' yang masih kurang benar

Seleksi Kerja

Sebuah perusahaan ingin melakukan penerimaan karyawan baru. Terdapat 4 lowongan kerja yang dibuka oleh perusahaan tersebut, yaitu pekerjaan A, B, C, dan D. Untuk melamar pada perusahaan tersebut, dibutuhkan pengalaman managerial dan pengalaman software engineer. Ketentuan untuk melamar pekerjaan adalah sebagai berikut.

1. Jika pelamar memiliki pengalaman managerial ≥ 2 tahun dan memiliki pengalaman menjadi software engineer ≥ 4 tahun, maka ybs bisa mendaftar untuk semua pekerjaan.
2. Jika pelamar memiliki kemampuan managerial < 2 tahun namun memiliki pengalaman menjadi software engineer ≥ 4 tahun, maka ybs bisa mendaftar untuk pekerjaan B. Selain itu, jika pengalaman menjadi software engineer < 4 tahun, maka ybs hanya dapat mendaftar untuk pekerjaan C.
3. Jika pengalaman menjadi software engineer < 4 tahun namun memiliki pengalaman managerial ≥ 2 tahun, maka ybs dapat mendaftar untuk pekerjaan D.
4. Semua pelamar dapat melamar untuk pekerjaan C.

Buatlah sebuah fungsi **seleksi** yang menerima input 2 buah integer, misalnya m, s, dan p, dengan m mewakili lama pengalaman managerial seseorang (dalam tahun, bernilai ≥ 0), s mewakili lama pengalaman menjadi software engineer seseorang (dalam tahun, bernilai ≥ 0), dan p mewakili jenis pekerjaan yang diinginkan (bernilai A..D).

Fungsi akan mengembalikan nilai True jika pelamar dengan pengalaman managerial m dan pengalaman menjadi software engineer s melamar untuk pekerjaan p. Jika kondisi tidak terpenuhi, maka akan dikembalikan nilai False.

Contoh:

```
> seleksi 0 1 'A'
```

```
False
```

```
> seleksi 0 0 'C'
```

```
True
```

```
> seleksi 2 5 'B'
```

```
True
```

Seleksi Kerja

Salah satu cara melihat perusahaan yang menerima:

m = managerial, s = software engineer

Kalimat ke-	Syarat A	Syarat B	Syarat C	Syarat D
1	$m \geq 2; s \geq 4$	$m \geq 2; s \geq 4$	$m \geq 2; s \geq 4$	$m \geq 2; s \geq 4$
2	$m \geq 2; s \geq 4$	$s \geq 4$	$m \geq 2$	$m \geq 2; s \geq 4$
3	$m \geq 2; s \geq 4$	$s \geq 4$	$m \geq 2$	$m \geq 2$
4	$m \geq 2; s \geq 4$	$s \geq 4$	((tidak ada syarat))	$m \geq 2$

Seleksi Kerja

Opsi Jawaban

```
module SeleksiKerja where
```

```
-- SELEKSI KERJA
```

```
-- DEFINISI DAN SPESIFIKASI
```

```
seleksi :: Int -> Int -> Char -> Bool
```

```
{- seleksi(m, s, p) -}
```

```
{- mengecek apakah seseorang dapat masuk ke perusahaan 'p' dengan pengalaman  
managerial 'm' dan software engineer 's' yang ia punya -}
```

```
-- REALISASI
```

```
seleksi m s p
```

```
  | (p == 'A') = (m >= 2) && (s >= 4)
```

```
  | (p == 'B') = (s >= 4)
```

```
  | (p == 'C') = True
```

```
  | (p == 'D') = (m >= 2)
```


Seleksi Kerja

Kesalahan Umum:

1. Kurang teliti dalam membaca serta menganalisis pernyataan dalam soal
2. Lupa menambahkan `Boolean(Bool)` sebagai output dari fungsi seleksi ini
3. Urutan if-then-else atau '|' yang masih kurang benar



Soal 4



Nama File: JamLembur.hs

Header: module JamLembur where

Yukirin adalah penyanyi terkenal yang merupakan *center* dari *idol group* ITB48. Minggu depan, Yukirin akan mengadakan konser di Gedung Sanasa Buyada Gashena di Inaugurasi Terang Bulan. Untuk mempersiapkan konser tersebut Yukirin memiliki kru yang sangat handal, tapi sangat strict dengan jam kerja. Kalau mereka pulang melebihi jam pulang kerja yang ditentukan yaitu pukul 16:20:00, mereka minta dibayar lembur. Karena kru ini sangat handal, Yukirin tidak punya pilihan lain selain menuruti kemauan mereka daripada konsernya berantakan.

Buatlah sebuah fungsi (definisi, spesifikasi, dan realisasi) **jamLembur**, yang menerima input tiga buah bilangan integer yang merepresentasikan jam (0..23), menit (0..59) dan detik (0..59). Ketiga input tersebut menunjukkan jam pulang dari kru konser. Jam pulang kru konser yang resmi adalah jam 16:30:00. Dari input jam pulang kru, fungsi akan menghasilkan tuple yang terdiri atas:

- Lembur atau tidak (True: lembur; False: tidak lembur)
- Selisih dari jam pulang karyawan dengan jam pulang resmi kru berupa nilai selisih dalam jam, menit, dan detik (3 integer). Petunjuk: ubah masukan jam, menit, dan detik ke jumlah detik; cari selisih dalam detik; kembalikan selisih ke jam, menit, detik.

Asumsi: masukan selalu berupa integer positif dan pasti valid merepresentasikan jam sesuai definisi di atas.

Contoh aplikasi fungsi dan hasilnya

```
> jamLembur 17 15 00
(True,0,45,0)
> jamLembur 1 5 20
(False,15,24,40)
> jamLembur 0 0 0
(False,16,30,0)
```

Jam Lembur

Opsi Jawaban

```
module JamLembur where
```

```
-- DEFINISI DAN SPESIFIKASI
```

```
jamLembur :: Int -> Int -> Int -> (Bool, Int, Int, Int)
```

```
-- REALISASI
```

```
jamLembur j m d =
```

```
  let
```

```
    selisihDetik j m d = abs( ((j*3600) + (m*60) + d) - ((16 * 3600) + (30 * 60)) )
```


```
  in
```

```
    if (j >= 16 && m >= 30 && d > 0) || (j > 16) then
```

```
      (True, div (selisihDetik j m d) 3600, div (mod (selisihDetik j m d) 3600) 60, mod (selisihDetik j m d) 60)
```

```
    else (False, div (selisihDetik j m d) 3600, div (mod (selisihDetik j m d) 3600) 60, mod (selisihDetik j m d) 60)
```

Kesalahan Umum:

1. Kurang teliti dalam membaca serta menganalisis pernyataan dalam soal
 2. Urutan if-then-else atau '}' yang masih kurang benar
- 

Jam Terlambat

Time limit	1 s
Memory limit	64 MB

Nama File: JamTerlambat.hs

Header: module JamTerlambat where

Yukirin adalah penyanyi terkenal yang merupakan *center* dari *idol group* ITB48. Minggu depan, Yukirin akan mengadakan konser di Gedung Sanasa Buyada Gashena di Inaugurasi Terang Bulan. Konser akan mulai tepat pada pukul 08.30. Di Inaugurasi Terang Bulan, Anda dikatakan terlambat apabila datang lewat dari waktu mulai acara. Sementara bila Anda datang tepat waktu atau lebih awal, Anda tidak dikatakan terlambat.

Penonton konser tentunya memiliki tingkat kekecewaan tertentu. Untuk satu detik setiap Yukirin terlambat, tingkat kekecewaan bertambah 10 poin. Namun apabila Yukirin tidak terlambat, penonton tidak akan kecewa.

Tuliskan fungsi **jamTerlambat** yang menerima input tiga buah bilangan integer yang merepresentasikan jam (0..23), menit (0..59), detik (0..59). Diberikan waktu mulai konser 08.30. Fungsi tersebut mengembalikan tuple yang terdiri atas :

- Selisih antara waktu input dengan waktu mulai konser berupa nilai selisih dalam jam, menit, dan detik (3 integer). Petunjuk: ubah masukan jam, menit, dan detik ke jumlah detik; cari selisih dalam detik; kembalikan selisih ke jam, menit, detik.
- Terlambat atau tidak (true: terlambat, false: tidak terlambat)
- Tingkat kekecewaan penonton, jumlah selisih detik keterlambatan dikali 10 jika terlambat. 0 jika tidak terlambat.

Asumsi: masukan selalu berupa integer positif dan pasti valid merepresentasikan jam sesuai definisi di atas.

Contoh aplikasi fungsi dan hasilnya:

```
> jamTerlambat 7 55 0
(0,35,0,False,0)
> jamTerlambat 8 45 20
(0,15,20,True,9200)
> jamTerlambat 10 0 01
(1,30,1,True,54010)
```

Jam Terlambat

Opsi Jawaban

```
module JamTerlambat where
  -- DEFINISI DAN SPESIFIKASI
  convert :: Int -> Int -> Int -> Int
  jamTerlambat :: Int -> Int -> Int -> (Int, Int, Int, Bool, Int)

  -- REALISASI
  convert h m d = (h*3600 + m*60 + d)
  jamTerlambat h m d =
    let jamMulaiToDetik = (convert 8 30 0)
        jamMasukToDetik = (convert h m d)
        selisihDetik = if jamMulaiToDetik < jamMasukToDetik then jamMasukToDetik - jamMulaiToDetik
                        else jamMulaiToDetik - jamMasukToDetik
    in (div selisihDetik 3600,
        div (mod selisihDetik 3600) 60,
        mod (mod selisihDetik 3600) 60,
        jamMulaiToDetik < jamMasukToDetik,
        if jamMulaiToDetik < jamMasukToDetik then selisihDetik*10 else 0)
```

Konversi Jam

Time limit	1 s
Memory limit	64 MB

Nama File : KonversiJam.hs

Nama Modul : KonversiJam

John dan kekasihnya menjalani hubungan jarak jauh, mereka terpisah oleh jarak dan waktu. John berada di Indonesia dan kekasihnya berada di Eropa yang memiliki waktu 7 jam lebih lambat dibanding Indonesia. Suatu hari John terbangun dari tidurnya dan ia tersadar bahwa ia lupa bahwa kemarin kekasihnya berulang tahun dan ia belum mengucapkannya. Beruntung dengan adanya selisih waktu, John mungkin masih sempat mengucapkan selamat ulang tahun.

Bantulah John dengan membuat sebuah fungsi bernama **konversiJam** yang dapat menerima input 3 integer yang merupakan waktu John terbangun dalam format jam(0..23) menit(0..59) detik(0..59) yang dapat mengeluarkan output berupa **tuple** yang berisi:

- Apakah John masih sempat mengucapkan selamat ulang tahun (True: sempat, False: tidak).
- Waktu setempat dimana kekasih John berada.

Asumsi: input selalu valid.

Contoh:

```
> konversiJam 06 00 00  
(True,23,0,0)
```


Konversi Jam

Opsi Jawaban

```
module KonversiJam where
```

```
-- Definisi dan Spesifikasi
```

```
konversiJam :: Int -> Int -> Int -> (Bool,Int,Int,Int)
```

```
-- Realisasi
```

```
konversiJam j m d =
```

```
  if j >= 7
```

```
  then (False, j-7, m, d)
```

```
  else (True, 24-(7-j), m, d)
```

Soal

Kinerja Karyawan

Time limit	1 s
Memory limit	64 MB

Nama File: Kinerja.hs

Header: module Kinerja where

Buatlah sebuah fungsi **kinerja** (definisi, spesifikasi, dan realisasi), yang menerima input tiga buah bilangan integer yang merepresentasikan jam (0..23), menit (0..59) dan detik (0..59). Ketiga input tersebut menunjukkan jam selesainya pekerjaan seorang karyawan kebersihan di sebuah perusahaan. Perusahaan ingin memberikan bonus pada karyawan yang menyelesaikan pekerjaannya lebih cepat dan memberikan penalti untuk karyawan pekerjaannya lebih lambat dari standar jam selesai. Standar jam selesai bekerja adalah jam 17:30:00.

Fungsi akan menghasilkan sebuah tuple (j, m, d, s) dengan penjelasan sebagai berikut:

- j, m, dan d mewakili durasi antara jam selesainya pekerjaan karyawan dengan standar jam selesai berupa nilai dalam jam, menit, dan detik (3 integer). Hasilnya selalu positif. Petunjuk: ubah masukan jam, menit, dan detik ke jumlah detik; cari selisih dalam detik; kembalikan selisih ke jam, menit, detik.
- s mewakili status kinerja karyawan, apakah lebih awal, tepat waktu, atau terlambat (integer, dengan nilai 0: tepat waktu, 1: lebih awal, -1: terlambat).

Asumsi: masukan selalu berupa integer positif dan pasti valid merepresentasikan jam sesuai definisi di atas.

Contoh aplikasi fungsi:

```
> kinerja 17 0 0
(0,30,0,1)
> kinerja 17 30 0
(0,0,0,0)
> kinerja 18 30 5
(1,0,5,-1)
```

Haskell ▾

Kinerja Karyawan

Opsi Jawaban

```
-- module Kinerja where
```

```
-- Penilaian Kinerja Karyawan
```

```
-- DEFINISI DAN SPESIFIKASI
```

```
kinerja :: Int -> Int -> Int -> (Int, Int, Int, Int)
```

```
-- Fungsi menerima jam selesai pekerjaan karyawan dan mengembalikan berapa lama selisih waktu selesai pekerjaan dengan standar jam kerja dan menentukan apakah karyawan selesai tepat waktu/terlambat/lebih awal.
```

```
-- REALISASI
```

```
kinerja j m d =
```

```
  let
```

```
    jumlah_detik = (17*3600 + 30*60) - (j*3600 + m*60 + d)
```

```
  in
```

```
    (div (abs(jumlah_detik)) 3600,  
     div (mod (abs(jumlah_detik)) 3600) 60,  
     mod (mod (abs(jumlah_detik)) 3600) 60,  
     if (jumlah_detik < 0) then -1  
     else if (jumlah_detik > 0) then 1  
     else 0)
```

```
-- APLIKASI
```

```
-- kinerja 18 30 5
```

Kinerja Karyawan

Kesalahan Umum:

Analisis kasus dalam soal yang kurang tepat sehingga banyak yang menggunakan analisis kondisional yang tidak tepat.

Time limit	1 s
Memory limit	64 MB

Nama File : JamBangun.hs

Nama Modul : JamBangun

Jane adalah seorang yang sangat teratur, ia selalu tidur **tepat** selama 8 Jam, pada hari ini ia tidur pada pukul **23.45.00**.

Namun, teman sekamar Jane tiba tiba melihat bahwa ada pesan masuk pada ponsel Jane dari dosen pembimbingnya. Jane diminta untuk menelepon dosen tersebut besok pada jam tertentu.

Ada kemungkinan bahwa Jane terlambat bangun dan melewati panggilan yang sangat penting ini sehingga teman Jane harus membangunkan Jane jika Jane akan terlambat bangun.

Bantulah teman Jane untuk membuat program yang dapat menerima input 3 Integer berupa jam(0..23) menit(0..59) detik(0.59) yang merupakan jam yang ditentukan oleh dosen pembimbing Jane

Lalu program dapat mengeluarkan output berupa **tuple** yang berisi:

Apakah Jane akan bangun melewati jam yang telah ditentukan (True: ya, False: Tidak)

Waktu selisih antara waktu Jane bangun dan waktu yang ditentukan pak Dosen.

Asumsi: input selalu valid.

Contoh:

```
> jamBangun 07 15 00
```

```
(True,0,30,0)
```

JamBangun

Opsi Jawaban

module JamBangun **where**

-- JamBangun - jamBangun(j,m,d)

-- Definisi dan Spesifikasi

jamBangun :: **Int** -> **Int** -> **Int** -> (**Bool**,**Int**,**Int**,**Int**)

-- jamBangun(a,b,c,x) menghasilkan output tuple berupa bool (True untuk bangun melewati jam, False sebaliknya)

-- beserta selisih waktu antara waktu bangun dan waktu yang telah ditentukan

-- Realisasi

jamBangun j m d =

let total_detik_dosen = j * 60 * 60 + m * 60 + d

total_detik_bangun = 7 * 60 * 60 + 45 * 60 + 0

selisih = abs(total_detik_dosen - total_detik_bangun)

in

if total_detik_dosen < total_detik_bangun **then**

(**True**, div selisih 3600, mod(div selisih 60) 60, mod selisih 60)

else

(**False**, div selisih 3600, mod(div selisih 60) 60, mod selisih 60)

-- Contoh aplikasi

-- jamBangun 07 15 00

-- (True,0,30,0)

Kesalahan Umum:

1. Salah dalam menghitung selisih jam bangun jane dan waktu dosennya menelpon.
2. Jika melakukan perhitungan selisih dengan jumlah detik, peserta salah dalam mengkonversi kembali selisih tersebut ke bentuk jam, menit, detik.

Lama Tidur

Soal

Hari ini para panitia Arkavidia bekerja keras siang malam mempersiapkan acara besok hari. Namun, panitia jugalah seorang manusia yang pastinya butuh tidur yang cukup. Besok acara dimulai pada pukul 05.00 pagi, dan saat mereka selesai mempersiapkan acara besok. Para panitia merasa lelah dan langsung tidur di tempat. Para panitia ingin mengetahui apakah mereka dapat tidur dengan cukup (≥ 6 jam) dan menghitung lama mereka bisa tidur.

Bantulah panitia dengan membuatlah sebuah fungsi dengan nama `lamaTidur` yang dapat menerima input 3 integer yang merupakan jam (0..23), menit(0..59), detik(0..59) waktu mereka selesai mempersiapkan acara dan dapat mengeluarkan output tuple berisi lama waktu bisa tidur dalam format jam, menit, detik (selisih waktu input dari pukul 05.00 pagi).

Asumsi: input selalu valid.

Contoh:

```
> lamaTidur 1 0 0  
(False, 4,0,0)
```


Lama Tidur

Opsi Jawaban

module LamaTidur **where**

-- SPESIFIKASI

hmstoSeconds :: Int -> Int -> Int -> Int

lamaTidur :: Int -> Int -> Int -> (Bool, Int, Int, Int)

-- REALISASI

hmstoSeconds j m d = j * 3600 + m * 60 + d

lamaTidur j m d =

let

waktuTidur =

if j < 5 **then** (hmstoSeconds 5 0 0) - (hmstoSeconds j m d)

else (hmstoSeconds 24 0 0) - (hmstoSeconds j m d) + (hmstoSeconds 5 0 0)

jam = div waktuTidur 3600

menit = div (waktuTidur - jam * 3600) 60

detik = waktuTidur - hmstoSeconds jam menit 0

in (waktuTidur >= 6 * 3600, jam, menit, detik)

Kesalahan Umum:

1. Kurang teliti dalam membaca serta menganalisis soal
2. Penggunaan *syntax* yang salah menyebabkan program gagal dikompilasi



Soal 5



Sum Of Digits

Nama File: SumOfDigits.hs

Header: module SumOfDigits where

Buatlah definisi, spesifikasi, dan realisasi dari fungsi **sumOfDigits**, yang merupakan fungsi menghitung hasil penjumlahan dari setiap bilangan tunggal yang terdapat di dalam sebuah bilangan integer positif atau 0.

Misalnya:

$\text{sumOfDigits}(234) = 2 + 3 + 4 = 9$

$\text{sumOfDigits}(38) = 3 + 8 = 11$

$\text{sumOfDigits}(5) = 5$

Fungsi ini harus diselesaikan menggunakan pendekatan rekursif.

```
sumOfDigits :: Int -> Int
-- sumOfDigits n menghasilkan penjumlahan setiap bilangan tunggal yang membentuk n
-- prekondisi n >= 0
```

Contoh aplikasi fungsi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	sumOfDigits 123	6	$1+2+3 = 6$
2.	sumOfDigits 234	9	$2+3+4 = 9$
3.	sumOfDigits 1234	10	$1+2+3+4 = 10$

Sum Of Digits

Opsi Jawaban

```
module SumOfDigits where
```

```
-- DEFINISI DAN SPESIFIKASI
```

```
sumOfDigits :: Int -> Int
```

```
{-
```

```
sumOfDigits(n)
```

```
Menerima masukkan n
```

```
Mengeluarkan jumlah setiap digit pada n
```

```
-}
```

```
-- REALISASI
```

```
sumOfDigits n
```

```
  | n == 0 = 0
```

```
-- Basis
```

```
  | otherwise = mod n 10 + sumOfDigits(div n 10)
```

```
-- Rekurens
```



Soal 6



Cacah Kelipatan

Nama File: NbKelipatanX.hs

Header: module NbKelipatanX where

Buatlah definisi, spesifikasi, dan realisasi fungsi **nbKelipatanX** yang menerima masukan dua buah integer positif (integer > 0), misalnya **m** dan **n**, serta sebuah integer positif lain, yaitu **x**, dan menghasilkan banyaknya bilangan kelipatan **x** di antara **m** dan **n** (**m** dan **n** termasuk) dengan menggunakan ekspresi rekursif.

Bilangan **y** disebut kelipatan bilangan **x**, jika **y** habis dibagi dengan **x**.

Prekondisi/syarat/asumsi yang berlaku adalah $m \leq n$ dan $x \leq n$.

Contoh aplikasi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	nbKelipatanX 1 1 1	1	Kelipatan 1 di antara [1..1] adalah 1
2.	nbKelipatanX 1 10 2	5	Kelipatan 2 di antara [1..10] adalah 2, 4, 6, 8, 10
3.	nbKelipatanX 5 14 3	3	Kelipatan 3 di antara [5..14] adalah 6, 9, 12

Cacah Kelipatan

Opsi Jawaban

module NbKelipatanX **where**

-- DEFINISI DAN SPESIFIKASI

nbKelipatanX :: Int -> Int -> Int -> Int

{Prekondisi: m <= n dan x <= n}

{Menerima masukan m, n, x}

{Mengeluarkan jumlah kelipatan x dalam rentang m sampai dengan n (inklusif)}

-- REALISASI

nbKelipatanX m n x

| m == n && mod m x == 0 = 1

| m == n && mod m x /= 0 = 0

| m /= n && mod m x == 0 =

1 + nbKelipatanX (m+1) n x

| otherwise =

nbKelipatanX (m+1) n x

Penjumlahan Kelipatan

Soal

Nama File: SumKelipatanX.hs

Header: module SumKelipatanX where

Buatlah definisi, spesifikasi, dan realisasi fungsi **sumKelipatanX** yang menerima masukan dua buah integer positif (integer > 0), misalnya **m** dan **n**, serta sebuah integer positif lain, yaitu **x**, dan menghasilkan jumlah total bilangan kelipatan **x** di antara **m** dan **n** (**m** dan **n** termasuk) dengan menggunakan ekspresi rekursif.

Bilangan **y** disebut kelipatan bilangan **x**, jika **y** habis dibagi dengan **x**.

Prekondisi/syarat/asumsi yang berlaku adalah $m \leq n$ dan $x \leq n$.

Contoh aplikasi fungsi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	sumKelipatanX 1 1 1	1	Kelipatan 1 di antara [1..1] adalah 1
2.	sumKelipatanX 1 10 2	30	Kelipatan 2 di antara [1..10] adalah 2, 4, 6, 8, 10 $2+4+6+8+10 = 30$
3.	sumKelipatanX 5 14 3	27	Kelipatan 3 di antara [5..14] adalah 6, 9, 12 $6+9+12 = 27$

Penjumlahan Kelipatan

Kesalahan Umum

1. Menentukan basis dan rekurens yang salah
2. Tidak paham soal atau tidak paham konsep rekursi

Penjumlahan Kelipatan

Opsi Jawaban

module SumKelipatanX where

-- Sum Kelipatan X -SumKelipatanX(a,b,x)

-- DEFINISI DAN SPESIFIKASI

sumKelipatanX :: Int -> Int -> Int -> Int

{- sumKelipatanX(a,b,x) menjumlahkan semua bilangan kelipatan x di range [a, b] -}

-- REALISASI

sumKelipatanX a b x

 | (mod a x /= 0) && (a <= b) = sumKelipatanX (a+1) b x

 | (a > b) = 0

 | (a <= b) && (mod a x == 0) = a + sumKelipatanX (a+1) b x

-- APLIKASI

-- sumKelipatanX 1 10 2



Soal Bonus



Hitung Bensin

Nama Modul : HitungBensin

Setelah setahun tidak pulang kampung, akhirnya Tuan Vin pun memberanikan diri untuk meminta cuti kepada bosnya. Bosnya sebenarnya ingin langsung menyetujui cuti Tuan Vin. Akan tetapi, dia ingin Tuan Vin membereskan pekerjaannya terlebih dahulu. Pekerjaan Tuan Vin sebenarnya cukup mudah. Dia hanya perlu menyiapkan bensin untuk seluruh kendaraan perusahaannya.

Kendaraan perusahaan Tuan Vin memiliki rute yang sangat unik. Awalnya, kendaraan tersebut akan terletak pada posisi X. Kemudian jika X adalah bilangan genap, kendaraan tersebut akan bergerak ke titik $X/2$. Jika X adalah bilangan ganjil, kendaraan tersebut akan bergerak ke posisi $(3X + 1)$. Hal ini terus dilakukan sampai kendaraan tersebut sampai ke kantor pusat yang terletak pada posisi 1. Untuk setiap perpindahan posisi, kendaraan tersebut akan menghabiskan bensin sebanyak 1 unit. Jika pada awalnya suatu kendaraan terletak pada posisi 11, kendaraan tersebut akan berpindah ke $(11 * 3 + 1) = 34$. Kemudian, kendaraan tersebut kemudian berpindah ke posisi 17,52,26,13,40,20,10,5,16,8,4,2 dan berakhir pada posisi 1 sehingga kendaraan tersebut menghabiskan bensin sebanyak 14 unit.

Tiap harinya, akan ada kendaraan yang berangkat dari posisi A sampai dengan posisi B. Tuan Vin kemudian menjumlahkan banyaknya bensin yang dibutuhkan untuk tiap-tiap kendaraan dari posisi A sampai dengan posisi B. Tuan Vin takut dia tidak sempat menyelesaikan kalkulasinya sebelum hari cutinya tiba. Sebagai teman baik Tuan Vin, Anda pun ingin membantu Tuan Vin dengan membuat sebuah fungsi untuk menghitung bensin yang perlu disiapkan. Fungsi **hitungBensin** menerima 2 buah bilangan bulat, A dan B ($A \leq B$). Fungsi ini kemudian mengeluarkan sebuah bilangan bulat yang menunjukkan konsumsi bensin dari tiap-tiap kendaraan dari A sampai B. Tentu saja sebagai programmer yang baik, Anda harus membuat definisi, spesifikasi, dan realisasi fungsi ini.

Contoh :

```
> hitungBensin 11 11
```

```
14
```

```
> hitungBensin 1 10
```

```
67
```

Keterangan:

Hitung Bensin

Opsi Jawaban

module HitungBensin **where**

hitungBensin :: Int -> Int -> Int

hitungBiaya :: Int -> Int

hitungBiaya x

| x == 1 = 0

| mod x 2 == 0 = 1 + hitungBiaya (div x 2)

| otherwise = 1 + hitungBiaya (x*3 + 1)

hitungBensin a b

| a == b = hitungBiaya a

| otherwise = hitungBiaya b + hitungBensin a (b-1)

Hitung Bensin

Kesalahan Umum

1. Tidak dikerjain hehe
2. Tidak paham soal atau tidak paham konsep rekursi

→ *Bonus ga wajib, gapapa, have fun :D*