**POLITEKNIK NEGERI SEMARANG JURUSAN TEKNIK ELEKTRO**

# PROGRAM STUDI STR TEKNOLOGI REKAYASA KOMPUTER

**JOBSHEET 02:**

**DASAR KELAS DAN FUNGSI PADA BAHASA PEMROGRAMAN PYTHON**



**Dosen:**

Ir. Prayitno, S.ST., M.T., Ph.D.

**Nama Mahasiswa :** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**NIM Mahasiswa :** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tahun Akademik 2025**

## A. Tujuan Instruksional Khusus

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Memahami dan mengimplementasikan fungsi dalam Python, termasuk parameter dan return value.
2. Menggunakan konsep modularitas dalam pemecahan masalah dengan mendefinisikan fungsi.
3. Mendeklarasikan kelas dalam Python.
4. Menggunakan atribut dan method dalam kelas.

## B. Alat dan Bahan

**Perangkat Lunak**:

* Sistem operasi Windows/Linux/macOS (sesuai ketersediaan).
* Python 3.x (versi terbaru yang stabil).
* IDE atau Code Editor (misalnya, PyCharm, VS Code, atau IDLE). **Perangkat Keras**:
* Komputer/Laptop dengan spesifikasi memadai untuk menjalankan Python. **Bahan Pendukung**:
* Modul/slide perkuliahan yang menjelaskan dasar pemrograman Python.
* Koneksi internet (opsional, untuk referensi tambahan).

**C. Dasar Teori**

## 1. Fungsi dalam Python

Fungsi adalah blok kode yang digunakan untuk melakukan tugas tertentu dan dapat digunakan kembali dalam program. Fungsi dalam Python dideklarasikan dengan def, memiliki parameter opsional, dan dapat mengembalikan nilai menggunakan return. Contoh fungsi sederhana:

def tambah(a, b): return a + b

hasil = tambah(3, 5)

print(hasil) # Output: 8

## 2. Modularitas

Modularitas memungkinkan pemrograman yang lebih terstruktur dengan memecah kode menjadi bagian-bagian kecil yang lebih mudah dikelola dan dipahami. Modularisasi: Membagi kode menjadi beberapa modul dengan fungsi atau kelas tertentu. Modul: Dalam Python, modul adalah file yang berisi kode yang dapat digunakan kembali. Modul dapat berupa fungsi, kelas, atau variabel yang diorganisir dalam sebuah file Python.

## 3. Kelas dan Objek dalam Python

Kelas adalah blueprint untuk membuat objek. Objek adalah instance dari kelas.

Deklarasi kelas sederhana:

class Mahasiswa:

|  |
| --- |
| def \_\_init\_\_(self, nama, nim):  self.nama = nama self.nim = nim  def tampilkan\_info(self):  print(f"Nama: {self.nama}, NIM: {self.nim}") |

## 4. Atribut dan Method

Atribut adalah variabel yang terkait dengan objek, sedangkan method adalah fungsi yang dideklarasikan di dalam kelas dan bekerja pada objek tersebut.

Perbedaan fungsi dan method:

* **Fungsi**: Berdiri sendiri, tidak terikat dengan objek.
* **Method**: Dideklarasikan di dalam kelas dan beroperasi pada objek kelas tersebut.

Contoh method dalam kelas:

|  |
| --- |
| class Persegi: def \_\_init\_\_(self, sisi):  self.sisi = sisi  def luas(self):  return self.sisi \*\* 2    p = Persegi(4)  print(p.luas()) # Output: 16 |

**D. Langkah Praktikum**

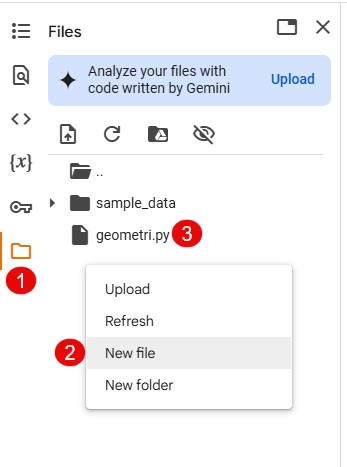
## Praktikum 1: Membuat dan Menggunakan Fungsi

Kode Program

|  |
| --- |
| 01: # Fungsi Built-in  02: # Fungsi print() adalah contoh fungsi built-in untuk mencetak output  03: print("Ini adalah contoh fungsi built-in") 04:  05: # Fungsi len() adalah contoh fungsi built-in untuk menghitung panjang suatu objek  06: kata = "Pemrograman"  07: panjang\_kata = len(kata)  08: print(f"Panjang kata '{kata}' adalah: {panjang\_kata}") 09:  10: # Fungsi max() adalah contoh fungsi built-in untuk mencari nilai maksimum dari sebuah daftar  11: angka = [10, 5, 30, 40, 25]  12: nilai\_max = max(angka)  13: print(f"Nilai maksimum dalam daftar {angka} adalah: {nilai\_max}")  14:  15: # Fungsi User-Defined  16: # Fungsi dengan satu parameter (Menerima satu nilai input)  17: def cetak\_kuadrat(angka):  18: # Menghitung kuadrat dari angka yang diterima sebagai parameter |
| 19: print(f"Kuadrat dari {angka} adalah: {angka \*\* 2}") 20:  21: # Fungsi dengan beberapa parameter (Menerima lebih dari satu nilai input) 22: def hitung\_luas\_persegi\_panjang(panjang, lebar):  23: # Menghitung luas persegi panjang 24: return panjang \* lebar  25:  26: # Fungsi dengan beberapa tipe parameter (Menerima berbagai jenis data)  27: def info\_mahasiswa(nama, umur, ipk):  28: # Mencetak informasi mahasiswa  29: print(f"Nama: {nama}, Umur: {umur}, IPK: {ipk}")  30:  31: # Fungsi tanpa return value (Non-return value)  32: def sapa\_pengguna(nama):  33: # Fungsi ini hanya mencetak sapaan tanpa mengembalikan nilai  34: print(f"Halo, {nama}! Selamat datang di dunia Python.")  35:  36: # Fungsi dengan return value  37: def hitung\_keliling\_persegi(sisi):  38: # Mengembalikan keliling persegi 39: return 4 \* sisi  40:  41: # Pemanggilan fungsi dengan satu parameter 42: cetak\_kuadrat(5)  43:  44: # Pemanggilan fungsi dengan beberapa parameter  45: luas = hitung\_luas\_persegi\_panjang(10, 5) 46: print(f"Luas persegi panjang: {luas}")  47:  48: # Pemanggilan fungsi dengan beberapa tipe parameter 49: info\_mahasiswa("Budi", 22, 3.8)  50:  51: # Pemanggilan fungsi tanpa return value 52: sapa\_pengguna("Andi")  53:  54: # Pemanggilan fungsi dengan return value  55: keliling = hitung\_keliling\_persegi(5)  56: print(f"Keliling persegi dengan sisi 5 adalah: {keliling}") 57: |

## Praktikum 2: Modularisasi Kode dengan Fungsi

1. Pisahkan fungsi perhitungan luas dan keliling ke dalam modul geometri.py.
2. Buat file Python pada google colab dengan cara 1) klik tombol files pada menu google colab, 2) tekan tombol new file, 3) ubah nama file menjadi geometri.py



1. Masukkan kode program berikut pada geometri.py

|  |
| --- |
| 01: # Modul geometri.py berisi fungsi terkait geometri  02:  03: def hitung\_luas\_persegi(sisi): 04: return sisi \* sisi  05:  06: def hitung\_luas\_persegi\_panjang(panjang, lebar): 07: return panjang \* lebar  08:  09: def hitung\_luas\_lingkaran(jari\_jari): 10: return 3.14 \* (jari\_jari \*\* 2)  11:  12: def hello\_pray():  13: return "Echo hallo pray" |

1. Pada file utama ketikkan kode berikut.

Kode Program

|  |
| --- |
| 01: # File utama main.py yang mengimpor modul geometri.py  02:  03: # Mengimpor modul geometri 04: import geometri  05:  06: def main():  07: # Menggunakan fungsi dari modul meong  08: luas\_persegi = meong.hitung\_luas\_persegi(5)  09: print(f"Luas persegi dengan sisi 5 adalah: {luas\_persegi}")  10:  11: luas\_persegi\_panjang = meong.hitung\_luas\_persegi\_panjang(10, 5) 12: print(f"Luas persegi panjang dengan panjang 10 dan lebar 5 adalah:  {luas\_persegi\_panjang}")  13:  14: luas\_lingkaran = meong.hitung\_luas\_lingkaran(7)  15: print(f"Luas lingkaran dengan jari-jari 7 adalah: {luas\_lingkaran}") 16:  17: if \_\_name\_\_ == "\_\_main\_\_": 18: main()  19: |

## Praktikum 3: Membuat Kelas Sederhana

Kode Program

|  |
| --- |
| 01: # Kelas Buku untuk merepresentasikan buku di perpustakaan  02: class Buku:  03: def \_\_init\_\_(self, judul, pengarang, tahun\_terbit):  04: self.judul = judul  05: self.pengarang = pengarang  06: self.tahun\_terbit = tahun\_terbit  07: self.status = "Tersedia" # Status buku, default adalah Tersedia 08:  09: def tampilkan\_info(self):  10: print(f"Judul: {self.judul}")  11: print(f"Pengarang: {self.pengarang}")  12: print(f"Tahun Terbit: {self.tahun\_terbit}") 13: print(f"Status: {self.status}")  14:  15: def pinjam(self):  16: if self.status == "Tersedia":  17: self.status = "Dipinjam"  18: print(f"Buku '{self.judul}' telah dipinjam.")  19: else:  20: print(f"Buku '{self.judul}' sedang dipinjam.")  21:  22: def kembalikan(self):  23: if self.status == "Dipinjam":  24: self.status = "Tersedia"  25: print(f"Buku '{self.judul}' telah dikembalikan.")  26: else:  27: print(f"Buku '{self.judul}' tidak sedang dipinjam.")  28:  29: # Membuat objek dari kelas Buku |
| 30: buku1 = Buku("Pemrograman Python", "John Doe", 2021)  31: buku2 = Buku("Data Science untuk Pemula", "Jane Smith", 2020)  32:  33: # Menggunakan metode objek Buku  34: buku1.tampilkan\_info() 35: buku2.pinjam()  36:  37: # Mengubah status buku dan menampilkan informasi  38: buku2.kembalikan()  39: buku1.pinjam()  40: buku1.tampilkan\_info()  41: 42:  43: # Kelas Mahasiswa untuk merepresentasikan Mahasiswa dalam kelas  44:  45: # Definisi kelas 46: class Mahasiswa:  47: # Konstruktor (\_\_init\_\_) untuk menginisialisasi atribut objek  48: def \_\_init\_\_(self, nama, nim, umur):  49: self.nama = nama # Atribut objek nama  50: self.nim = nim # Atribut objek nim  51: self.umur = umur # Atribut objek umur  52:  53: # Metode untuk menampilkan informasi mahasiswa  54: def tampilkan\_info(self):  55: print(f"Nama: {self.nama}")  56: print(f"NIM: {self.nim}")  57: print(f"Umur: {self.umur} tahun")  58:  59: # Metode untuk merubah umur mahasiswa  60: def ubah\_umur(self, umur\_baru): 61: self.umur = umur\_baru  62:  63: # Membuat objek (instance) dari kelas Mahasiswa  64: mahasiswa1 = Mahasiswa("Andi", "12345", 20)  65: mahasiswa2 = Mahasiswa("Budi", "67890", 22)  66:  67: # Menggunakan metode dari kelas Mahasiswa  68: mahasiswa1.tampilkan\_info()  69: print() # Baris kosong 70: mahasiswa2.tampilkan\_info()  71:  72: # Mengubah umur mahasiswa1 menggunakan metode ubah\_umur  73: mahasiswa1.ubah\_umur(21)  74:  75: # Menampilkan info mahasiswa setelah umur diubah  76: print("\nSetelah mengubah umur mahasiswa1:") 77: mahasiswa1.tampilkan\_info() 78: |

**Praktikum 4: Membuat dan Menggunakan Method dalam Kelas** Kode Program:

|  |
| --- |
| 01: # Kelas untuk menjelaskan atribut dan metode dalam kelas  02: class Mobil:  03: def \_\_init\_\_(self, merk, warna, tahun, harga): 04: # Atribut yang dimiliki oleh objek Mobil  05: self.merk = merk  06: self.warna = warna  07: self.tahun = tahun  08: self.harga = harga  09:  10: # Fungsi tanpa return value, hanya mencetak informasi 11: def tampilkan\_info(self):  12: print(f"Mobil {self.merk} berwarna {self.warna}, tahun  {self.tahun}, harga: Rp {self.harga}")  13:  14: # Fungsi dengan satu parameter 15: def diskon(self, persen\_diskon):  16: # Menghitung harga setelah diskon  17: diskon\_harga = self.harga \* (persen\_diskon / 100)  18: harga\_setelah\_diskon = self.harga - diskon\_harga  19: print(f"Harga setelah diskon {persen\_diskon}%: Rp {harga\_setelah\_diskon}")  20: # Tidak mengembalikan nilai, hanya mencetak harga setelah diskon 21:  22: # Fungsi dengan return value, menghitung usia mobil berdasarkan tahun 23: def hitung\_usia(self, tahun\_sekarang):  24: usia = tahun\_sekarang - self.tahun 25: return usia  26:  27: # Fungsi dengan beberapa parameter  28: def perbarui\_harga(self, harga\_baru, tahun\_baru):  29: self.harga = harga\_baru  30: self.tahun = tahun\_baru  31: print(f"Harga dan tahun mobil {self.merk} diperbarui menjadi Rp  {self.harga} dan tahun {self.tahun}")  32:  33: # Membuat objek mobil  34: mobil1 = Mobil("Toyota", "Hitam", 2015, 300000000) 35: mobil2 = Mobil("Honda", "Merah", 2018, 250000000)  36:  37: # Menggunakan metode tanpa return value  38: mobil1.tampilkan\_info()  39: mobil2.tampilkan\_info()  40:  41: # Menggunakan metode dengan satu parameter (diskon)  42: mobil1.diskon(10)  43: mobil2.diskon(15)  44:  45: # Menggunakan metode dengan return value (hitung usia mobil)  46: usia\_mobil1 = mobil1.hitung\_usia(2025)  47: usia\_mobil2 = mobil2.hitung\_usia(2025)  48:  49: print(f"Usia mobil1 pada tahun 2025: {usia\_mobil1} tahun") 50: print(f"Usia mobil2 pada tahun 2025: {usia\_mobil2} tahun") 51:  52: # Menggunakan metode dengan beberapa parameter (perbarui harga dan tahun)  53: mobil1.perbarui\_harga(280000000, 2022)  54: mobil2.perbarui\_harga(240000000, 2021) |

## E. Hasil Praktikum

Lengkapi hasil tabel praktikum berikut:

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama Praktikum** | **Hasil Praktikum** |
| 1 | Praktikum 01. Fungsi pada Python | praktikum 1  **HASIL :** |
| 2 | Praktikum 02: Modularisasi | modul_geometri  praktikum 2  **HASIL :** |
| 3 | Praktikum 03: Kelas dan Objek | praktikum 3  **HASIL :** |
| 4 | Praktikum 04: membuat dan menggunakan atribut dan  fungsi di dalam kelas | praktikum 4  **HASIL :** |

## F. Penugasan

1. Kumpulkan Laporan Praktikum dari jobsheet ini dalam bentuk Microsoft word sesuai dengan format jobsheet praktikum dan dikumpulkan di web LMS. (JANGAN DALAM BENTUK PDF)
2. Kumpulkan luaran kode praktikum dalam bentuk ipynb yang sudah diunggah pada akun github masing-masing. Lampirkan tautan github yang sudah di unggah melalui laman LMS.
3. **Buat Program**: Lihat kembali praktikum 02. Modularisasi dan 03. Kelas sederahan. Buat program sederhana menggunakan dua konsep diatas tentang (Pilih salah satu: A. Penjualan HP, B. jadwal kelas, C. Parkir roda dua).

## G. Kesimpulan

Kesimpulan dari jobsheet ini adalah bahwa pemrograman berorientasi objek (PBO) memungkinkan pengorganisasian kode yang lebih baik dengan menggunakan konsep-konsep seperti kelas, objek, atribut, dan metode. Dengan memahami bagaimana mendefinisikan atribut dalam kelas, menggunakan fungsi baik dengan parameter maupun return value, serta membangun objek untuk melakukan operasi, mahasiswa dapat merancang dan mengelola aplikasi secara modular dan efisien. Penugasan ini bertujuan untuk memberikan pemahaman yang mendalam mengenai penerapan konsep PBO dalam konteks dunia nyata, seperti manajemen buku di perpustakaan atau pengelolaan akun bank.

## H. Daftar Pustaka

1. Lutz, M. (2013). *Learning Python*. O'Reilly Media.
2. Guttag, J. V. (2016). *Introduction to Computation and Programming Using Python*. MIT Press.
3. Python Software Foundation. *Python 3 Documentation*. <https://docs.python.org/3/>