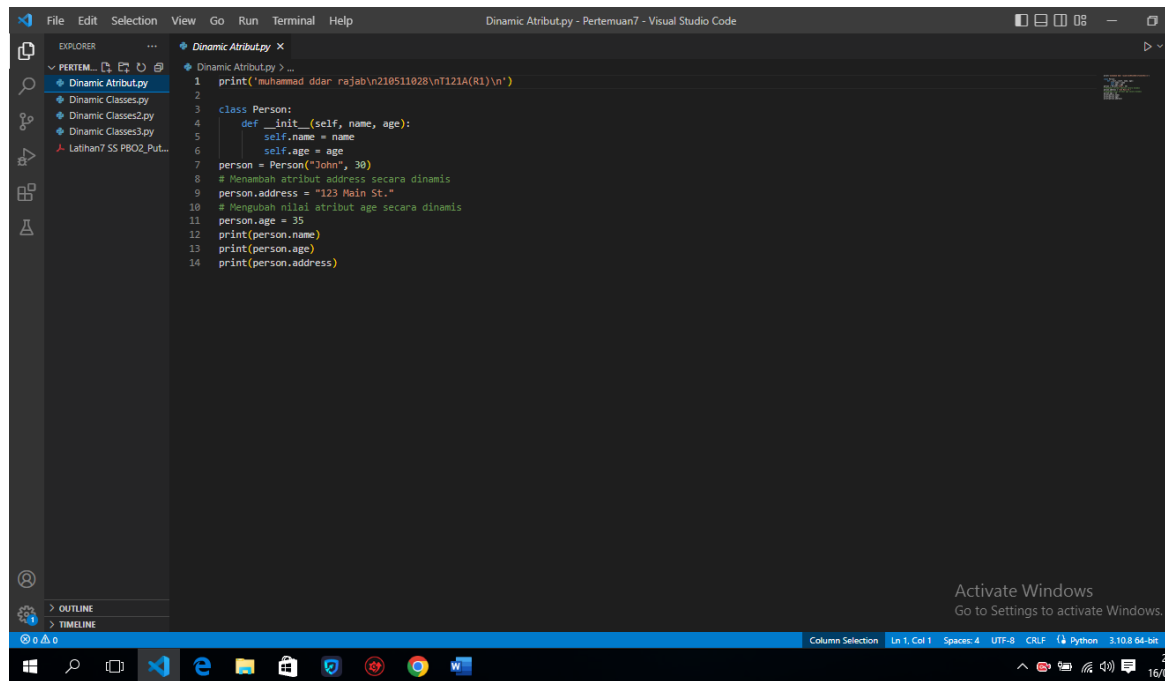


nama : muhammad iddar rajab
kelas : R1
nim : 210511028

1. dinamicAtribut

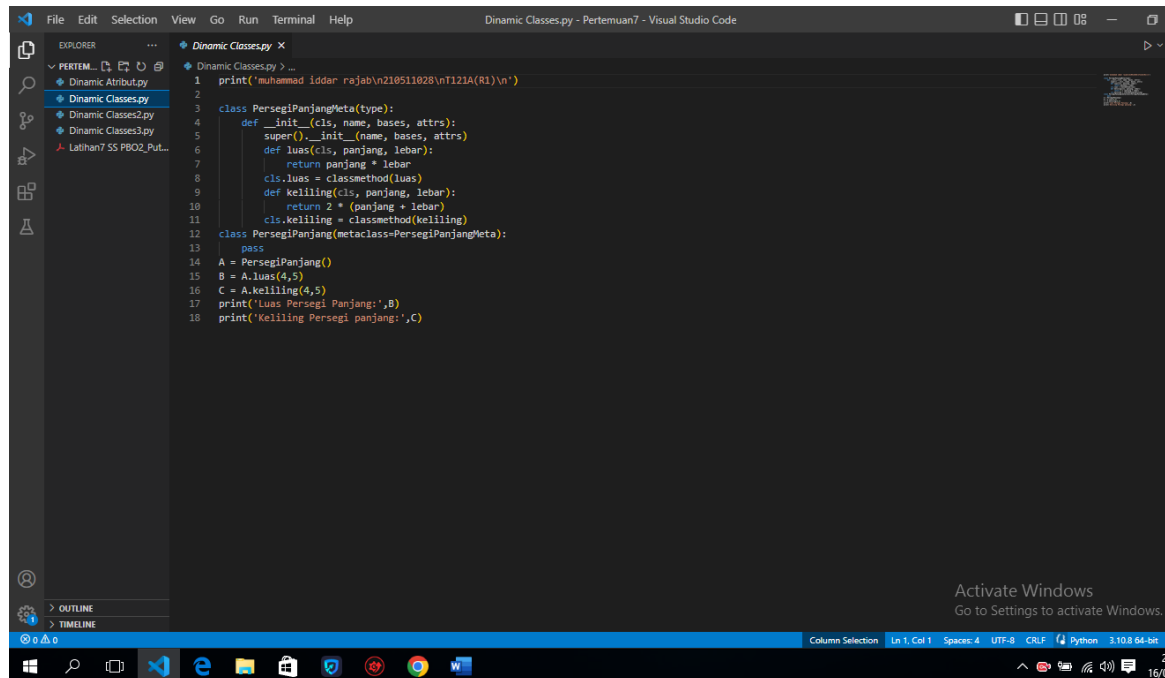


The screenshot shows a Visual Studio Code editor window titled 'Dynamic Atribut.py - Pertemuan7 - Visual Studio Code'. The Explorer sidebar on the left shows a project named 'PERTEMUAN7' with files 'Dynamic Atribut.py', 'Dynamic Classes.py', 'Dynamic Classes2.py', 'Dynamic Classes3.py', and 'Latihan7 SS PBO2 Put...'. The main editor displays the code for 'Dynamic Atribut.py' with the following content:

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2
3 class Person:
4     def __init__(self, name, age):
5         self.name = name
6         self.age = age
7
8 person = Person("John", 30)
9 # Menambah atribut address secara dinamis
10 person.address = "123 Main St."
11 # Mengubah nilai atribut age secara dinamis
12 person.age = 35
13 print(person.name)
14 print(person.age)
15 print(person.address)
```

The status bar at the bottom indicates 'Column Selection', 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', and '3.10.8 64-bit'.

2. dinamicClasses

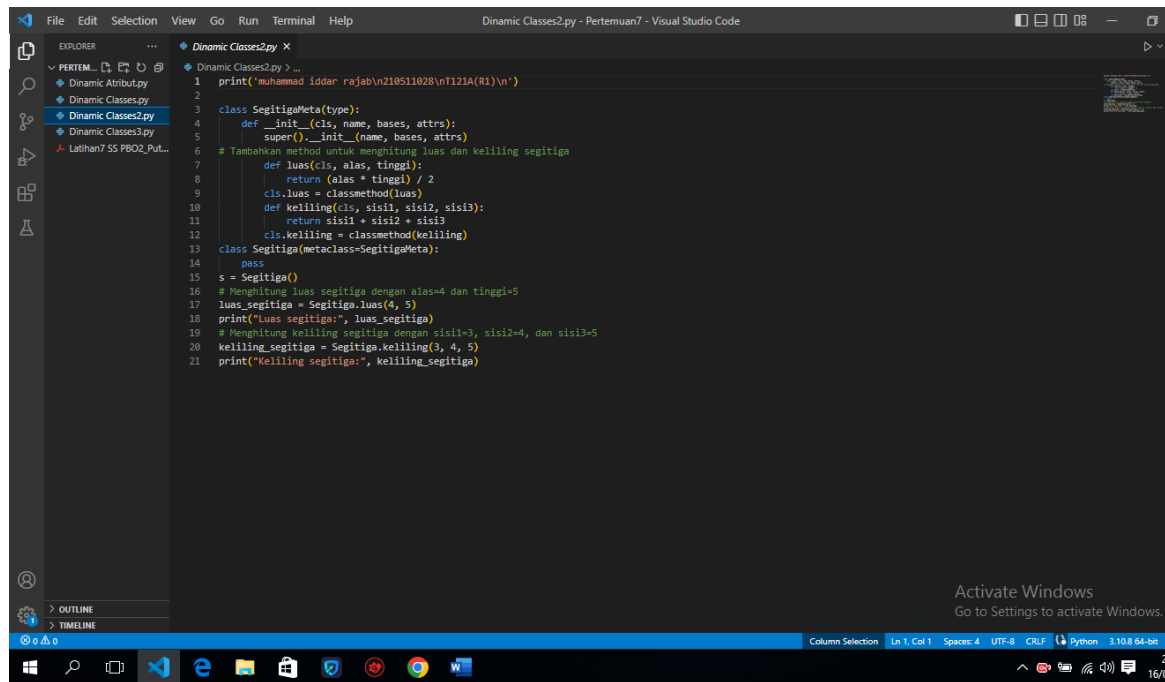


The screenshot shows a Visual Studio Code editor window titled 'Dynamic Classes.py - Pertemuan7 - Visual Studio Code'. The Explorer sidebar on the left shows the same project structure as the previous screenshot. The main editor displays the code for 'Dynamic Classes.py' with the following content:

```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2
3 class PersegiPanjangMeta(type):
4     def __init__(cls, name, bases, attrs):
5         super().__init__(name, bases, attrs)
6         def luas(cls, panjang, lebar):
7             return panjang * lebar
8         cls.luas = classmethod(luas)
9         def keliling(cls, panjang, lebar):
10             return 2 * (panjang + lebar)
11         cls.keliling = classmethod(keliling)
12
13 class PersegiPanjang(metaclass=PersegiPanjangMeta):
14     pass
15
16 A = PersegiPanjang()
17 B = A.luas(4,5)
18 C = A.keliling(4,5)
19 print('Luas Persegi Panjang:',B)
20 print('Keliling Persegi panjang:',C)
```

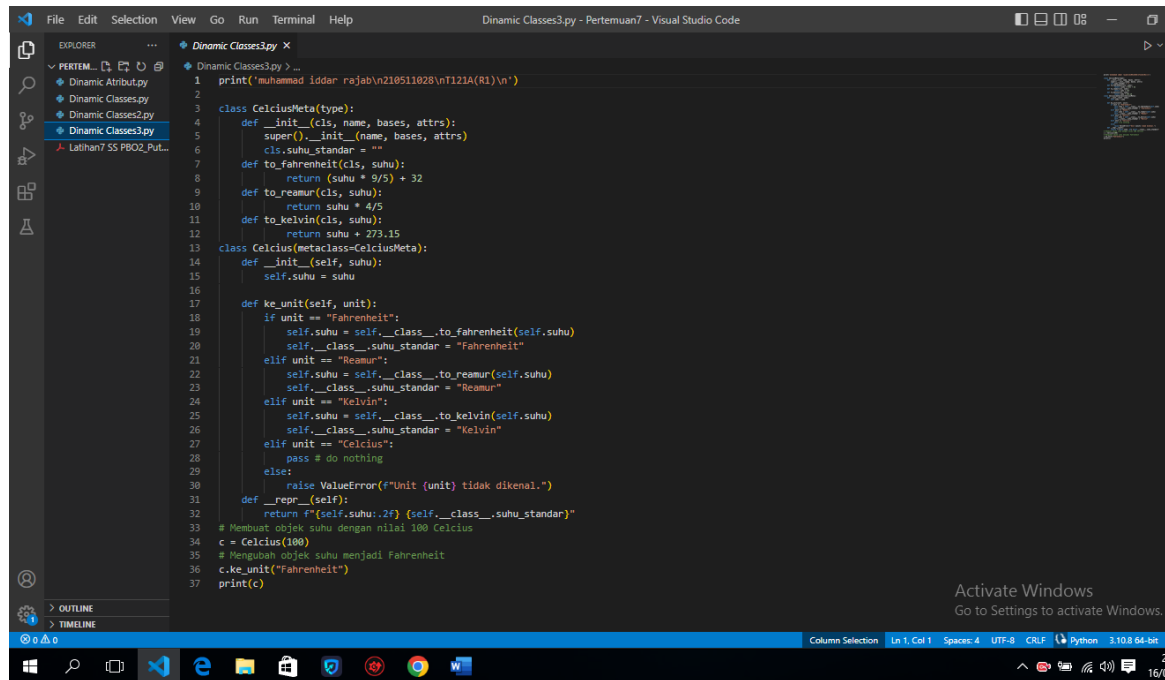
The status bar at the bottom indicates 'Column Selection', 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', and '3.10.8 64-bit'.

3. dinamicClasses2



```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2
3 class SegitigaMeta(type):
4     def __init__(cls, name, bases, attrs):
5         super().__init__(name, bases, attrs)
6         # Tambahkan method untuk menghitung luas dan keliling segitiga
7         def luas(cls, alas, tinggi):
8             return (alas * tinggi) / 2
9         cls.luas = classmethod(luas)
10        def keliling(cls, sisi1, sisi2, sisi3):
11            return sisi1 + sisi2 + sisi3
12        cls.keliling = classmethod(keliling)
13    class Segitiga(metaclass=SegitigaMeta):
14        pass
15    s = Segitiga()
16    # Menghitung luas segitiga dengan alas=4 dan tinggi=5
17    luas_segitiga = Segitiga.luas(4, 5)
18    print("Luas segitiga:", luas_segitiga)
19    # Menghitung keliling segitiga dengan sisi1=3, sisi2=4, dan sisi3=5
20    keliling_segitiga = Segitiga.keliling(3, 4, 5)
21    print("Keliling segitiga:", keliling_segitiga)
```

4. dinamicClasses3



```
1 print('muhammad iddar rajab\n210511028\nT121A(R1)\n')
2
3 class CelciusMeta(type):
4     def __init__(cls, name, bases, attrs):
5         super().__init__(name, bases, attrs)
6         cls.suhu_standar = ""
7         def to_fahrenheit(cls, suhu):
8             return (suhu * 9/5) + 32
9         def to_reamur(cls, suhu):
10            return suhu * 4/5
11        def to_kelvin(cls, suhu):
12            return suhu + 273.15
13    class Celcius(metaclass=CelciusMeta):
14        def __init__(self, suhu):
15            self.suhu = suhu
16
17        def ke_unit(self, unit):
18            if unit == "Fahrenheit":
19                self.suhu = self.__class__.to_fahrenheit(self.suhu)
20                self.__class__.suhu_standar = "Fahrenheit"
21            elif unit == "Reamur":
22                self.suhu = self.__class__.to_reamur(self.suhu)
23                self.__class__.suhu_standar = "Reamur"
24            elif unit == "Kelvin":
25                self.suhu = self.__class__.to_kelvin(self.suhu)
26                self.__class__.suhu_standar = "Kelvin"
27            elif unit == "Celcius":
28                pass # do nothing
29            else:
30                raise ValueError("Unit (unit) tidak dikenal.")
31        def __repr__(self):
32            return f"{self.suhu:.2f} ({self.__class__.suhu_standar})"
33    # Membuat objek suhu dengan nilai 100 Celcius
34    c = Celcius(100)
35    # Mengubah objek suhu menjadi Fahrenheit
36    c.ke_unit("Fahrenheit")
37    print(c)
```