

## Lembar Kerja Praktikum KOM120C Pemrograman

### 10: Functional Programming III

#### PETUNJUK PRAKTIKUM

##### Review Functional Programming

- Pada FP, fungsi diperlakukan sebagai First Class Citizens, mirip bagaimana object diperlakukan pada OOP
- Suatu program dapat terdiri dari gabungan berbagai fungsi, di mana suatu fungsi bisa menjadi input atau bahkan output dari fungsi lain
- Biasanya menggunakan fungsi murni, yakni fungsi yang tidak dipengaruhi faktor luar dan tidak memiliki side-effect
- Compiler Scala Online: <https://onecompiler.com/scala>

##### sorted/sortBy/sortWith

sorted: pengurutan dengan urutan default (misal: angka dari kecil ke besar)

```
val numbers = List(5, 3, 1, 4, 2)
println(numbers.sorted) // List(1, 2, 3, 4, 5)
```

sortBy: pengurutan dengan menggunakan fungsi pembandingan untuk mengurutkan (misal: urutkan string berdasarkan panjangnya)

```
val numbers = List("ayam", "bebek", "kambing", "sapi")
println(numbers.sortBy(_.length)) // List(ayam, sapi, bebek, kambing)
```

sortWith: pengurutan dengan menggunakan kriteria perbandingan selain default (misal: dari besar ke kecil)

```
val numbers = List(5, 3, 1, 4, 2)
println(numbers.sortWith(_ > _)) // List(5, 4, 3, 2, 1)
```

HOF juga sangat berguna untuk mengolah data bertipe string, berikut beberapa fungsi-fungsi HOF untuk tipe data strings:

- Length: panjang string
- Split: memecah string berdasarkan karakter tertentu (tokenisasi)
- Reverse: membalik string
- Contains: apakah string mengandung substring tertentu atau tidak

- startsWith/endsWith: memeriksa apakah string dimulai/diakhir dengan substring tertentu
- toLowerCase/toUpperCase: mengubah case (kecil/besar)
- indexOf: mencari indeks kemunculan pertama karakter tertentu di dalam string
- Dan sebagainya...

## TUGAS

1. Buat program untuk membaca input berupa N buah bilangan bulat, kemudian kelompokkan menjadi bilangan prima dan non-prima. (Gunakan groupBy dan buat fungsi terpisah untuk mengecek bilangan prima atau bukan)

Contoh Input
5 2 6 7 13 25
Contoh Output
Prima: 2, 7, 13 Non-prima: 6, 25
<pre>object NumberClassifier {   def main(args: Array[String]): Unit = {     val n = scala.io.StdIn.readInt()     val numbers = scala.io.StdIn.readLine().split(" ").map(_toInt)     val (primes, nonPrimes) = numbers.groupBy (isPrime). partition (_._1)     println("Prima: " + primes.values.flatten.mkString(", "))     println("Non-prima: " + nonPrimes.values.flatten.mkString(", "))   }    def isPrime (num: Int): Boolean = {     if (num &lt;=1) {       false     } else if (num == 2){       true     }     else {       val limit = math.sqrt(num).toInt + 1       val isDivisible = (2 to limit).exists(i =&gt; num % i == 0)       !isDivisible     }   } }</pre>

```
}  
}
```

2. Buat program untuk mengurutkan N daftar nama mahasiswa. Namun, pengurutan memperhatikan panjang string terlebih dahulu kemudian barulah memperhatikan abjad. Pengurutan dilakukan secara *ascending* dan keluaran dalam bentuk *uppercase*.

Contoh Inputya

```
10  
Adam  
Bella  
Candra  
Dian  
Erika  
Fadhila  
Guntur  
Hani  
Indra  
Jelita
```

Contoh Output

ADAM  
DIAN  
HANI  
BELLA  
ERIKA  
INDRA  
CANDRA  
GUNTUR  
JELITA  
FADHILA

```
object StudentName {  
  def main(args: Array[String]): Unit =  
  {  
    val n = scala.io.StdIn.readInt()  
    val names = Array.ofDim[String] (n)  
  
    for (i <- 0 until n) {  
      names (i) = scala.io.StdIn.readLine().toUpperCase  
    }  
    val urutanNama = names.sortWith((a,b)=>{  
      if (a.length != b.length) a.length < b.length  
      else a < b  
    })  
    urutanNama.foreach(println)  
  }  
}
```

3. Buat program untuk membaca input berupa N string kemudian mengembalikan string-string yang merupakan palindrome (*case insensitive*). List baru tersebut harus diurutkan secara berkebalikan dari urutan leksikografis, dan semua string dalam list harus dalam huruf kecil.

Contoh Input

```
7
LeVel
Y
makan
raCecar
IPB
Deed
Abcde
```

Contoh Output

```
y
racecar
level
deed
```

```
import scala.io.StdIn
import scala.math.Ordering
object PalindromeReverse {
  def main(args: Array[String]): Unit = {
    val n = StdIn.readInt()
    val strings = List.fill(n) (StdIn.readLine().toLowerCase)
  }

  val palindromes = strings.filter(isPalindrome).sorted(Ordering[String].reverse)
  palindromes.foreach(println)
  def isPalindrome (str: String): Boolean = {
    str == str.reverse
  }
}
```

4. Buat Program untuk mencari jumlah kata-kata spesifik yang muncul dari sebuah teks. Diberikan integer N dan N buah kata, cari jumlah kata kata itu dalam teks

Contoh Input

```
3
saya makan minum
saya suka makan buah dan minum teh tetapi saya tidak suka minum kopi
```

## Contoh Output

5

```
object WordCount {  
  def countWords (word: String, sentence: List[String]): Int = {  
    sentence.count(_.toLowerCase == word.toLowerCase)  
  }  
  def main(args: Array[String]): Unit = {  
    val numOfWorksStr = scala.io.StdIn.readLine().trim()  
    val numOfWorks = Integer.parseInt(numOfWorksStr)  
    val targetWords = scala.io.StdIn.readLine().split(" ").toList  
    val inputSentence = scala.io.StdIn.readLine().split(" ").toList  
    var result = targetWords.map(word => countWords (word, inputSentence)).sum  
    printf("%d", result)  
  }  
}
```

5. Diberikan sebuah string dengan format json. Extract key dan value dari string json tersebut secara manual

## Contoh Input

```
{'kota': 'bogor', 'universitas': 'IPB', 'jurusan': 'Ilmu Komputer'}
```

## Contoh Output

```
key : kota, universitas, jurusan  
value: bogor, IPB, Ilmu Komputer
```

```
object Json {  
  def main(args: Array[String]): Unit = {  
    val jsonString = "{ 'kota': 'bogor', 'universitas': 'IPB', 'jurusan': 'Ilmu  
Komputer' }"  
    val keyValueStrings = jsonString.split(",")  
    .map(_._trim().replaceAll("[{ }]", ""))  
    .toList  
    .filter(!_._isEmpty)  
    val keys=keyValueStrings.map(_._split(":")(0)._trim())  
    val values =keyValueStrings.map(_._split(":")(1)._trim())  
    println("keys: " + keys.mkString(", "))  
    println("values: " + values.mkString(", "))  
  }  
}
```

```
}
}
```

6. Buat program untuk mengelompokkan plat nomor kendaraan bermotor beberapa daerah, yaitu yang berawalan A, AB, B, dan F. Perlu diperhatikan bahwa plat nomor kendaraan memiliki format sebagai berikut: beberapa huruf yang menjadi kode daerah, diikuti beberapa angka, dan terakhir beberapa huruf. Kelompokkan menjadi sesuai dengan empat daerah tersebut jika dipastikan daftar plat nomor pasti hanya berasal dari antara keempat daerah tersebut. Output harus urutan [A, AB, B, F] dan jika tidak memiliki anggota tidak perlu dicetak.

#### Contoh Input

```
5
A1234BCD
B2345AAA
AB9999BA
A6009KID
AB1234BCD
```

#### Contoh Output

```
A: A1234BCD, A6009KID
AB: AB9999BA, AB1234BCD
B: B2345AAA
```

#### Penjelasan

F tidak perlu di print karena tidak memiliki anggota

```
object PlatNumber {
  def main(args: Array[String]): Unit = {
    val n=scala.io.StdIn.readInt()
    val platNumbers = List.fill(n) (scala.io.StdIn.readLine())
    val groupedNumbers = platNumbers.groupBy (getRegionCode)
    val sortedGroups = List("A", "AB", "B", "F")
    sortedGroups.foreach { group => groupedNumbers.get(group) match {
      case Some(numbers) =>
        println(s"$group: ${numbers.mkString(", ")}")
      case None =>
    }
  }
}

def getRegionCode (plateNumber: String): String = {
```

```
plateNumber.takeWhile(_isLetter).toUpperCase  
}  
}
```



