

Lembar Kerja Praktikum KOM120C Pemrograman 13

TUGAS

Note: Selesaikan semua soal dengan menggunakan OOP di scala

1. Udin merupakan seorang admin koperasi simpan pinjam di suatu kota di Udinesia. Ia tentunya mencatat nama dan saldo setiap nasabah di koperasinya. Perhatikan bahwa saldo seorang nasabah bisa minus. Tugas anda membantu Udin untuk membuat pencatatan yang lebih mudah menggunakan Scala.

Diberikan bilangan bulat Q. Q baris berikutnya berisi salah satu operasi dibawah:

- Simpan Nama Nominal
- Pinjam Nama Nominal
- CekSaldo Nama

Untuk setiap operasi CekSaldo, keluarkan saldo yang sesuai.

*Tidak ada bunga yang diterapkan di koperasi tempat Udin bekerja.

Contoh Input

```
7
Simpan Awaludin 50000
Pinjam Akhirudin 200000
CekSaldo Awaludin
Simpan Akhirudin 25000
Pinjam Awaludin 750000
CekSaldo Awaludin
CekSaldo Akhirudin
```

Contoh Output

```
50000
-700000
175000
```

```
import scala.io.StdIn.readLine
import scala.collection.mutable

class operasi(val name: String, private var balance: Int = 0) {
  def simpan(amount: Int): Unit = balance += amount
```

```

def pinjam(amount: Int): Unit = balance -= amount
def cekSaldo(): Int          = balance
}

class user {
  private val accounts = mutable.Map.empty[String, operasi]

  private def getOrCreate(name: String): operasi =
    accounts.getOrElseUpdate(name, new operasi(name))

  def simpan(name: String, amount: Int): Unit =
    getOrCreate(name).simpan(amount)

  def pinjam(name: String, amount: Int): Unit =
    getOrCreate(name).pinjam(amount)

  def cekSaldo(name: String): Int =
    getOrCreate(name).cekSaldo()
}

object Main extends App {
  val manager = new user
  val q       = readLine().toInt

  for (_ <- 1 to q) {
    val parts = readLine().split(" ", 3)
    parts(0) match {
      case "Simpan"    => manager.simpan(parts(1), parts(2).toInt)
      case "Pinjam"    => manager.pinjam(parts(1), parts(2).toInt)
      case "CekSaldo" => println(manager.cekSaldo(parts(1)))
      case other       => Console.err.println(s"Typo: $other")
    }
  }
}

```

Output:

```

7
Simpan Awaludin 50000
Pinjam Akhirudin 200000
CekSaldo Awaludin
50000
Simpan Akhirudin 25000
Pinjam Awaludin 750000
CekSaldo Awaludin
-700000
CekSaldo Akhirudin
-175000

```

2. Naofumi iwapetani sering melupakan nama teman - temannya suatu hari naofumi salah mengira orang lain sebagai temannya dan menanyakan jadwal untuk kelas dasar penggunaan perisai agar tidak mengulangi kesalahannya ia memutuskan untuk menyimpan semua nama teman dan deskripsi orang tersebut (berupa list sifat / penampilan), setelah selesai ia dapat melakukan pencarian terhadap nama orang atau deskripsi orang yang ia baru temui untuk menentukan apakah orang yang ia temui adalah teman atau bukan, Note untuk deskripsi urutan nama tidak harusurut.

Contoh Input

```
Ahmad Santoso: cerdas, cermat, tampan
Bernomali Anomali: anomali, suka pisang
Tungitina Nununina : berbentuk lonjong, cantik
Alomani Nuturina: anomali, cantik

Deskripsi: cantik
Nama: Ahmad Santoso
Deskripsi: anomali
Nama: Lunael beristi
```

Contoh Output

```
Cantik :
- Alomani Nuturina
- Tungitina Nununina

Ahmad Santoso adalah teman

Anomali :
- Bernomali Anomali
- Alomani Nuturina

Lunael Beristi bukan teman
```

```
import scala.io.StdIn
```

```

import scala.collection.mutable

case class Friend(name: String, descriptions: Set[String])

class FriendDir {
  private val friendsByName = mutable.Map.empty[String, Friend]
  private val namesByDesc = mutable.Map.empty[String, mutable.Set[String]]

  def addFriend(nameRaw: String, descRaw: Seq[String]): Unit = {
    val name = nameRaw.trim.split("\\s+").map(_.capitalize).mkString(" ")
    val descs = descRaw.map(_.trim.toLowerCase).toSet
    val friend = Friend(name, descs)
    friendsByName(name) = friend

    for (d <- descs) {
      val setNames = namesByDesc.getOrElseUpdate(d, mutable.Set.empty)
      setNames += name
    }
  }

  def description(descRaw: String): Seq[String] = {
    val key = descRaw.trim.toLowerCase
    namesByDesc.get(key).map(_.toSeq.sorted).getOrElse(Seq.empty)
  }

  def isFriend(nameRaw: String): Boolean = {
    val name = nameRaw.trim.split("\\s+").map(_.capitalize).mkString(" ")
    friendsByName.contains(name)
  }
}

object FriendList {
  def main(args: Array[String]): Unit = {
    val dir = new FriendDir

    val lines = Iterator.continually(StdIn.readLine()).takeWhile(_ != null).map(_.trim).filter(_.nonEmpty)

    for (line <- lines) {
      if (line.toLowerCase.startsWith("deskripsi:")) {
        val desc = line.split(":", 2)(1)
        val matches = dir.description(desc)
        println(s"${desc.trim.capitalize} :")
        matches.foreach(name => println(s"- $name"))

      } else if (line.toLowerCase.startsWith("nama:")) {
        val nameQ = line.split(":", 2)(1)
        if (dir.isFriend(nameQ)) {
          println(s"${nameQ.trim.split("\\s+").map(_.capitalize).mkString(" ")} adalah teman")
        } else {

```

```

        println(s"${nameQ.trim.split("\\s+").map(_.capitalize).mkString("
    ")} bukan teman")
    }

    } else {
        val Array(namePart, descPart) = line.split(":", 2)
        val descList = descPart.split(",").toSeq
        dir.addFriend(namePart, descList)
    }
}
}
}
}
}

```

Output:

```

Compiling project (Scala 3.6.4, JVM (21))
Compiled project (Scala 3.6.4, JVM (21))
Ahmad Santoso: cerdas, cermat, tampan
Bernomali Anomali: anomali, suka pisang
Tungitina Nununina : berbentuk lonjong, cantik
Alomani Nuturina: anomali, cantik
Deskripsi: cantik
Nama: Ahmad Santoso
Deskripsi: anomali
Cantik :
- Alomani Nuturina
- Tungitina Nununina
Ahmad Santoso adalah teman
Anomali :
- Alomani Nuturina
- Bernomali Anomali
Nama: Lunael beristi
Lunael Beristi bukan teman

```