Cavita Santi Krisnamurthy
20.01.4548

```
class Queue:
    def _init_(self):
        self.items = []
    def isEmpty(self):
        return self.items == []
    def enqueue(self, item):
        self.items.insert(0, item)
    def dequeue(self):
        return self.items.pop()
    def size(self):
        return len(self.items)

class Printer:
    def _init_(self, ppm):
        self.pagerate = ppm
        self.currentTask = None
        self.timeRemaining = 0
    def tick(self):
        if self.currentTask != None:
            self.timeRemaining = self.timeRemaining - 1
        if self.timeRemaining <= 0:
                self.currentTask = None
    def busy(self):
        if self.currentTask != None:
            return True
        else:
            return False
    def startNext(self, newtask):
        self.currentTask = newtask
        self.timeRemaining = newtask.getPages() * 60/self.pagerate

class Task:
    def _init_(self, time):
        self.timestamp = time
        self.pages = random.randrange(1, 11)
    def getStamp(self):
        return self.timestamp
    def getPages(self):
        return self.pages
    def waitTime(self, currenttime):
        return currenttime - self.timestamp
    def simulation(numSeconds, pagesPerMinute, total_students):
    for i in range(total_students):
        labprinter = Printer(pagesPerMinute)
        printQueue = Queue()
        waitingtimes = []
     for currentSecond in range(numSeconds):
```

```
            if newPrintTask():
                task = Task(currentSecond)
                printQueue.enqueue(task)

            if (not labprinter.busy()) and (not printQueue.isEmpty()):
                nexttask = printQueue.dequeue()
                waitingtimes.append(nexttask.waitTime(currentSecond))
                labprinter.startNext(nexttask)

            labprinter.tick()

        averageWait = sum(waitingtimes)/len(waitingtimes)
        print("Average Wait %6.2f secs %3d tasks remaining." %
            (averageWait, printQueue.size()))


def newPrintTask():
    num = random.randrange(1, 181)
    if num == 180:
        return True
    else:
        return False


total_students = int(input("Berapa banyak jumlah siswa? "))

simulation(3600, 5, total_students)
```