Bahan Ajar

Chapter 12

Materi Pembelajaran

Matakuliah :

INTERFACING PERIPHERAL

Kode Matakuliah : SKO 20416

Prodi : SISTEM KOMPUTER

Dosen Pengampu Matakuliah:

Bayu Nugroho, S.Kom., M.Eng

Chapter 12

Tables of Content

Python Classes and Objects
- Create a Class
- Create Object
- The __init__() Function
- Object Methods
- Modify, Delete Object Properties
- The pass Statement

Tugas Mandiri

Python Classes and Objects

Python is an object oriented programming language. Almost everything in Python is an object, with its properties and methods. A Class is like an object constructor, or a "blueprint" for creating objects.

Create a Class

To create a class, use the keyword class:

Example

Create a class named MyClass, with a property named x:

```
class MyClass:
  x = 5
```

Create Object

Now we can use the class named MyClass to create objects:

Example

Create an object named p1, and print the value of x:

```
p1 = MyClass()
print(p1.x)
```

The __init__() Function

The examples above are classes and objects in their simplest form, and are not really useful in real life applications.

To understand the meaning of classes we have to understand the built-in __init__() function.

All classes have a function called __init__(), which is always executed when the class is being initiated.

Use the __init__() function to assign values to object properties, or other operations that are necessary to do when the object is being created:

The __init__() Function

Create a class named Person, use the __init__() function to assign values for name and age:

Example

Create a class named Person, use the __init__() function to assign values for name and age:

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

Object Methods

Objects can also contain methods. Methods in objects are functions that belong to the object.

Let us create a method in the Person class:

Example

Insert a function that prints a greeting, and execute it on the p1 object:

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name)

p1 = Person("John", 36)
p1.myfunc()
```

Modify Object Properties

You can modify properties on objects like this:

Example

Set the age of p1 to 40:

```
p1.age = 40
```

Delete Object Properties

You can delete properties on objects by using the del keyword:

Example

Delete the age property from the p1 object:

```
del p1.age
```

The pass Statement

class definitions cannot be empty, but if you for some reason have a class definition with no content, put in the pass statement to avoid getting an error.

Example

```
class Person:
  pass
```

Tugas Mandiri (teori):

1. Does Python support class inheritance? Explained
2. Which method should you define to create default values in a class constructor?
   a. __del__()
   b. __init__()
   c. init() d. set_init()

Tugas Mandiri (prakt):

Tuliskan script program python dengan fungsi **Classes and Objects** untuk menghidupkan 6 buah LED di Raspberry Pi.

end