

**LAPORAN UTS**

**PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)**



**2411102441207 - Rivoldy Makatita**

**2411102441249 - Hervino islami fasha**

**2411102441204 Radittyta - Nanda Saputra**

**2411102441285 - Hafidzal Mufty**

**2411102441221 - Athaya Hassya Fausta**

**2411102441209 - Andi Fathur Rahman Ismail**

**2411102441242 - Muhammad Rifqi Ihsan**

**KELOMPOK 12**

**FAKULTAS SAINS DAN TEKNOLOGI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR**

## BAB I

### PENDAHULUAN

#### A. Latar Belakang

Sektor layanan penyewaan (rental) perangkat konsol telah mengalami pertumbuhan signifikan seiring meningkatnya popularitas *video game* modern. Operasional manajemen rental yang masih dilakukan secara manual sering kali menimbulkan masalah efisiensi, seperti kesalahan pencatatan inventaris, kesulitan pelacakan transaksi, dan integritas data yang rendah.

Oleh karena itu, diperlukan pengembangan Sistem Manajemen Rental Perangkat Konsol yang terkomputerisasi. Sistem ini akan dibangun menggunakan paradigma Pemrograman Berorientasi Objek (PBO) untuk memastikan kode yang dihasilkan terstruktur, modular, dan mudah dikembangkan. Proyek ini dibagi menjadi tiga tahapan yang saling berkesinambungan, yang secara progresif menerapkan konsep PBO, mulai dari struktur dasar hingga integrasi pustaka standar:

1. Project 1: Implementasi Struktur Dasar dan Enkapsulasi.
2. Project 2: Implementasi *Inheritance* dan *Polymorphism*.
3. Project 3: Implementasi Modularisasi dan Integrasi Pustaka Standar.

#### B. Tujuan

- Menerapkan konsep Object-Oriented Programming (OOP) dalam bentuk sistem nyata.
- Mengelola data perangkat konsol, pelanggan, dan transaksi penyewaan secara efisien.
- Memahami penerapan enkapsulasi, inheritance, dan polymorphism dalam pengembangan sistem.
- Mengimplementasikan modularisasi untuk memisahkan logika program ke dalam beberapa file (modul) agar mudah dikelola.
- Menggunakan pustaka standar Python untuk membantu proses penyimpanan data dan pengelolaan waktu penyewaan.

#### C. Tinjauan Pustaka

Pemrograman berorientasi objek (Object-Oriented Programming/OOP) merupakan paradigma pemrograman yang berfokus pada konsep objek yang memiliki atribut dan perilaku. Pendekatan ini mempermudah pengembang dalam membangun sistem yang terstruktur, efisien, dan mudah dipelihara (Booch, 2007).

Salah satu konsep utama dalam OOP adalah **enkapsulasi**, yaitu proses menyembunyikan data agar tidak diakses secara langsung oleh pengguna. Dengan konsep ini, data pelanggan dan perangkat dalam sistem manajemen rental dapat terlindungi dari perubahan yang tidak diinginkan (Wirth, 1996).

Konsep **inheritance** dan **polymorphism** juga berperan penting dalam pengembangan sistem. Inheritance memungkinkan kelas turunan mewarisi atribut dari kelas induk, sedangkan polymorphism membuat metode dapat memiliki perilaku berbeda sesuai dengan objeknya (Stroustrup, 2013).

Selain itu, **modularisasi** digunakan agar program terbagi dalam beberapa bagian seperti pengelolaan pelanggan, perangkat, dan transaksi. Pendekatan ini memudahkan pemeliharaan dan pengembangan sistem (Pressman, 2010). Python dipilih karena memiliki sintaks sederhana dan pustaka standar seperti **datetime** dan **json** yang mendukung pengelolaan waktu serta penyimpanan data (Lutz, 2021).

## **BAB II**

### **ALAT DAN BAHAN**

#### **A. Alat**

1. Komputer dan laptop
2. Koneksi internet
3. Visual Studio Code

#### **B. Bahan**

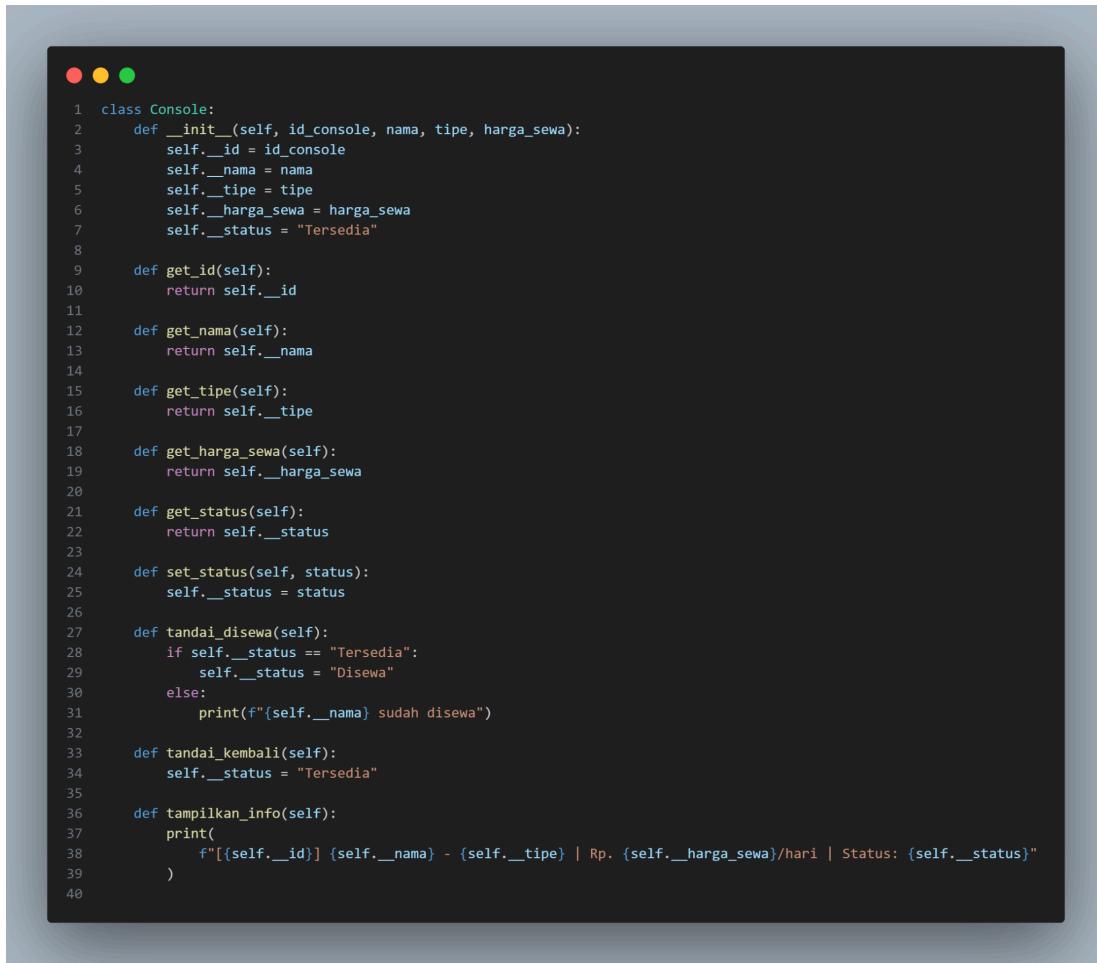
1. Bahasa Pemrograman Python

## BAB III

### PROSEDUR KERJA

#### 1. Project 1 — Struktur Dasar dan Enkapsulasi

##### File console.py



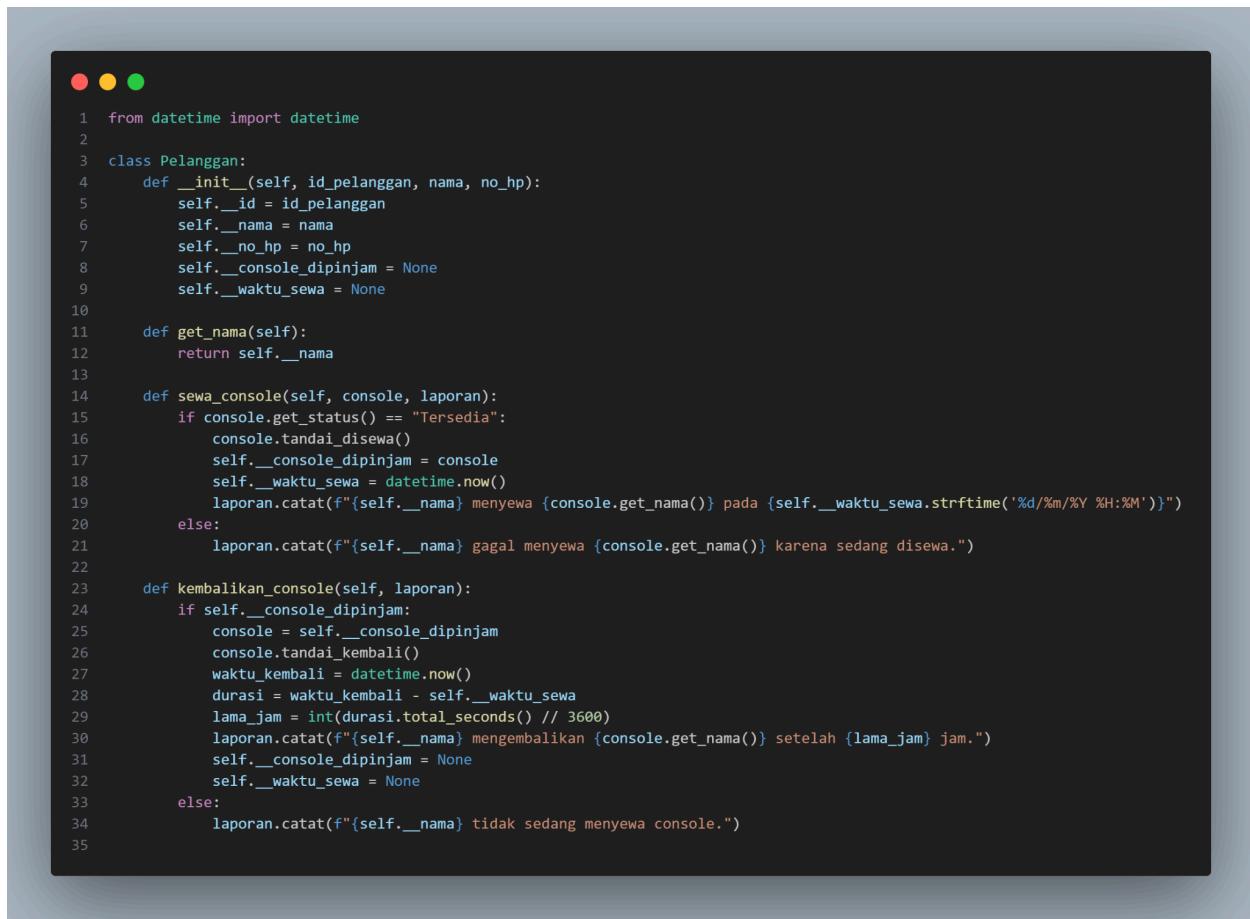
```
1 class Console:
2     def __init__(self, id_console, nama, tipe, harga_sewa):
3         self.__id = id_console
4         self.__nama = nama
5         self.__tipe = tipe
6         self.__harga_sewa = harga_sewa
7         self.__status = "Tersedia"
8
9     def get_id(self):
10        return self.__id
11
12    def get_nama(self):
13        return self.__nama
14
15    def get_tipe(self):
16        return self.__tipe
17
18    def get_harga_sewa(self):
19        return self.__harga_sewa
20
21    def get_status(self):
22        return self.__status
23
24    def set_status(self, status):
25        self.__status = status
26
27    def tandai_disewa(self):
28        if self.__status == "Tersedia":
29            self.__status = "Disewa"
30        else:
31            print(f"{self.__nama} sudah disewa")
32
33    def tandai_kembali(self):
34        self.__status = "Tersedia"
35
36    def tampilkan_info(self):
37        print(
38            f"[{self.__id}] {self.__nama} - {self.__tipe} | Rp. {self.__harga_sewa}/hari | Status: {self.__status}"
39        )
40
```

## File nintendo.py



```
1  from datetime import datetime
2  from models.console import Console
3
4  class Nintendo(Console):
5      def __init__(self, id_console, nama, harga_sewa, warna):
6          super().__init__(id_console, nama, "Nintendo", harga_sewa)
7          self.__warna = warna
8
9      def tampilkan_info(self):
10         waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
11         print(f"[Nintendo] {self.get_nama()} - Warna: {self.__warna} | Rp{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}")
```

## File pelanggan.py



```
1  from datetime import datetime
2
3  class Pelanggan:
4      def __init__(self, id_pelanggan, nama, no_hp):
5          self.__id = id_pelanggan
6          self.__nama = nama
7          self.__no_hp = no_hp
8          self.__console_dipinjam = None
9          self.__waktu_sewa = None
10
11     def get_nama(self):
12         return self.__nama
13
14     def sewa_console(self, console, laporan):
15         if console.get_status() == "Tersedia":
16             console.tandai_disewa()
17             self.__console_dipinjam = console
18             self.__waktu_sewa = datetime.now()
19             laporan.catat(f"{self.__nama} menyewa {console.get_nama()} pada {self.__waktu_sewa.strftime('%d/%m/%Y %H:%M')}")
20         else:
21             laporan.catat(f"{self.__nama} gagal menyewa {console.get_nama()} karena sedang disewa.")
22
23     def kembalikan_console(self, laporan):
24         if self.__console_dipinjam:
25             console = self.__console_dipinjam
26             console.tandai_kembali()
27             waktu_kembali = datetime.now()
28             durasi = waktu_kembali - self.__waktu_sewa
29             lama_jam = int(durasi.total_seconds() // 3600)
30             laporan.catat(f"{self.__nama} mengembalikan {console.get_nama()} setelah {lama_jam} jam.")
31             self.__console_dipinjam = None
32             self.__waktu_sewa = None
33         else:
34             laporan.catat(f"{self.__nama} tidak sedang menyewa console.")
```

## File playstasion.py



```
1  from datetime import datetime
2  from models.console import Console
3
4
5  class Playstation(Console):
6      def __init__(self, id_console, nama, harga_sewa, versi):
7          super().__init__(id_console, nama, "PlayStation", harga_sewa)
8          self.__versi = versi
9
10     def tampilan_info(self):
11         waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
12         print(
13             f"[PS] {self.get_nama()} (Versi {self.__versi}) - Rp{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}"
14         )
15
```

## File xbox.py



```
1  from datetime import datetime
2  from models.console import Console
3
4
5  class Xbox(Console):
6      def __init__(self, id_console, nama, harga_sewa, region):
7          super().__init__(id_console, nama, "Xbox", harga_sewa)
8          self.__region = region
9
10     def tampilan_info(self):
11         waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
12         print(f"[Xbox] {self.get_nama()} (Region {self.__region}) - Rp{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}")
```

Tahap pertama bertujuan membangun fondasi sistem dengan mendefinisikan entitas utama dan melindungi integritas data melalui enkapsulasi.

### A. Struktur Dasar (Kelas dan Objek)

Sistem dibangun dari dua kelas utama yang menjadi fondasi data:

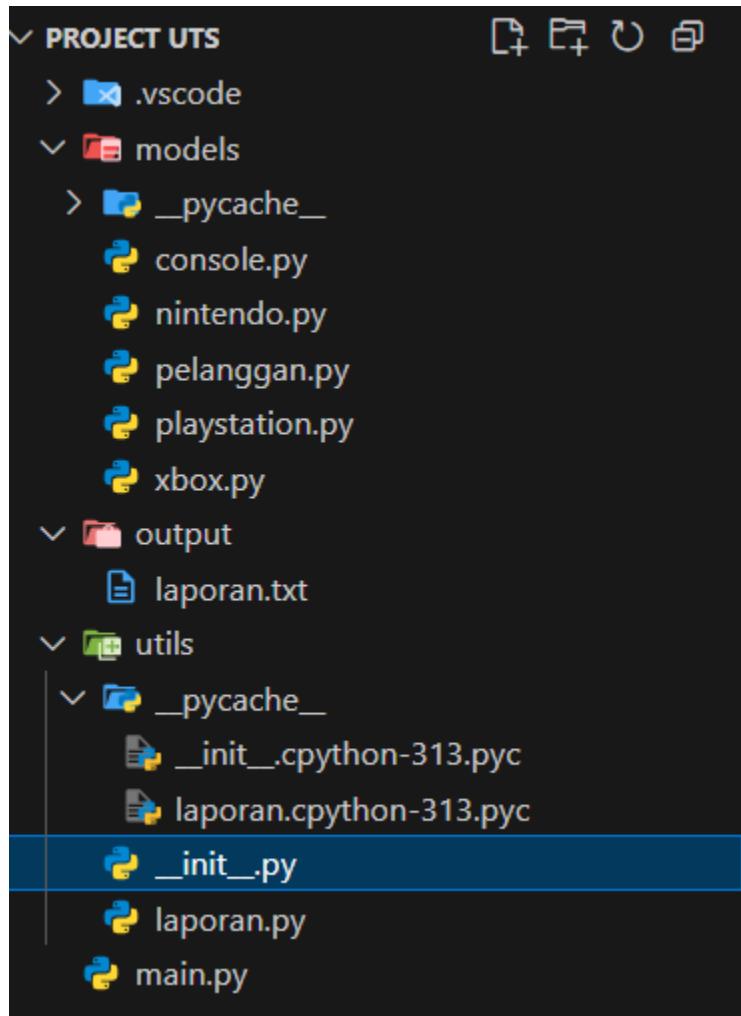
1. class Console: Kelas induk untuk semua perangkat konsol.
2. class Pelanggan : Kelas untuk mengelola data penyewa.

### B. Penerapan Enkapsulasi

Enkapsulasi diterapkan pada kedua kelas utama dengan mendefinisikan semua atribut kunci menggunakan sintaks atribut privat (`__nama_atribut`).

- class Console : Atribut seperti self.\_\_id, self.\_\_nama, self.\_\_harga\_sewa, dan self.\_\_status dilindungi. Akses ke atribut ini hanya diberikan melalui metode *getter* (get\_id, get\_nama, dll.) dan metode *setter* (set\_status). Metode publik seperti tandai\_disewa() juga memiliki logika internal yang mengontrol perubahan status, memastikan status konsol hanya berubah jika memenuhi syarat (if self.\_\_status == "Tersedia":).
- class Pelanggan : Atribut sensitif seperti self.\_\_id, self.\_\_nama, dan status sewa (self.\_\_console\_dipinjam) dibuat privat untuk menjaga kerahasiaan dan integritas data pelanggan.

## 2. Project 2 — Inheritance dan Polymorphism



## File console.py

```
1  class Console:
2      def __init__(self, id_console, nama, tipe, harga_sewa):
3          self.__id = id_console
4          self.__nama = nama
5          self.__tipe = tipe
6          self.__harga_sewa = harga_sewa
7          self.__status = "Tersedia"
8
9      def get_id(self):
10         return self.__id
11
12     def get_nama(self):
13         return self.__nama
14
15     def get_tipe(self):
16         return self.__tipe
17
18     def get_harga_sewa(self):
19         return self.__harga_sewa
20
21     def get_status(self):
22         return self.__status
23
24     def set_status(self, status):
25         self.__status = status
26
27     def tandai_disewa(self):
28         if self.__status == "Tersedia":
29             self.__status = "Disewa"
30         else:
31             print(f"{self.__nama} sudah disewa")
32
33     def tandai_kembali(self):
34         self.__status = "Tersedia"
35
36     def tampilkan_info(self):
37         print(
38             f"[{self.__id}] {self.__nama} - {self.__tipe} | Rp. {self.__harga_sewa}/hari | Status: {self.__status}"
39         )
40
```

## File nintendo.py

```
1  from datetime import datetime
2  from models.console import Console
3
4  class Nintendo(Console):
5      def __init__(self, id_console, nama, harga_sewa, warna):
6          super().__init__(id_console, nama, "Nintendo", harga_sewa)
7          self.__warna = warna
8
9      def tampilkan_info(self):
10         waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
11         print(f"[Nintendo] {self.get_nama()} - Warna: {self.__warna} | Rp.{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}")
```

## File pelanggan.py

```
 1  from datetime import datetime
 2
 3  class Pelanggan:
 4      def __init__(self, id_pelanggan, nama, no_hp):
 5          self.__id = id_pelanggan
 6          self.__nama = nama
 7          self.__no_hp = no_hp
 8          self.__console_dipinjam = None
 9          self.__waktu_sewa = None
10
11      def get_nama(self):
12          return self.__nama
13
14      def sewa_console(self, console, laporan):
15          if console.get_status() == "Tersedia":
16              console.tandai_disewa()
17              self.__console_dipinjam = console
18              self.__waktu_sewa = datetime.now()
19              laporan.catat(f"{self.__nama} menyewa {console.get_nama()} pada {self.__waktu_sewa.strftime('%d/%m/%Y %H:%M')}")
20          else:
21              laporan.catat(f"{self.__nama} gagal menyewa {console.get_nama()} karena sedang disewa.")
22
23      def kembalikan_console(self, laporan):
24          if self.__console_dipinjam:
25              console = self.__console_dipinjam
26              console.tandai_kembali()
27              waktu_kembali = datetime.now()
28              durasi = waktu_kembali - self.__waktu_sewa
29              lama_jam = int(durasi.total_seconds() // 3600)
30              laporan.catat(f"{self.__nama} mengembalikan {console.get_nama()} setelah {lama_jam} jam.")
31              self.__console_dipinjam = None
32              self.__waktu_sewa = None
33          else:
34              laporan.catat(f"{self.__nama} tidak sedang menyewa console.")
35
```

## File playstation.py

```
 1  from datetime import datetime
 2  from models.console import Console
 3
 4
 5  class Playstation(Console):
 6      def __init__(self, id_console, nama, harga_sewa, versi):
 7          super().__init__(id_console, nama, "PlayStation", harga_sewa)
 8          self.__versi = versi
 9
10      def tampilan_info(self):
11          waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
12          print(
13              f"[PS] {self.get_nama()} (Versi {self.__versi}) - Rp{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}"
14          )
15
```

## File xbox.py

```
● ● ●
1  from datetime import datetime
2  from models.console import Console
3
4  class Xbox(Console):
5      def __init__(self, id_console, nama, harga_sewa, region):
6          super().__init__(id_console, nama, "Xbox", harga_sewa)
7          self.__region = region
8
9      def tampilkan_info(self):
10         waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
11         print(f"[Xbox] {self.get_nama()} (Region {self.__region}) - Rp{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}")
```

## File laporan.py

```
● ● ●

1  import os
2  from datetime import datetime
3
4  class Laporan:
5      def __init__(self, nama_file="laporan.txt"):
6          self.nama_file = nama_file
7          self.log_data = []
8
9      def catat(self, pesan):
10         waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
11         log = f"[{waktu}] {pesan}"
12         self.log_data.append(log)
13         print(log)
14
15     def simpan(self):
16         folder = "output"
17         if not os.path.exists(folder):
18             os.mkdir(folder)
19
20         path_file = os.path.join(folder, self.nama_file)
21         with open(path_file, "w", encoding="utf-8") as file:
22             file.write("== LAPORAN AKTIVITAS RENTAL KONSEL ==\n\n")
23             for log in self.log_data:
24                 file.write(log + "\n")
25
26         print(f"laporan tersimpan di: {path_file}")
27
```

### File laporan.txt

```
output > laporan.txt
1 === LAPORAN AKTIVITAS RENTAL KONSOLE ===
2
3 [21/10/2025 21:55] Andi menyewa PlayStation 5 pada 21/10/2025 21:55
4 [21/10/2025 21:55] Budi gagal menyewa PlayStation 5 karena sedang disewa.
5 [21/10/2025 21:55] Andi mengembalikan PlayStation 5 setelah 0 jam.
6 [21/10/2025 21:55] Budi menyewa PlayStation 5 pada 21/10/2025 21:55
7 [21/10/2025 21:55] Budi mengembalikan PlayStation 5 setelah 0 jam.
8
```

### File init.py

```
utils > __init__.py
1 Generate code (Ctrl+I), or select a Language (Ctrl+K M). Start typing to dismiss or don't show this again.
```

### File main.py

```
● ● ●

1  from models.playstation import Playstation
2  from models.xbox import Xbox
3  from models.nintendo import Nintendo
4  from models.pelanggan import Pelanggan
5  from utils.laporan import Laporan
6
7  if __name__ == "__main__":
8      # Inisialisasi laporan
9      laporan = Laporan()
10
11     # Membuat objek konsol
12     ps5 = Playstation("C001", "PlayStation 5", 70000, "Digital Edition")
13     xbox = Xbox("C002", "Xbox Series X", 65000, "Asia")
14     switch = Nintendo("C003", "Nintendo Switch OLED", 50000, "Putih")
15
16     # Membuat pelanggan
17     andi = Pelanggan("P001", "Andi", "08123456789")
18     budi = Pelanggan("P002", "Budi", "08129876543")
19
20     print("== DAFTAR KONSEL ==")
21     for k in [ps5, xbox, switch]:
22         k.tampilkan_info()
23
24     print("\n== SIMULASI TRANSAKSI ==")
25     andi.sewa_console(ps5, laporan)
26     budi.sewa_console(ps5, laporan)
27     andi.kembalikan_console(laporan)
28     budi.sewa_console(ps5, laporan)
29     budi.kembalikan_console(laporan)
30
31     laporan.simpan()
```

## Output :

```
PS C:\Users\Infinix\Project UTS> & C:/Users/Infinix/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Infinix/Project UTS/main.py"
--- DAFTAR KONSEL ---
[PS] PlayStation 5 (Versi Digital Edition) - Rp70000/hari | Status: Tersedia | Dicatat: 21/10/2025 23:44
[Xbox] Xbox Series X (Region Asia) - Rp65000/hari | Status: Tersedia | Dicatat: 21/10/2025 23:44
[Nintendo] Nintendo Switch OLED - Warna: Putih | Rp50000/hari | Status: Tersedia | Dicatat: 21/10/2025 23:44

--- SIMULASI TRANSAKSI ---
[21/10/2025 23:44] Andi menyewa PlayStation 5 pada 21/10/2025 23:44
[21/10/2025 23:44] Budi gagal menyewa PlayStation 5 karena sedang disewa.
[21/10/2025 23:44] Andi mengembalikan PlayStation 5 setelah 0 jam.
[21/10/2025 23:44] Budi menyewa Playstation 5 pada 21/10/2025 23:44
[21/10/2025 23:44] Budi mengembalikan PlayStation 5 setelah 0 jam.
Laporan tersimpan di: output\laporan.txt
PS C:\Users\Infinix\Project UTS>
```

## 1. implementasi *Inheritance* (Pewarisan)

Tahap awal adalah inisialisasi objek-objek utama sistem. Kelas Playstation, Xbox, dan Nintendo adalah kelas turunan yang telah mewarisi semua sifat dan metode dasar dari kelas induk Console.

- Saat objek ps5, xbox, dan switch dibuat, mereka secara otomatis membawa atribut dasar seperti ID, Nama, Harga Sewa, dan Status dari kelas induk, berkat mekanisme *Inheritance*.
- Pewarisan ini memastikan bahwa semua objek konsol, meskipun memiliki jenis yang berbeda, dapat diperlakukan sebagai satu tipe umum (yaitu "Console") oleh fungsi atau *method* lain dalam sistem.

## 2. Implementasi *Polymorphism* (Banyak Bentuk)

Konsep *Polymorphism* didemonstrasikan melalui perulangan yang menampilkan informasi konsol:

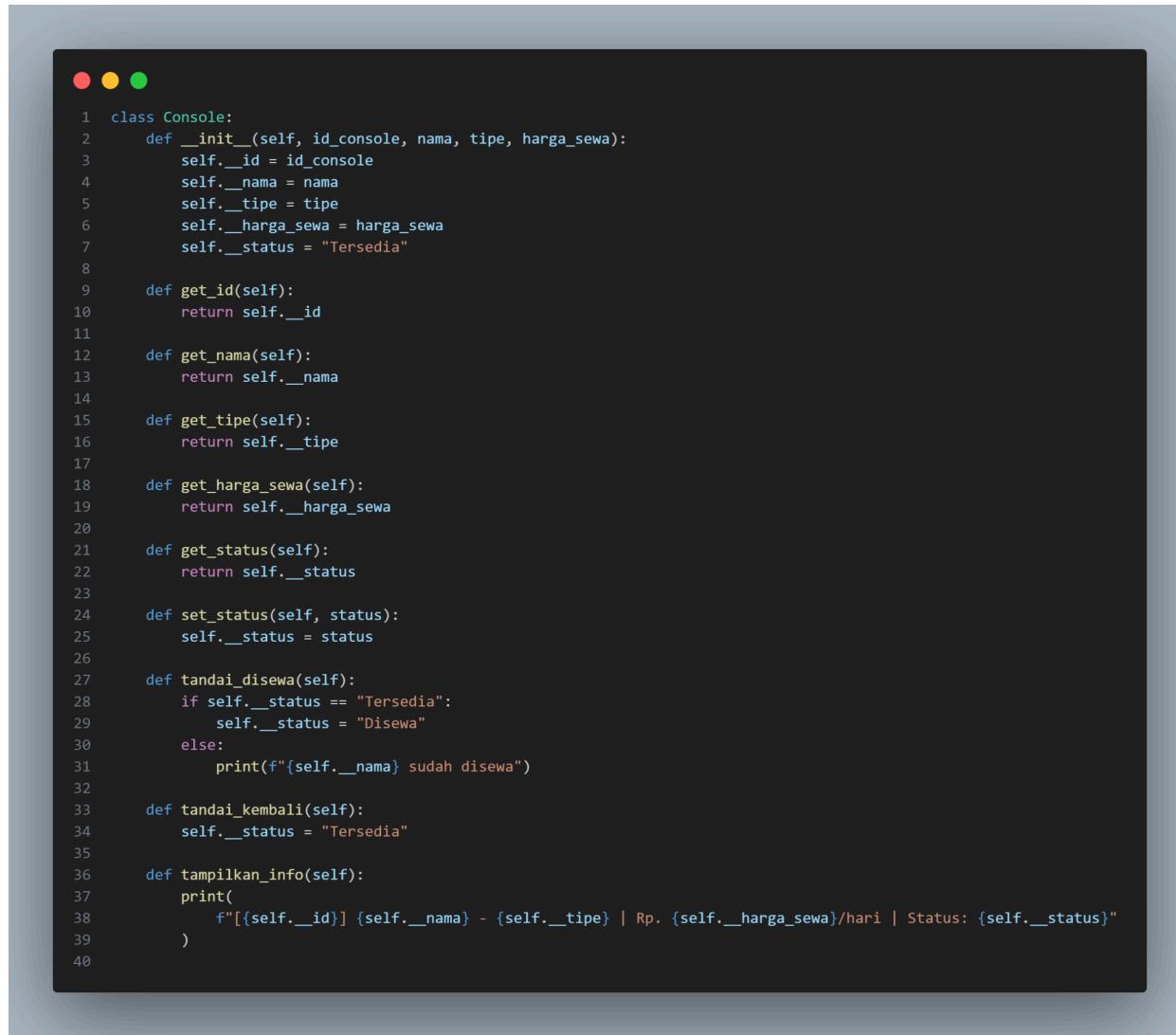
1. Panggilan Tunggal: Terdapat perulangan sederhana yang mengiterasi seluruh objek konsol (ps5, xbox, switch) dalam satu daftar.
2. Perilaku Berbeda: Pada setiap iterasi, perintah k.tampilkan\_info() dipanggil. Karena metode *tampilkan\_info()* telah ditimpa (*override*) di setiap kelas turunan, setiap objek merespons panggilan tersebut dengan perilaku yang berbeda:

- Objek ps5 menampilkan format informasi yang menyertakan detail spesifik "Versi" (khas Playstation).
- Objek xbox menampilkan format informasi yang menyertakan detail spesifik "Region" (khas Xbox).
- Objek switch menampilkan format informasi yang menyertakan detail spesifik "Warna" (khas Nintendo).

Polimorfisme ini memungkinkan sistem untuk mengelola dan menampilkan detail berbagai jenis konsol secara seragam menggunakan satu perintah, membuat kode lebih fleksibel dan mudah diperluas.

### 3. Project 3 — Modularisasi dan Integrasi Pustaka Standar

#### File console.py



```
1 class Console:
2     def __init__(self, id_console, nama, tipe, harga_sewa):
3         self.__id = id_console
4         self.__nama = nama
5         self.__tipe = tipe
6         self.__harga_sewa = harga_sewa
7         self.__status = "Tersedia"
8
9     def get_id(self):
10        return self.__id
11
12    def get_nama(self):
13        return self.__nama
14
15    def get_tipe(self):
16        return self.__tipe
17
18    def get_harga_sewa(self):
19        return self.__harga_sewa
20
21    def get_status(self):
22        return self.__status
23
24    def set_status(self, status):
25        self.__status = status
26
27    def tandai_disewa(self):
28        if self.__status == "Tersedia":
29            self.__status = "Disewa"
30        else:
31            print(f"{self.__nama} sudah disewa")
32
33    def tandai_kembali(self):
34        self.__status = "Tersedia"
35
36    def tampilkan_info(self):
37        print(
38            f"[{self.__id}] {self.__nama} - {self.__tipe} | Rp. {self.__harga_sewa}/hari | Status: {self.__status}"
39        )
```

#### File nintendo.py



```
1 from datetime import datetime
2 from models.console import Console
3
4 class Nintendo(Console):
5     def __init__(self, id_console, nama, harga_sewa, warna):
6         super().__init__(id_console, nama, "Nintendo", harga_sewa)
7         self.__warna = warna
8
9     def tampilkan_info(self):
10        waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
11        print(f"[Nintendo] {self.get_nama()} - Warna: {self.__warna} | Rp.{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}")
```

## File pelanggan.py

```
 1  from datetime import datetime
 2
 3  class Pelanggan:
 4      def __init__(self, id_pelanggan, nama, no_hp):
 5          self.__id = id_pelanggan
 6          self.__nama = nama
 7          self.__no_hp = no_hp
 8          self.__console_dipinjam = None
 9          self.__waktu_sewa = None
10
11      def get_nama(self):
12          return self.__nama
13
14      def sewa_console(self, console, laporan):
15          if console.get_status() == "Tersedia":
16              console.tandai_disewa()
17              self.__console_dipinjam = console
18              self.__waktu_sewa = datetime.now()
19              laporan.catat(f"{self.__nama} menyewa {console.get_nama()} pada {self.__waktu_sewa.strftime('%d/%m/%Y %H:%M')}")
20          else:
21              laporan.catat(f"{self.__nama} gagal menyewa {console.get_nama()} karena sedang disewa.")
22
23      def kembalikan_console(self, laporan):
24          if self.__console_dipinjam:
25              console = self.__console_dipinjam
26              console.tandai_kembali()
27              waktu_kembali = datetime.now()
28              durasi = waktu_kembali - self.__waktu_sewa
29              lama_jam = int(durasi.total_seconds() // 3600)
30              laporan.catat(f"{self.__nama} mengembalikan {console.get_nama()} setelah {lama_jam} jam.")
31              self.__console_dipinjam = None
32              self.__waktu_sewa = None
33          else:
34              laporan.catat(f"{self.__nama} tidak sedang menyewa console.")
35
```

## File playstation.py

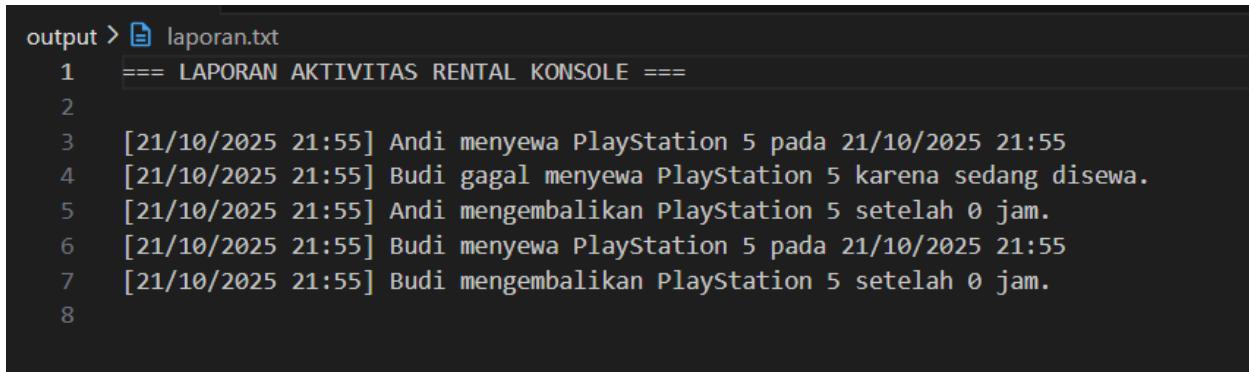
```
 1  from datetime import datetime
 2  from models.console import Console
 3
 4
 5  class Playstation(Console):
 6      def __init__(self, id_console, nama, harga_sewa, versi):
 7          super().__init__(id_console, nama, "PlayStation", harga_sewa)
 8          self.__versi = versi
 9
10      def tampilan_info(self):
11          waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
12          print(
13              f"[PS] {self.get_nama()} (Versi {self.__versi}) - Rp{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}"
14          )
15
```

## File xbox.py



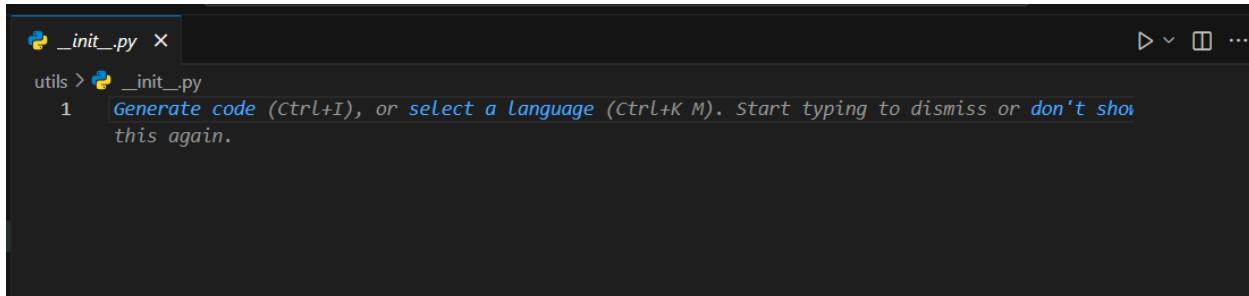
```
 1  from datetime import datetime
 2  from models.console import Console
 3
 4  class Xbox(Console):
 5      def __init__(self, id_console, nama, harga_sewa, region):
 6          super().__init__(id_console, nama, "Xbox", harga_sewa)
 7          self.__region = region
 8
 9      def tampilkan_info(self):
10          waktu = datetime.now().strftime("%d/%m/%Y %H:%M")
11          print(f"[Xbox] {self.get_nama()} (Region {self.__region}) - Rp{self.get_harga_sewa()}/hari | Status: {self.get_status()} | Dicatat: {waktu}")
```

## File laporan.txt



```
output > laporan.txt
1 === LAPORAN AKTIVITAS RENTAL KONSOLE ===
2
3 [21/10/2025 21:55] Andi menyewa PlayStation 5 pada 21/10/2025 21:55
4 [21/10/2025 21:55] Budi gagal menyewa PlayStation 5 karena sedang disewa.
5 [21/10/2025 21:55] Andi mengembalikan PlayStation 5 setelah 0 jam.
6 [21/10/2025 21:55] Budi menyewa PlayStation 5 pada 21/10/2025 21:55
7 [21/10/2025 21:55] Budi mengembalikan PlayStation 5 setelah 0 jam.
8
```

## File init.py



```
utils > __init__.py
1 Generate code (Ctrl+I), or select a language (Ctrl+K M). Start typing to dismiss or don't show this again.
```

### File main.py

```
● ● ●

1  from models.playstation import Playstation
2  from models.xbox import Xbox
3  from models.nintendo import Nintendo
4  from models.pelanggan import Pelanggan
5  from utils.laporan import Laporan
6
7  if __name__ == "__main__":
8      # Inisialisasi laporan
9      laporan = Laporan()
10
11     # Membuat objek konsol
12     ps5 = Playstation("C001", "PlayStation 5", 70000, "Digital Edition")
13     xbox = Xbox("C002", "Xbox Series X", 65000, "Asia")
14     switch = Nintendo("C003", "Nintendo Switch OLED", 50000, "Putih")
15
16     # Membuat pelanggan
17     andi = Pelanggan("P001", "Andi", "08123456789")
18     budi = Pelanggan("P002", "Budi", "08129876543")
19
20     print("== DAFTAR KONSEL ==")
21     for k in [ps5, xbox, switch]:
22         k.tampilkan_info()
23
24     print("\n== SIMULASI TRANSAKSI ==")
25     andi.sewa_console(ps5, laporan)
26     budi.sewa_console(ps5, laporan)
27     andi.kembalikan_console(laporan)
28     budi.sewa_console(ps5, laporan)
29     budi.kembalikan_console(laporan)
30
31     laporan.simpan()
```

## Output :

```
PS C:\Users\Infinix\Project UTS> & C:/Users/Infinix/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Infinix/Project UTS/main.py"
--- DAFTAR KONSEL ---
[PS] PlayStation 5 (Versi Digital Edition) - Rp70000/hari | Status: Tersedia | Dicatat: 21/10/2025 23:44
[Xbox] Xbox Series X (Region Asia) - Rp65000/hari | Status: Tersedia | Dicatat: 21/10/2025 23:44
[Nintendo] Nintendo Switch OLED - Warna: Putih | Rp50000/hari | Status: Tersedia | Dicatat: 21/10/2025 23:44

--- SIMULASI TRANSAKSI ---
[21/10/2025 23:44] Andi menyewa PlayStation 5 pada 21/10/2025 23:44
[21/10/2025 23:44] Budi gagal menyewa PlayStation 5 karena sedang disewa.
[21/10/2025 23:44] Andi mengembalikan PlayStation 5 setelah 0 jam.
[21/10/2025 23:44] Budi menyewa Playstation 5 pada 21/10/2025 23:44
[21/10/2025 23:44] Budi mengembalikan PlayStation 5 setelah 0 jam.
Laporan tersimpan di: output\laporan.txt
PS C:\Users\Infinix\Project UTS>
```

Kode utama ini berfungsi untuk menguji dan mendemonstrasikan hasil implementasi Modularisasi pada struktur proyek dan Integrasi Pustaka Standar ke dalam fungsionalitas sistem.

## 1. Implementasi Modularisasi (Struktur Kode Terorganisir)

Konsep Modularisasi diterapkan untuk memecah sistem menjadi unit-unit logis yang independen, membuatnya lebih rapi, terstruktur, dan mudah dikelola. Hal ini ditunjukkan oleh serangkaian pernyataan import di awal kode:

- Pemisahan Entitas: Kelas-kelas utama dan turunan (Playstation, Xbox, Nintendo, Pelanggan) diimpor dari *package* models (misalnya, from models.playstation import Playstation).
- Pemisahan Fungsionalitas Pendukung: Kelas Laporan yang memiliki tugas khusus (pencatatan aktivitas) diimpor dari *package* utils (from utils.laporan import Laporan).

Pemisahan ini memastikan bahwa setiap bagian kode memiliki tanggung jawab yang jelas. Misalnya, kelas-kelas di models hanya bertanggung jawab atas representasi data, sementara kelas di utils hanya bertanggung jawab atas fungsi *utility* (dukungan).

## 2. Integrasi Pustaka Standar (Pencatatan Data Efisien)

*Integrasi Pustaka Standar* membuktikan bahwa sistem memanfaatkan modul bawaan Python untuk menambah fungsionalitas tanpa harus menulis kode dari nol.

Kode ini menunjukkan bagaimana objek laporan yang diinisialisasi dari Laporan() menjadi titik sentral pemanfaatan pustaka:

- Integrasi datetime: Pustaka standar datetime (yang terintegrasi dalam kelas Laporan dan Pelanggan) digunakan untuk menangani semua aspek waktu. Dalam simulasi transaksi, datetime memastikan bahwa:
  - Waktu sewa (`self.__waktu_sewa`) dicatat secara akurat saat `andi.sewa_console()` dipanggil.
  - Setiap pesan yang dicatat melalui `laporan.catat()` memiliki stempel waktu (*timestamp*) yang presisi.
- Penyederhanaan Perhitungan Durasi: (Terjadi di metode `kembalikan_console` pada objek Pelanggan). Pustaka datetime secara otomatis memfasilitasi perhitungan durasi sewa yang kompleks ( $\text{durasi} = \text{waktu_kembali} - \text{self.__waktu_sewa}$ ), yang kemudian dikonversi menjadi jam.

## BAB IV

## PENUTUP

### A. Kesimpulan

Proyek pengembangan Sistem Manajemen Rental Perangkat Konsol ini telah berhasil diimplementasikan dengan menerapkan secara komprehensif konsep-konsep inti dari Pemrograman Berorientasi Objek (PBO) melalui tiga tahapan proyek yang saling berhubungan.

1. **Enkapsulasi dan Struktur Dasar (Project 1):** Fondasi sistem berhasil dibangun dengan mendefinisikan kelas-kelas utama (Console, Pelanggan) dan menerapkan Enkapsulasi. Penerapan atribut privat (\_) dan metode kontrol (seperti `tandai_disewa()`) memastikan bahwa integritas data sensitif, seperti status ketersediaan konsol, terlindungi dari modifikasi yang tidak terotorisasi, yang merupakan prinsip PBO fundamental untuk menciptakan kode yang aman.
2. **Inheritance dan Polymorphism (Project 2):** Sistem menunjukkan fleksibilitas melalui Pewarisan (*Inheritance*), di mana jenis konsol spesifik (Playstation, Xbox, Nintendo) mewarisi properti dasar dari kelas induk Console. Selain itu, Polimorfisme didemonstrasikan melalui metode `tampilkan_info()`, yang memungkinkan objek berbeda merespons panggilan yang sama dengan perilaku spesifik mereka masing-masing, menjadikan kode lebih modular dan mudah dikembangkan untuk jenis konsol baru.
3. **Modularisasi dan Integrasi Pustaka (Project 3):** Efisiensi pengembangan dan keterbacaan kode ditingkatkan melalui Modularisasi (pemisahan kelas ke dalam *package* models dan utils) dan Integrasi Pustaka Standar. Pemanfaatan pustaka datetime dan os ke dalam kelas Laporan dan Pelanggan membuktikan bahwa penggunaan *library* dapat mempercepat proses pemrograman, terutama untuk fungsionalitas kompleks seperti pencatatan stempel waktu dan perhitungan durasi sewa, sesuai dengan tujuan efisiensi dan pemrograman terstruktur.

Secara keseluruhan, proyek ini membuktikan bahwa penguasaan teknik PBO dan pemanfaatan pustaka standar secara terintegrasi mampu menghasilkan aplikasi manajemen rental yang terstruktur, fleksibel, efisien, dan siap dikembangkan lebih lanjut di masa mendatang.

## **DAFTAR PUSTAKA**

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
- Holovaty, A., & Kaplan-Moss, J. (2009). *The Definitive Guide to Django: Web Development Done Right*. Apress.
- Lutz, M. (2021). *Learning Python* (5th ed.). O'Reilly Media.
- McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O'Reilly Media.
- Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.