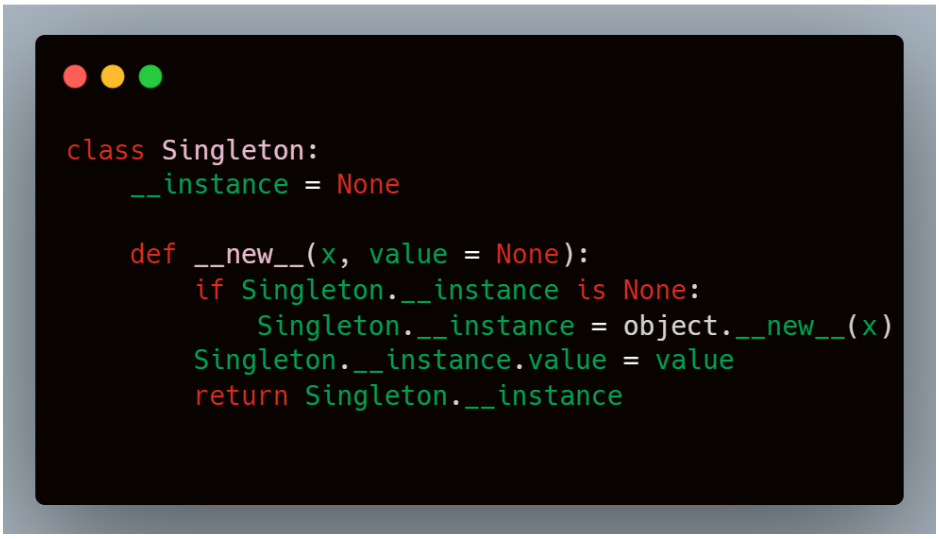
Nama : Rani Meliyana Putri

NIM : 11191062

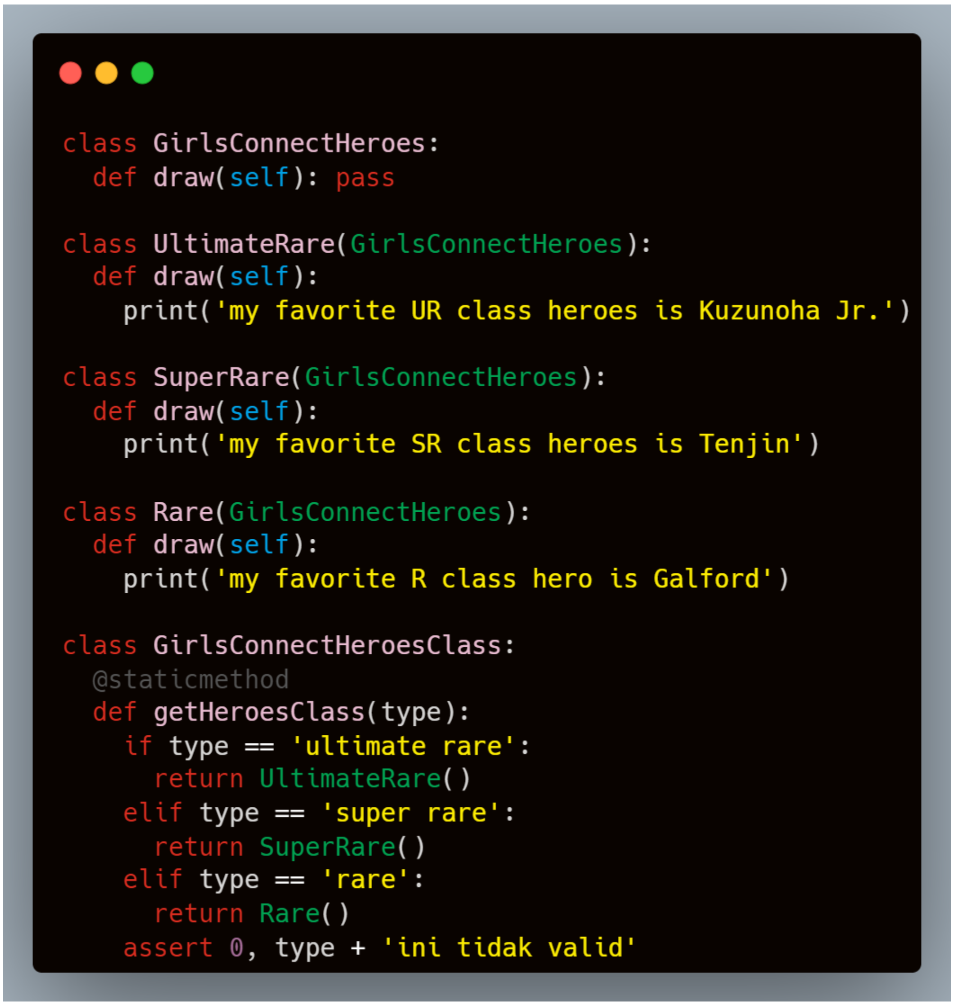
TUGAS 3 – PEMROGRAMAN FUNGSIONAL B

1. Singleton



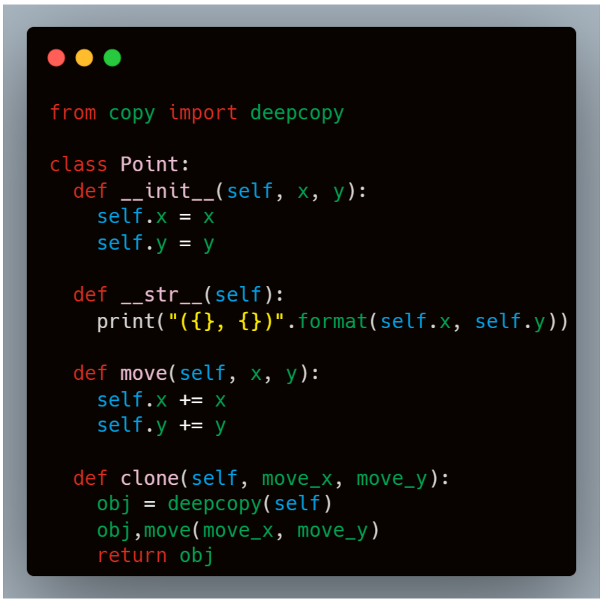
```python
class Singleton:
    __instance = None

    def __new__(x, value = None):
        if Singleton.__instance is None:
            Singleton.__instance = object.__new__(x)
        Singleton.__instance.value = value
        return Singleton.__instance
```

2. Abstract Factory



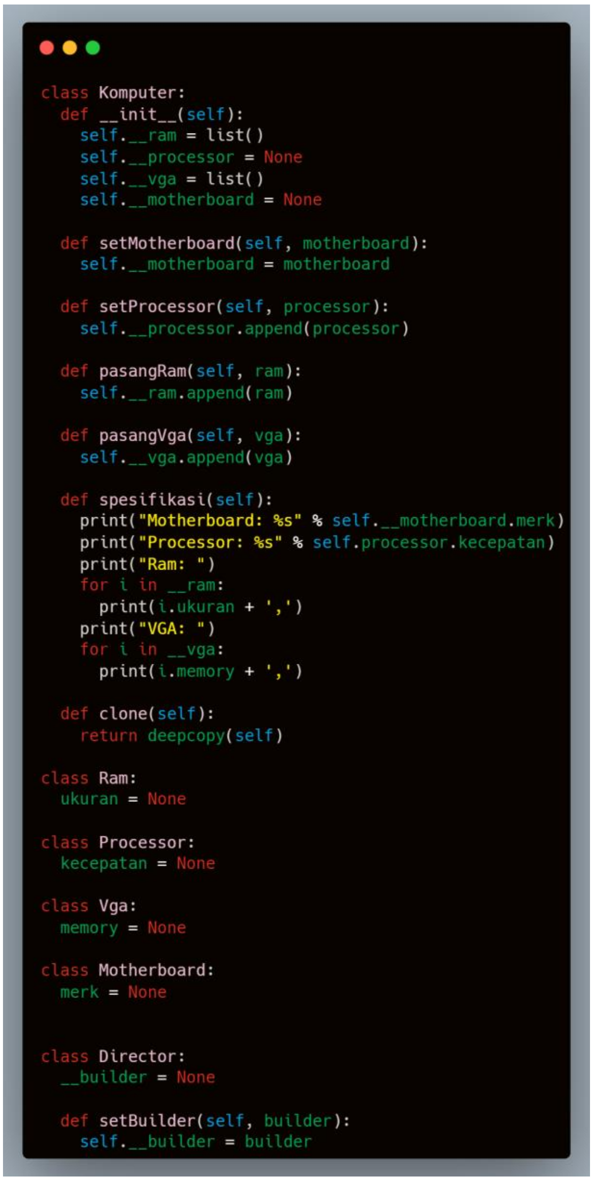
```python
class GirlsConnectHeroes:
  def draw(self): pass

class UltimateRare(GirlsConnectHeroes):
  def draw(self):
    print('my favorite UR class heroes is Kuzunoha Jr.')

class SuperRare(GirlsConnectHeroes):
  def draw(self):
    print('my favorite SR class heroes is Tenjin')

class Rare(GirlsConnectHeroes):
  def draw(self):
    print('my favorite R class hero is Galford')

class GirlsConnectHeroesClass:
  @staticmethod
  def getHeroesClass(type):
    if type == 'ultimate rare':
      return UltimateRare()
    elif type == 'super rare':
      return SuperRare()
    elif type == 'rare':
      return Rare()
    assert 0, type + 'ini tidak valid'
```

3. Prototype

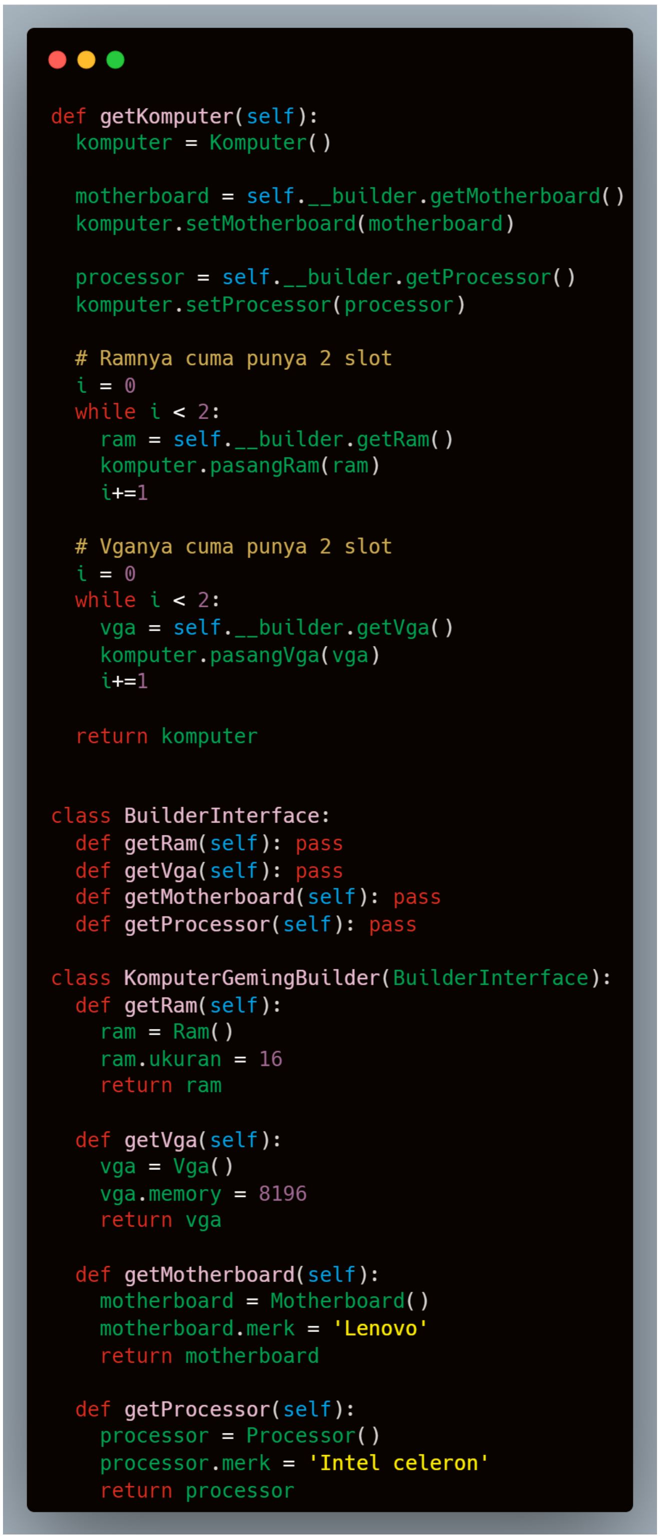


```python
from copy import deepcopy

class Point:
  def __init__(self, x, y):
    self.x = x
    self.y = y

  def __str__(self):
    print("({}, {})".format(self.x, self.y))

  def move(self, x, y):
    self.x += x
    self.y += y

  def clone(self, move_x, move_y):
    obj = deepcopy(self)
    obj,move(move_x, move_y)
    return obj
```

4. Builder



```python
class Komputer:
  def __init__(self):
    self.__ram = list()
    self.__processor = None
    self.__vga = list()
    self.__motherboard = None

  def setMotherboard(self, motherboard):
    self.__motherboard = motherboard

  def setProcessor(self, processor):
    self.__processor.append(processor)

  def pasangRam(self, ram):
    self.__ram.append(ram)

  def pasangVga(self, vga):
    self.__vga.append(vga)

  def spesifikasi(self):
    print("Motherboard: %s" % self.__motherboard.merk)
    print("Processor: %s" % self.processor.kecepatan)
    print("Ram: ")
    for i in __ram:
      print(i.ukuran + ',')
    print("VGA: ")
    for i in __vga:
      print(i.memory + ',')

  def clone(self):
    return deepcopy(self)

class Ram:
  ukuran = None

class Processor:
  kecepatan = None

class Vga:
  memory = None

class Motherboard:
  merk = None


class Director:
  __builder = None

  def setBuilder(self, builder):
    self.__builder = builder
```



```python
def getKomputer(self):
  komputer = Komputer()

  motherboard = self.__builder.getMotherboard()
  komputer.setMotherboard(motherboard)

  processor = self.__builder.getProcessor()
  komputer.setProcessor(processor)

  # Ramnya cuma punya 2 slot
  i = 0
  while i < 2:
    ram = self.__builder.getRam()
    komputer.pasangRam(ram)
    i+=1

  # Vganya cuma punya 2 slot
  i = 0
  while i < 2:
    vga = self.__builder.getVga()
    komputer.pasangVga(vga)
    i+=1

  return komputer


class BuilderInterface:
  def getRam(self): pass
  def getVga(self): pass
  def getMotherboard(self): pass
  def getProcessor(self): pass

class KomputerGemingBuilder(BuilderInterface):
  def getRam(self):
    ram = Ram()
    ram.ukuran = 16
    return ram

  def getVga(self):
    vga = Vga()
    vga.memory = 8196
    return vga

  def getMotherboard(self):
    motherboard = Motherboard()
    motherboard.merk = 'Lenovo'
    return motherboard

  def getProcessor(self):
    processor = Processor()
    processor.merk = 'Intel celeron'
    return processor
```