LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 10

2411102441106

Aisha Hannah Heriawan

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

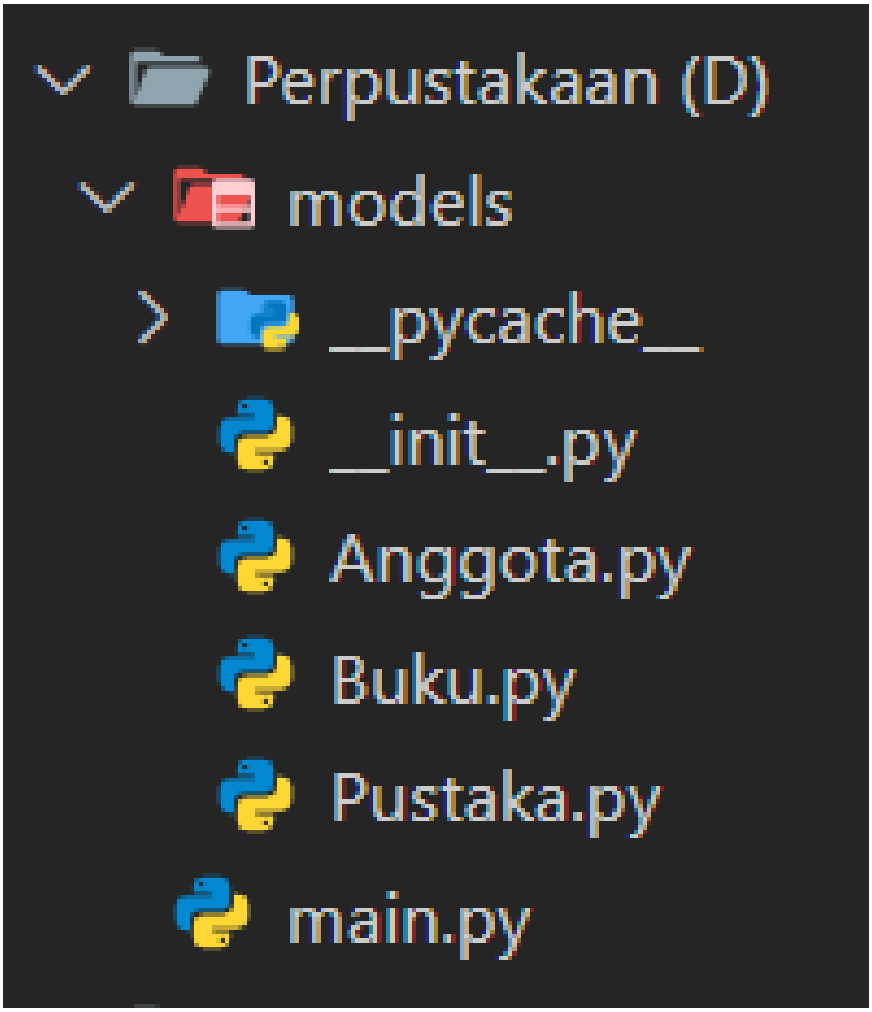
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

EMAIL : 2411102441106@umkt.ac.id

Implementasi Desain

Percobaan praktikum

Struktur lengkap folder perpustakaan



Kode lengkap “Buku.py”

```python
class Buku:
    def __init__(self, judul, penulis):
        self.judul = judul
        self.penulis = penulis
        self.dipinjam = False

    def tampilkan_info(self):
        status = "Dipinjam" if self.dipinjam else "Tersedia"
        return f"Buku: {self.judul}, Penulis: {self.penulis}
, Status: {status}"
```

Kode lengkap “Anggota.py”

```python
class Anggota:
    def __init__(self, nama):
        self.nama = nama
        self.daftar_buku = []

    def pinjam_buku(self, buku):
        if not buku.dipinjam:
            buku.dipinjam = True
            self.daftar_buku.append(buku)
            print(f"{self.nama} meminjam buku '{buku.judul}'")
        else:
            print(f"Buku '{buku.judul}' sedang dipinjam orang lain.")

    def kembalikan_buku(self, buku):
        if buku in self.daftar_buku:
            buku.dipinjam = False
            self.daftar_buku.remove(buku)
            print(f"{self.nama} mengembalikan buku '{buku.judul}'")
        else:
            print(f"{self.nama} tidak meminjam buku '{buku.judul}'")
```

EMAIL : 2411102441106@umkt.ac.id

Kode lengkap “Pustaka.py”

```python
class Pustaka:
    def __init__(self):
        self.koleksi_buku = []

    def tambah_buku(self, buku):
        self.koleksi_buku.append(buku)

    def tampilkan_koleksi(self):
        for buku in self.koleksi_buku:
            print(buku.tampilkan_info())
```

Kode lengkap “Main.py”

```python
from models.Buku import Buku
from models.Pustaka import Pustaka
from models.Anggota import Anggota

buku1 = Buku("Harry Potter", "J.K. Rowling")
buku2 = Buku("The Hobbit", "J.R.R. Tolkien")

pustaka = Pustaka()
pustaka.tambah_buku(buku1)
pustaka.tambah_buku(buku2)

anggotal = Anggota("Andi")
anggotal.pinjam_buku(buku1)
anggotal.pinjam_buku(buku2)

print("\nStatus Koleksi Pustaka:")
pustaka.tampilkan_koleksi()

anggotal.kembalikan_buku(buku1)
print("\nStatus Koleksi Setelah Pengembalian:")
pustaka.tampilkan_koleksi()
```

EMAIL : 2411102441106@umkt.ac.id

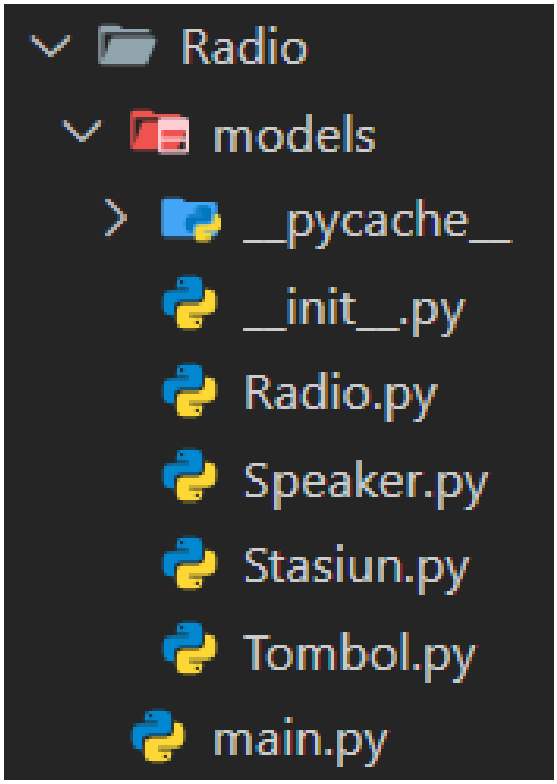
Output:

```
Status Koleksi Pustaka:
Buku: Harry Potter, Penulis: J.K. Rowling, Status: Dipinjam
Buku: The Hobbit, Penulis: J.R.R. Tolkien, Status: Dipinjam
Andi mengembalikan buku 'Harry Potter'

Status Koleksi Setelah Pengembalian:
Buku: Harry Potter, Penulis: J.K. Rowling, Status: Tersedia
Buku: The Hobbit, Penulis: J.R.R. Tolkien, Status: Dipinjam
(venv) PS C:\Users\User\OneDrive\Aish>
```

Tugas Mandiri

Struktur folder “Radio”



Kode lengkap “Radio.py”

```python
from models.Speaker import Speaker
from models.Tombol import Tombol

class Radio:
    def __init__(self, daftar_stasiun, stasiun_awal_index=0):
        self._speaker = Speaker()
        self._daftar_stasiun = daftar_stasiun
        self._indeks_stasiun_aktif = stasiun_awal_index
        self._tombol_tune = Tombol("Next")
        print("Radio Digital dihidupkan.")

    def putar_stasiun(self):
        stasiun_aktif = self._daftar_stasiun[self._indeks_stasiun_aktif]
        info = stasiun_aktif.get_info()
        self._speaker.putar_suara(info)

    def ganti_stasiun(self):
        if self._tombol_tune.ditekan():
            self._speaker.set_volume(40)
            self._indeks_stasiun_aktif = (self._indeks_stasiun_aktif + 1) % len(self._daftar_stasiun)
            print(f"Radio sedang tuning... ninung ninung eaa...")
            self.putar_stasiun()
```

EMAIL : 2411102441106@umkt.ac.id

Kode lengkap “Speaker.py”

```python
class Speaker:
    def __init__(self):
        self._volume = 50
        print("Speaker Radio disiapkan.")

    def set_volume(self, level):
        self._volume = max(0, min(100, level))
        print(f"Volume diatur ke: {self._volume}")

    # mengubah status Speaker (berdasarkan input Radio)
    def putar_suara(self, info_stasiun):
        if self._volume > 0:
            print(f"Speaker memutar: {info_stasiun} [Vol: {self._volume}]")
        else:
            print("Speaker senyap. (Volume 0)")

# komposisi (tidak akan berfungsi sebagai speaker tanpa radio)
```

Kode lengkap “Stasiun.py”

```python
class Stasiun:
    def __init__(self, nama, frekuensi):
        self._nama = nama
        self._frekuensi = frekuensi

    def get_info(self):
        return f"Stasiun: {self._nama} ({self._frekuensi} MHz)"
# mengembalikan informasi

# agregasi (bisa berdiri sendiri tanpa radio)
```

EMAIL : 2411102441106@umkt.ac.id

Kode lengkap “Tombol.py”

```python
class Tombol:
    def __init__(self, fungsi):
        self._fungsi = fungsi

    def ditekan(self):
        print(f"Tombol '{self._fungsi}' **Ditekan**.")
        return True

# asosisasi (alat bantu, dapat diganti/digunakan dengan yg lain)
```

Kode lengkap “Main.py”

```python
from models.Radio import Radio
from models.Stasiun import Stasiun

stasiun_a = Stasiun("Radio Edukasi", 107.0)
stasiun_b = Stasiun("Radio Musik Pop", 99.8)
stasiun_c = Stasiun("Radio Berita", 88.0)
daftar_stasiun = [stasiun_a, stasiun_b, stasiun_c]

radio_saya = Radio(daftar_stasiun)

print("\n--- Putaran Awal ---")

# memutar stasiun pertama
radio_saya.putar_stasiun()

print("\n--- Ganti Stasiun ---")

# ganti stasiun
radio_saya.ganti_stasiun()

# ganti stasiun
radio_saya.ganti_stasiun()
```

EMAIL : 2411102441106@umkt.ac.id

Output:

```
Speaker Radio disiapkan.
Radio Digital dihidupkan.

--- Putaran Awal ---
Speaker memutar: Stasiun: Radio Edukasi (107.0 MHz) [Vol: 50]

--- Ganti Stasiun ---
Tombol 'Next' **Ditekan**.
Volume diatur ke: 40
Radio sedang tuning... ninung ninung eaa...
Speaker memutar: Stasiun: Radio Musik Pop (99.8 MHz) [Vol: 40]
Tombol 'Next' **Ditekan**.
Volume diatur ke: 40
Radio sedang tuning... ninung ninung eaa...
Speaker memutar: Stasiun: Radio Berita (88.0 MHz) [Vol: 40]
(venv) PS C:\Users\User\OneDrive\Aish>
```

Interaksi:

1. Objek A memanggil method pbjek B

   Saat tombol ditekan, objek utama **Radio** (A) memutuskan untuk mengatur ulang volume dan memanggil *method* set_volume() pada objek **Speaker** (B). Ini adalah panggilan *method* langsung dari A ke B. Terlihat pada output bagian “volume diatur ke: 40”

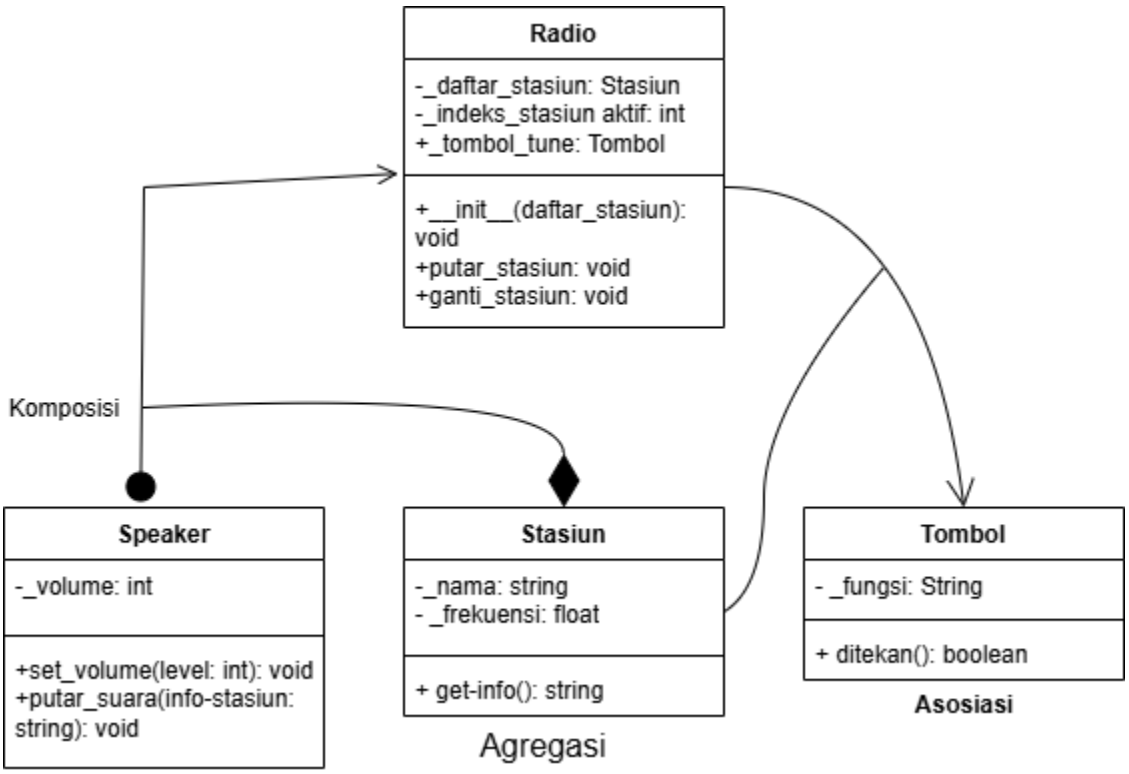
2. Objek B mengubah status objek C

   Terlihat pada output bagian “Speaker memutar: stasiun: radio musik pop (99.8 MHz) [Vol: 40]. Ketika Radio (B) memanggil putar_suara() pada Speaker (C), ini secara esensi mengubah status Speaker (C) dari diam menjadi bersuara (atau mengubah output suaranya)

3. Objek C mengembalikan nilai ke objek A

   Objek Radio (A) perlu tahu informasi apa yang akan diputar. Ia mengambil objek Stasiun (C) yang baru dipilih, dan memanggil *method* get_info() pada objek tersebut. Stasiun (C) kemudian mengembalikan nilai berupa string informasi stasiun kepada Radio (A). Terlihat pada output bagian “Stasiun: Radio musik pop (99.9 MHz”

Desain UML



EMAIL : 2411102441106@umkt.ac.id