Nama: Rifki Fadilah

Kelas: R1

NIM: 210511011

PBO2 Tugas2

Program perpustakaan Menerapkan Inheritance Program

Perintah Database:

```
import mysql.connector as mc

class DBConnection:

    def __init__(self):
        self.host = DB_HOST = "localhost"
        self.port = DB_PORT = 3306
        self.name = DB_NAME = "kampusku"
        self.user = DB_USER = "root"
        self.password = DB_PASSWORD = ""
        self.conn = None
        self.cursor = None
        self.result = None
        self.connected = False
        self.affected = 0
        self.connect()

    @property
    def connection_status(self):
        return self.connected

    def connect(self):
        try:
            self.conn = mc.connect(host = self.host,
                                    port = self.port,
                                    database = self.name,
                                    user = self.user,
                                    password = self.password)
            self.connected = True
            self.cursor=self.conn.cursor()
        except mc.Error as e:
            self.connected = False
        return self.conn

    def disconnect(self):
        if(self.connected==True):
            self.conn.close
        else:
            self.conn = None

    def findOne(self, sql):
        self.connect()
        self.cursor.execute(sql)
        self.result = self.cursor.fetchone()
        #a = self.cursor.rowcount
        #if(a>0):
        #   self.result = res
        #else:
        #   self.result = None
        return self.result

    def findAll(self, sql):
        self.connect()
        self.result = self.cursor.execute(sql)
        self.result = self.cursor.fetchall()
        return self.result

    def insert(self, sql):
        self.connect()
        self.cursor.execute(sql)
        self.conn.commit()
        self.affected = self.cursor.rowcount
        return self.affected

    def update(self, sql, val):
        self.connect()
        self.cursor.execute(sql, val)
        self.conn.commit()
        self.affected = self.cursor.rowcount
        return self.affected

    def delete(self, sql):
        self.connect()
        self.cursor.execute(sql)
        self.conn.commit()
        self.affected = self.cursor.rowcount
        return self.affected

    def show(self, sql):
        self.connect()
        self.cursor.execute(sql)
        self.result = self.cursor.fetchone()
        return self.result

    @property
    def info(self):
        if(self.connected==True):
            return "Server is running on " + self.host + ' using port ' +
str(self.port)
        else:
            return "Server is offline."

mydb = DBConnection()
c = mydb.info
print(c)
```

Tabel Database 1:

```
-- phpMyAdmin SQL Dump
-- version 5.0.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Dec 26, 2022 at 02:38 AM
-- Server version: 10.4.17-MariaDB
-- PHP Version: 7.4.13

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
```

```
-- Database: `kampusbaru`
--

-- --------------------------------------------------------

--
-- Table structure for table `perpustakaan`
--

CREATE TABLE `perpustakaan` (
  `idprsp` int(15) NOT NULL,
  `ida` varchar(10) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `jk` varchar(10) NOT NULL,
  `alamat` varchar(20) NOT NULL,
  `buku` varchar(50) NOT NULL,
  `tahunterbit` varchar(10) NOT NULL,
  `kategori` varchar(50) NOT NULL,
  `penulis` varchar(50) NOT NULL,
  `penerbit` varchar(50) NOT NULL,
  `peminjaman` varchar(50) NOT NULL,
  `pengembalian` varchar(20) NOT NULL,
  `telat` varchar(20) NOT NULL,
  `denda` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `karyawan`
--

INSERT INTO `perpustakaan` (`idprsp`, `ida`, `nama`, `jk`, `alamat`, `buku`,
`tahunterbit`, `kategori`, `penulis`, `penerbit`, `peminjaman`, `pengembalian`,
`telat`, `denda`) VALUES

(1, '210511011', 'Rifki Fadilah', 'Laki-Laki', 'Palimanan', 'Cara Ngoding',
'2009', 'Ensiklopedia',  'Jono', 'Yanto', '1/1/2023', '9/1/2023','Telat', 'Rp
2.000'),
(2, '210511031', 'M.Hilman Humaini', 'Laki-Laki', 'Majalengka', 'Kamus Bahasa
Jepang', '2012', 'Kamus', 'Jono', 'Yanto' ,'1/1/2023', '7/1/2023','Tidak Telat',
'Rp 0'),
(3, '210511029', 'Tegar Trisakti P.', 'Laki-Laki', 'Mundu', 'Naruto X Boruto',
'2020', 'Komik', 'Jono', 'Yanto' ,'1/1/2023', '10/1/2023','Telat', 'Rp 4.000');


--
```

```
-- Indexes for dumped tables
--

--
-- Indexes for table `karyawan`
--
ALTER TABLE `perpustakaan`
  ADD PRIMARY KEY (`idprsp`),
  ADD UNIQUE KEY `ida` (`ida`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `karyawan`
--
ALTER TABLE `perpustakaan`
  MODIFY `idprsp` int(15) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Tabel Database 2:

```
-- phpMyAdmin SQL Dump
-- version 5.0.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Dec 26, 2022 at 02:38 AM
-- Server version: 10.4.17-MariaDB
-- PHP Version: 7.4.13

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `kampusku`
--

-- --------------------------------------------------------

--
-- Table structure for table `buku`
--

CREATE TABLE `buku` (
  `idbk` int(11) NOT NULL,
  `kodebuku` varchar(10) NOT NULL,
  `judul` varchar(50) NOT NULL,
  `penulis` varchar(20) NOT NULL,
  `penerbit` varchar(20) NOT NULL,
  `tahun` char(5) NOT NULL,
  `kategori` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `buku`
--

INSERT INTO `buku` (`idbk`, `kodebuku`, `judul`, `penulis`, `penerbit`,
`tahun`,`kategori`) VALUES
(1, '202301', 'Memasak Mudah', 'Sugiono', 'Gramedia', '2010','Ensiklopedia'),
(2, '202302', 'Memasak Mudah', 'Sugiono', 'Gramedia', '2010','Ensiklopedia'),
(3, '202303', 'Memasak Mudah', 'Sugiono', 'Gramedia', '2010','Ensiklopedia');


--
-- Indexes for dumped tables
--

--
-- Indexes for table `buku`
--
ALTER TABLE `buku`
  ADD PRIMARY KEY (`idbk`),
  ADD UNIQUE KEY `kodebuku` (`kodebuku`);

--
-- AUTO_INCREMENT for dumped tables
```

```
--

--
-- AUTO_INCREMENT for table `buku`
--
ALTER TABLE `buku`
  MODIFY `idbk` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Fungsi/Perintah 1:

```
from db import DBConnection as mydb


class Perpustakaan:

    def __init__(self):
        self.__idprsp=None

        self.__ida=None
        self.__nama=None
        self.__jk=None
        self.__alamat=None
        self.__buku=None
        self.__tahunterbit=None
        self.__kategori=None
        self.__penulis=None
        self.__penerbit=None
        self.__peminjaman=None
        self.__pengembalian=None
        self.__telat=None
        self.__denda=None

        self.__info = None
        self.conn = None
        self.affected = None
        self.result = None
    @property
    def info(self):
        if(self.__info==None):
            return "ID Anggota:" + self.__ida + "\n" + "Nama:" + self.__nama +
"\n" + "JK" + self.__jk +"\n" + "Alamat:" + self.__alamat + "\n" + "Judul Buku" +
self.__buku + "Tahun Terbit:" + self.__tahunterbit + "\n" + "\n" + "Kategori:" +
self.__kategori + "\n" + "Penulis:" + self.__penulis + "\n" + "Penerbit:" +
self.__penerbit + "\n" + "Peminjaman:" + self.__peminjaman + "\n" +
"Pengembalian:" + self.__pengembalian + "\n" + "Telat:" + self.__telat + "\n" +
"denda:" + self.__denda
        else:
            return self.__info

    @info.setter
    def info(self, value):
        self.__info = value

    @property
    def idprsp(self):
        return self.__idprsp


    @property
    def ida(self):
        return self.__ida
    @ida.setter
    def ida(self, value):
        self.__ida = value


    @property
    def nama(self):
        return self.__nama
    @nama.setter
    def nama(self, value):
        self.__nama = value


    @property
    def jk(self):
        return self.__jk
    @jk.setter
    def jk(self, value):
        self.__jk = value
    @property
    def alamat(self):
        return self.__alamat
    @alamat.setter
    def alamat(self, value):
        self.__alamat = value


    @property
    def buku(self):
        return self.__buku
    @buku.setter
    def buku(self, value):
        self.__buku = value


    @property
    def tahunterbit(self):
        return self.__tahunterbit
    @tahunterbit.setter
    def tahunterbit(self, value):
        self.__tahunterbit = value


    @property
    def kategori(self):
        return self.__kategori
    @kategori.setter
    def kategori(self, value):
        self.__kategori = value


    @property
    def penulis(self):
        return self.__penulis
    @penulis.setter
    def penulis(self, value):
        self.__penulis = value


    @property
    def penerbit(self):
        return self.__penerbit
    @penerbit.setter
    def penerbit(self, value):
        self.__penerbit = value
    @property
    def peminjaman(self):
        return self.__peminjaman
    @peminjaman.setter
    def peminjaman(self, value):
        self.__peminjaman = value


    @property
    def pengembalian(self):
        return self.__pengembalian
    @pengembalian.setter
    def pengembalian(self, value):
        self.__pengembalian = value


    @property
    def telat(self):
        return self.__telat
    @telat.setter
    def telat(self, value):
        self.__telat = value


    @property
    def denda(self):
        return self.__denda
    @denda.setter
    def denda(self, value):
        self.__denda = value



    def simpan(self):
        self.conn = mydb()
        val = (self.__ida, self.__nama, self.__jk, self.__alamat, self.__buku,
self.__tahunterbit, self.__kategori, self.__peminjaman, self.__penulis,
self.__penerbit, self.__pengembalian, self.__telat, self.__denda)
        sql="INSERT INTO perpustakaan (ida, nama, jk, alamat, buku, tahunterbit,
kategori, peminjaman, penulis, penerbit, pengembalian, telat, denda) VALUES " +
str(val)
        self.affected = self.conn.insert(sql)
        self.conn.disconnect
        return self.affected
    def update(self, id):
        self.conn = mydb()
        val = (self.__ida, self.__nama, self.__jk, self.__alamat, self.__buku,
self.__tahunterbit, self.__kategori, self.__peminjaman, self.__penulis,
self.__penerbit, self.__pengembalian, self.__telat, self.__denda, id)
        sql="UPDATE perpustakaan SET ida=%s, nama=%s, jk=%s, alamat=%s, buku=%s,
tahunterbit=%s, kategori=%s, peminjaman=%s, penulis=%s, penerbit=%s,
pengembalian=%s, telat=%s, denda=%s WHERE idprsp=%s"
        self.affected = self.conn.update(sql, val)
        self.conn.disconnect
        return self.affected

    def updateByNIM(self, ida):
        self.conn = mydb()
        val = (self.__nama, self.__jk, self.__alamat, self.__buku,
self.__tahunterbit, self.__kategori, self.__peminjaman, self.__penulis,
self.__penerbit, self.__pengembalian, self.__telat, self.__denda, ida)
        sql="UPDATE perpustakaan SET nama=%s, jk=%s, alamat=%s, buku=%s,
tahunterbit=%s, kategori=%s, peminjaman=%s, penulis=%s, penerbit=%s,
pengembalian=%s, telat=%s, denda=%s WHERE ida=%s"
        self.affected = self.conn.update(sql, val)
        self.conn.disconnect
        return self.affected

    def delete(self, id):
        self.conn = mydb()
        sql="DELETE FROM perpustakaan WHERE idprsp='" + str(id) + "'"
        self.affected = self.conn.delete(sql)
        self.conn.disconnect
        return self.affected

    def deleteByNIM(self, ida):
        self.conn = mydb()
        sql="DELETE FROM perpustakaan WHERE ida='" + str(ida) + "'"
        self.affected = self.conn.delete(sql)
        self.conn.disconnect
        return self.affected

    def getByID(self, id):
        self.conn = mydb()
        sql="SELECT * FROM perpustakaan WHERE idprsp='" + str(id) + "'"
        self.__ida = self.result[1]
        self.__nama = self.result[2]
        self.__jk = self.result[3]
        self.__alamat = self.result[4]
        self.__buku = self.result[5]
        self.__tahunterbit = self.result[6]
        self.__kategori = self.result[7]
        self.__penulis = self.result[8]
        self.__penerbit = self.result[9]
        self.__peminjaman = self.result[10]
        self.__pengembalian = self.result[11]
        self.__telat = self.result[12]
        self.__denda = self.result[13]
        self.conn.disconnect
        return self.result

    def getByNIM(self, ida):
        a=str(ida)
        b=a.strip()
        self.conn = mydb()
        sql="SELECT * FROM perpustakaan WHERE ida='" + b + "'"
        self.result = self.conn.findOne(sql)
        if(self.result!=None):
            self.__ida = self.result[1]
            self.__nama = self.result[2]
            self.__jk = self.result[3]
            self.__alamat = self.result[4]
            self.__buku = self.result[5]
            self.__tahunterbit = self.result[6]
            self.__kategori = self.result[7]
            self.__penulis = self.result[8]
            self.__penerbit = self.result[9]
            self.__peminjaman = self.result[10]
            self.__pengembalian = self.result[11]
            self.__telat = self.result[12]
            self.__denda = self.result[13]
            self.affected = self.conn.cursor.rowcount
        else:
            self.__ida = ''
            self.__nama = ''
            self.__jk = ''
            self.__alamat = ''
            self.__buku = ''
            self.__tahunterbit = ''
            self.__kategori = ''
            self.__penulis = ''
            self.__penerbit = ''
            self.__peminjaman = ''
            self.__pengembalian = ''
            self.__telat = ''
            self.__denda = ''
            self.affected = 0
        self.conn.disconnect
        return self.result

    def getAllData(self):
        self.conn = mydb()
        sql="SELECT * FROM perpustakaan"
        self.result = self.conn.findAll(sql)
        return self.result

a = Perpustakaan()
b = a.getAllData()
print(b)
```

Fungsi/Perintah 2:

```
from db import DBConnection as mydb


class Buku:

    def __init__(self):
        self.__idbk=None

        self.__kodebuku=None
        self.__judul=None
        self.__penulis=None
        self.__penerbit=None
        self.__tahun=None
        self.__kategori=None

        self.__info = None
        self.conn = None
        self.affected = None
        self.result = None


    @property
    def info(self):
        if(self.__info==None):
            return "Kode Buku:" + self.__kodebuku + "\n" + "Judul:" +
self.__judul + "\n" + "Penulis" + self.__penulis + "\n" + "Penerbit:" +
self.__penerbit + "\n" + "Tahun:" + self.__tahun + "\n" + "Kategori:" +
self.__kategori
        else:
            return self.__info

    @info.setter
    def info(self, value):
        self.__info = value

    @property
    def idbk(self):
        return self.__idbk

    @property
    def kodebuku(self):
        return self.__kodebuku

    @kodebuku.setter
    def kodebuku(self, value):
        self.__kodebuku = value

    @property
    def judul(self):
        return self.__judul

    @judul.setter
    def judul(self, value):
        self.__judul = value

    @property
    def penulis(self):
        return self.__penulis

    @penulis.setter
    def penulis(self, value):
        self.__penulis = value

    @property
    def penerbit(self):
        return self.__penerbit

    @penerbit.setter
    def penerbit(self, value):
        self.__penerbit = value

    @property
    def tahun(self):
        return self.__tahun

    @tahun.setter
    def tahun(self, value):
        self.__tahun = value

    @property
    def kategori(self):
        return self.__kategori

    @kategori.setter
    def kategori(self, value):
        self.__kategori = value

    def simpan(self):
        self.conn = mydb()
        val = (self.__kodebuku, self.__judul, self.__penulis, self.__penerbit,
self.__tahun, self.__kategori)
        sql="INSERT INTO buku (kodebuku, judul, penulis, penerbit, tahun,
kategori) VALUES " + str(val)
        self.affected = self.conn.insert(sql)
        self.conn.disconnect
        return self.affected

    def update(self, id):
        self.conn = mydb()
        val = (self.__kodebuku, self.__judul, self.__penulis, self.__penerbit,
self.__tahun, self.__kategori, id)
        sql="UPDATE buku SET kodebuku = %s, judul = %s, penulis=%s, penerbit=%s,
tahun=%s, kategori=%s WHERE idbk=%s"
        self.affected = self.conn.update(sql, val)
        self.conn.disconnect
        return self.affected

    def updateByNIM(self, kodebuku):
        self.conn = mydb()
        val = (self.__judul, self.__penulis, self.__penerbit, self.__tahun,
self.__kategori, kodebuku)
        sql="UPDATE buku SET judul = %s, penulis=%s, penerbit=%s, tahun=%s,
kategori=%s WHERE kodebuku=%s"
        self.affected = self.conn.update(sql, val)
        self.conn.disconnect
        return self.affected

    def delete(self, id):
        self.conn = mydb()
        sql="DELETE FROM buku WHERE idbk='" + str(id) + "'"
        self.affected = self.conn.delete(sql)
        self.conn.disconnect
        return self.affected

    def deleteByNIM(self, kodebuku):
        self.conn = mydb()
        sql="DELETE FROM buku WHERE kodebuku='" + str(kodebuku) + "'"
        self.affected = self.conn.delete(sql)
        self.conn.disconnect
        return self.affected

    def getByID(self, id):
        self.conn = mydb()
        sql="SELECT * FROM buku WHERE idbk='" + str(id) + "'"
        self.result = self.conn.findOne(sql)
        self.__kodebuku = self.result[1]
        self.__judul = self.result[2]
        self.__penulis = self.result[3]
        self.__penerbit = self.result[4]
        self.__tahun = self.result[5]
        self.__kategori = self.result[6]
        self.conn.disconnect
        return self.result

    def getByNIM(self, kodebuku):
        a=str(kodebuku)
        b=a.strip()
        self.conn = mydb()
        sql="SELECT * FROM buku WHERE kodebuku='" + b + "'"
        self.result = self.conn.findOne(sql)
        if(self.result!=None):
            self.__kodebuku = self.result[1]
            self.__judul = self.result[2]
            self.__penulis = self.result[3]
            self.__penerbit = self.result[4]
            self.__tahun = self.result[5]
            self.__kategori = self.result[6]
            self.affected = self.conn.cursor.rowcount
        else:
            self.__kodebuku = ''
            self.__judul = ''
            self.__penulis = ''
            self.__penerbit = ''
            self.__tahun = ''
            self.__kategori = ''
            self.affected = 0
        self.conn.disconnect
        return self.result

    def getAllData(self):
        self.conn = mydb()
        sql="SELECT * FROM buku"
        self.result = self.conn.findAll(sql)
        return self.result

a = Buku()
b = a.getAllData()
print(b)
```

Tampilan 1:

```
import tkinter as tk
from tkinter import Frame,Label,Entry,Button,Radiobutton,ttk,VERTICAL,YES,BOTH,END,Tk,W,StringVar,messagebox
from Perpustakaan import Perpustakaan

import os


os.system("cls")

class FrmPerpustakaan:

    def __init__(self, parent, title):
        self.parent = parent
        self.parent.geometry("870x470")
        self.parent.title(title)
        self.parent.protocol("WM_DELETE_WINDOW", self.onKeluar)
        self.ditemukan = None
        self.aturKomponen()
        self.onReload()
    def aturKomponen(self):
        mainFrame = Frame(self.parent, bd=10)
        mainFrame.pack(fill=BOTH, expand=YES)

        # Label
        Label(mainFrame, text='                         💖Daftar Pengunjung
Perpustakaan UMC    💖 ').grid(row=15, column=2,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='   ⭐  Data Anggota   ⭐').grid(row=0, column=4,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='ID Anggota       :').grid(row=1, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Nama                :').grid(row=2, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='JK                       :').grid(row=4, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Alamat              :').grid(row=3, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Judul Buku       :').grid(row=2, column=2,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Tahun Terbit     :').grid(row=3, column=2,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Kategori Buku  :').grid(row=4, column=2,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Penulis              :').grid(row=5, column=2,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Penerbit            :').grid(row=6, column=2,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Peminjaman     :').grid(row=5, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Pengembalian  :').grid(row=6, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Status                :').grid(row=7, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Denda                :').grid(row=7, column=2,
            sticky=W, padx=5, pady=5)


        # Textbox
        self.txtIDA = Entry(mainFrame)
        self.txtIDA.grid(row=1, column=1, padx=5, pady=5)
        self.txtIDA.bind("<Return>",self.onCari) # menambahkan event Enter key

        self.txtNama = Entry(mainFrame)
        self.txtNama.grid(row=2, column=1, padx=5, pady=5)
        self.txtAlamat = Entry(mainFrame)
        self.txtAlamat.grid(row=3, column=1, padx=5, pady=5)

        self.txtJudul = Entry(mainFrame)
        self.txtJudul.grid(row=2, column=2, padx=5, pady=5)

        self.txtPenulis = Entry(mainFrame)
        self.txtPenulis.grid(row=5, column=2, padx=5, pady=5)

        self.txtPenerbit = Entry(mainFrame)
        self.txtPenerbit.grid(row=6, column=2, padx=5, pady=5)

        self.txtTahun = Entry(mainFrame)
        self.txtTahun.grid(row=3, column=2, padx=5, pady=5)

        self.txtPeminjaman = Entry(mainFrame)
        self.txtPeminjaman.grid(row=5, column=1, padx=5, pady=5)

        self.txtPengembalian = Entry(mainFrame)
        self.txtPengembalian.grid(row=6, column=1, padx=5, pady=5)

        # Radio Button
        self.txtTelat = StringVar()
        self.Y = Radiobutton(mainFrame, text='Telat', value='Telat',
variable=self.txtTelat)
        self.Y.grid(row=7, column=1, padx=5, pady=5, sticky=W)
        self.Y.select() # set pilihan yg pertama
        self.T = Radiobutton(mainFrame, text='Tidak Telat', value='Tidak Telat',
variable=self.txtTelat)
        self.T.grid(row=8, column=1, padx=5, pady=5, sticky=W)


        # Combo Box
        self.txtKategori = StringVar()
        Cbo = ttk.Combobox(mainFrame, width = 16, textvariable =
self.txtKategori)
        Cbo.grid(row=4, column=2, padx=5, pady=5)
        # Adding combobox drop down list
        Cbo['values'] = ('Novel','Majalah', 'Kamus','Komik','Manga',
'Ensiklopedia', 'Biografi', 'Naskah', 'Light Novel')
        Cbo.current()

        self.txtJK = StringVar()
        Cbo = ttk.Combobox(mainFrame, width = 16, textvariable = self.txtJK)
        Cbo.grid(row=4, column=1, padx=5, pady=5)
        # Adding combobox drop down list
        Cbo['values'] = ('Laki-Laki','Perempuan')
        Cbo.current()

        self.txtDenda = StringVar()
        Cbo = ttk.Combobox(mainFrame, width = 16, textvariable = self.txtDenda)
        Cbo.grid(row=7, column=2, padx=5, pady=5)
        # Adding combobox drop down list
        Cbo['values'] = ('Rp 0','Rp 2.000', 'Rp 4.000','Rp 6.000','Rp 8.000', 'Rp 
10.000')
        Cbo.current()

        # Button
        self.btnSimpan = Button(mainFrame, text='Save', command=self.onSimpan, 
width=10,  fg= "white", bg="blue")
        self.btnSimpan.grid(row=9, column=1, padx=5, pady=5)
        self.btnClear = Button(mainFrame, text='Clear', command=self.onClear, 
width=10,  fg= "black", bg="yellow")
        self.btnClear.grid(row=9, column=2, padx=5, pady=5)
        self.btnHapus = Button(mainFrame, text='Delete', command=self.onDelete, 
width=10,  fg= "white", bg="red")
        self.btnHapus.grid(row=9, column=3, padx=5, pady=5)
        self.btnCari = Button(mainFrame, text='Search ID', command=self.onCari, 
width=10,  fg= "white", bg="green")
        self.btnCari.grid(row=1, column=2, padx=5, pady=5)

        # define columns
        columns = ('idprsp', 'ida', 'nama', 'jk', 'alamat', 'buku', 
'tahunterbit','kategori', 'penulis', 'penerbit', 'peminjaman', 'pengembalian', 
'telat', 'denda')

        self.tree = ttk.Treeview(mainFrame, columns=columns, show='headings')
        # define headings
        self.tree.heading('idprsp', text='No')
        self.tree.column('idprsp', width="25")
        self.tree.heading('ida', text='ID Anggota')
        self.tree.column('ida', width="80")
        self.tree.heading('nama', text='Nama')
        self.tree.column('nama', width="150")
        self.tree.column('nama', width="150")
        self.tree.heading('jk', text='Jk')
        self.tree.column('jk', width="80")
        self.tree.heading('alamat', text='Alamat')
        self.tree.column('alamat', width="100")
        self.tree.heading('buku', text='Judul Buku')
        self.tree.column('buku', width="200")
        self.tree.heading('tahunterbit', text='Tahun')
        self.tree.column('tahunterbit', width="50")
        self.tree.heading('kategori', text='Kategori')
        self.tree.column('kategori', width="85")
        self.tree.heading('penulis', text='Penulis')
        self.tree.column('penulis', width="85")
        self.tree.heading('penerbit', text='Penerbit')
        self.tree.column('penerbit', width="85")
        self.tree.heading('peminjaman', text='Peminjaman')
        self.tree.column('peminjaman', width="85")
        self.tree.heading('pengembalian', text='Pengembalian')
        self.tree.column('pengembalian', width="85")
        self.tree.heading('telat', text='Status')
        self.tree.column('telat', width="65")
        self.tree.heading('denda', text='Denda')
        self.tree.column('denda', width="75")
        # set tree position
        self.tree.place(x=0, y=355)
        self.onReload()

        columns = ('idprsp', 'ida', 'nama', 'jk', 'alamat')

        self.tree = ttk.Treeview(mainFrame, columns=columns, show='headings')
        # define headings
        self.tree.heading('idprsp', text='No')
        self.tree.column('idprsp', width="25")
        self.tree.heading('ida', text='ID Anggota')
        self.tree.column('ida', width="80")
        self.tree.heading('nama', text='Nama')
        self.tree.column('nama', width="150")
        self.tree.heading('jk', text='Jk')
        self.tree.column('jk', width="80")
        self.tree.heading('alamat', text='Alamat')
        self.tree.column('alamat', width="100")
        # set tree position
        self.tree.place(x=525, y=30)
        self.onReload()


    def onClear(self, event=None):
        self.txtIDA.delete(0,END)
        self.txtIDA.insert(END,"")
        self.txtNama.delete(0,END)
        self.txtNama.insert(END,"")
        self.txtJK.set("")
        self.txtAlamat.delete(0,END)
        self.txtAlamat.insert(END,"")
        self.txtJudul.delete(0,END)
        self.txtJudul.insert(END,"")
        self.txtTahun.delete(0,END)
        self.txtTahun.insert(END,"")
        self.txtKategori.set("")
        self.txtPenulis.delete(0,END)
        self.txtPenulis.insert(END,"")
        self.txtPenerbit.delete(0,END)
        self.txtPenerbit.insert(END,"")
        self.txtPeminjaman.delete(0,END)
        self.txtPeminjaman.insert(END,"")
        self.txtPengembalian.delete(0,END)
        self.txtPengembalian.insert(END,"")
        self.txtDenda.set("")

        self.btnSimpan.config(text="Simpan")
        self.Y.select()
        self.onReload()
        self.ditemukan = False

    def onReload(self, event=None):
        # get data perpustakaan
        prps = Perpustakaan()
        result = prps.getAllData()
        for item in self.tree.get_children():
            self.tree.delete(item)
        students=[]
        for row_data in result:
            students.append(row_data)

        for student in students:
            self.tree.insert('',END, values=student)

    def onCari(self, event=None):
        ida = self.txtIDA.get()
        prps = Perpustakaan()
        res = prps.getByNIM(ida)
        rec = prps.affected
        if(rec>0):
            messagebox.showinfo("showinfo", "Data Ditemukan")
            self.TampilkanData()
            self.ditemukan = True
        else:
            messagebox.showwarning("showwarning", "Data Tidak Ditemukan")
            self.ditemukan = False
            self.txtNama.focus()
        return res

    def TampilkanData(self, event=None):
        ida = self.txtIDA.get()
        prps = Perpustakaan()
        res = prps.getByNIM(ida)
        self.txtNama.delete(0,END)
        self.txtNama.insert(END,prps.nama)
        self.txtAlamat.delete(0,END)
        self.txtAlamat.insert(END,prps.alamat)
        self.txtJudul.delete(0,END)
        self.txtJudul.insert(END,prps.buku)
        self.txtTahun.delete(0,END)
        self.txtTahun.insert(END,prps.tahunterbit)
        self.txtPenulis.delete(0,END)
        self.txtPenulis.insert(END,prps.penulis)
        self.txtPenerbit.delete(0,END)
        self.txtPenerbit.insert(END,prps.penerbit)
        self.txtPeminjaman.delete(0,END)
        self.txtPeminjaman.insert(END,prps.peminjaman)
        self.txtPengembalian.delete(0,END)
        self.txtPengembalian.insert(END,prps.pengembalian)
        telat = prps.telat
        if(telat=="Tidak"):
            self.T.select()
        else:
            self.Y.select()
        self.txtKategori.set(prps.kategori)
        self.txtJK.set(prps.jk)
        self.txtDenda.set(prps.denda)
        self.btnSimpan.config(text="Update")

    def onSimpan(self, event=None):
        ida = self.txtIDA.get()
        nama = self.txtNama.get()
        jk = self.txtJK.get()
        alamat = self.txtAlamat.get()
        buku = self.txtJudul.get()
        tahun = self.txtTahun.get()
        kategori = self.txtKategori.get()
        penulis = self.txtPenulis.get()
        penerbit = self.txtPenerbit.get()
        peminjaman = self.txtPeminjaman.get()
        pengembalian = self.txtPengembalian.get()
        telat = self.txtTelat.get()
        denda = self.txtDenda.get()

        prps = Perpustakaan()
        prps.ida = ida
        prps.nama = nama
        prps.jk = jk
        prps.alamat = alamat
        prps.buku = buku
        prps.tahunterbit = tahun
        prps.kategori = kategori
        prps.penulis = penulis
        prps.penerbit = penerbit
        prps.peminjaman = peminjaman
        prps.pengembalian = pengembalian
        prps.telat = telat
        prps.denda = denda
        if(self.ditemukan==True):
            res = prps.updateByNIM(ida)
            ket = 'Diperbarui'
        else:
            res = prps.simpan()
            ket = 'Disimpan'

        rec = prps.affected
        if(rec>0):
            messagebox.showinfo("showinfo", "Data Berhasil "+ket)
        else:
            messagebox.showwarning("showwarning", "Data Gagal "+ket)
        self.onClear()
        return rec

    def onDelete(self, event=None):
        ida = self.txtIDA.get()
        prps = Perpustakaan()
        prps.ida = ida
        if(self.ditemukan==True):
            res = prps.deleteByNIM(ida)
            rec = prps.affected
        else:
            messagebox.showinfo("showinfo", "Data harus ditemukan dulu sebelum
dihapus")
            rec = 0

        if(rec>0):
            messagebox.showinfo("showinfo", "Data Berhasil dihapus")

        self.onClear()

    def onKeluar(self, event=None):
        # memberikan perintah menutup aplikasi
        self.parent.destroy()

if __name__ == '__main__':
    root2 = tk.Tk()
    aplikasi = FrmPerpustakaan,(root2, "Aplikasi Data Perpustakaan")
    root2.mainloop()
```

Tampilan 2:

```
import tkinter as tk
from tkinter import
Frame,Label,Entry,Button,Radiobutton,ttk,VERTICAL,YES,BOTH,END,Tk,W,StringVar,mes
sagebox
from Buku import Buku
import os


os.system("cls")

class FrmBuku:

    def __init__(self, parent, title):
        self.parent = parent
        self.parent.geometry("700x450")
        self.parent.title(title)
        self.parent.protocol("WM_DELETE_WINDOW", self.onKeluar)
        self.ditemukan = None
        self.aturKomponen()
        self.onReload()

    def aturKomponen(self):
        mainFrame = Frame(self.parent, bd=10)
        mainFrame.pack(fill=BOTH, expand=YES)
    # Label
    Label(mainFrame, text='❤Daftar Buku Perpustakaan
UMC  ❤').grid(row=15, column=1,
      sticky=W, padx=5, pady=5)
    Label(mainFrame, text='Kode Buku      :').grid(row=1, column=0,
      sticky=W, padx=5, pady=5)
    Label(mainFrame, text='Judul           :').grid(row=2,
column=0,
      sticky=W, padx=5, pady=5)
    Label(mainFrame, text='Penulis         :').grid(row=3, column=0,
      sticky=W, padx=5, pady=5)
    Label(mainFrame, text='Penerbit & Tahun:').grid(row=4, column=0,
      sticky=W, padx=5, pady=5)
    Label(mainFrame, text='Kategori        :').grid(row=5, column=0,
      sticky=W, padx=5, pady=5)

    # Textbox
    self.txtKodebuku = Entry(mainFrame)
    self.txtKodebuku.grid(row=1, column=1, padx=5, pady=5)
    self.txtKodebuku.bind("<Return>",self.onCari) # menambahkan event Enter
key

    self.txtJudul = Entry(mainFrame)
    self.txtJudul.grid(row=2, column=1, padx=5, pady=5)

    self.txtPenulis = Entry(mainFrame)
    self.txtPenulis.grid(row=3, column=1, padx=5, pady=5)

    self.txtPenerbit = Entry(mainFrame)
    self.txtPenerbit.grid(row=4, column=1, padx=5, pady=5)

    self.txtTahun = Entry(mainFrame)
    self.txtTahun.grid(row=4, column=2, padx=5, pady=5)

    # Radio Button
    #self.txtpenulis = StringVar()
    #self.L = Radiobutton(mainFrame, text='Laki-laki', value='L',
variable=self.txtpenulis)
    #self.L.grid(row=3, column=1, padx=5, pady=5, sticky=W)
    #self.L.select() # set pilihan yg pertama
    #self.P = Radiobutton(mainFrame, text='Perempuan', value='P',
variable=self.txtpenulis)
    #self.P.grid(row=4, column=1, padx=5, pady=5, sticky=W)
```

```
        # Combo Box
        self.txtKategori = StringVar()
        Cbo = ttk.Combobox(mainFrame, width = 17, textvariable =
self.txtKategori)
        Cbo.grid(row=5, column=1, padx=5, pady=5)
        # Adding combobox drop down list
        Cbo['values'] = ('Novel','Majalah', 'Kamus','Komik','Manga',
'Ensiklopedia', 'Biografi', 'Naskah', 'Light Novel')
        Cbo.current()



        # Button
        self.btnSimpan = Button(mainFrame, text='Save', command=self.onSimpan,
width=10,  fg= "white", bg="blue")
        self.btnSimpan.grid(row=6, column=0, padx=5, pady=5)
        self.btnClear = Button(mainFrame, text='Clear', command=self.onClear,
width=10,  fg= "black", bg="yellow")
        self.btnClear.grid(row=6, column=1, padx=5, pady=5)
        self.btnHapus = Button(mainFrame, text='Delete', command=self.onDelete,
width=10,  fg= "white", bg="red")
        self.btnHapus.grid(row=6, column=2, padx=5, pady=5)
        self.btnCari = Button(mainFrame, text='Cari Buku', command=self.onCari,
width=10,  fg= "white", bg="green")
        self.btnCari.grid(row=1, column=2, padx=5, pady=5)

        # define columns
        columns = ('idbk', 'kodebuku', 'judul','penulis','penerbit', 'tahun',
'kategori')

        self.tree = ttk.Treeview(mainFrame, columns=columns, show='headings')
        # define headings
        self.tree.heading('idbk', text='No')
        self.tree.column('idbk', width="30")
        self.tree.heading('kodebuku', text='Kode')
        self.tree.column('kodebuku', width="60")
        self.tree.heading('judul', text='Judul')
        self.tree.column('judul', width="200")
        self.tree.heading('penulis', text='Penulis')
        self.tree.column('penulis', width="100")
        self.tree.heading('penerbit', text='Penerbit')
        self.tree.column('penerbit', width="100")
        self.tree.heading('tahun', text='Tahun')
        self.tree.column('tahun', width="100")
        self.tree.heading('kategori', text='Kategori')
        self.tree.column('kategori', width="100")
        # set tree position
        self.tree.place(x=0, y=230)
        self.onReload()

    def onClear(self, event=None):
        self.txtKodebuku.delete(0,END)
        self.txtKodebuku.insert(END,"")
        self.txtJudul.delete(0,END)
        self.txtJudul.insert(END,"")
        self.txtPenulis.delete(0,END)
        self.txtPenulis.insert(END,"")
        self.txtPenerbit.delete(0,END)
        self.txtPenerbit.insert(END,"")
        self.txtTahun.delete(0,END)
        self.txtTahun.insert(END,"")
        self.txtKategori.set("")

        self.btnSimpan.config(text="Simpan")

        self.onReload()
        self.ditemukan = False

    def onReload(self, event=None):
        # get data mahasiswa
        bk = Buku()
        result = bk.getAllData()
        for item in self.tree.get_children():
            self.tree.delete(item)
        students=[]
        for row_data in result:
            students.append(row_data)

        for student in students:
            self.tree.insert('',END, values=student)

    def onCari(self, event=None):
        kodebuku = self.txtKodebuku.get()
        bk = Buku()
        res = bk.getByNIM(kodebuku)
        rec = bk.affected
        if(rec>0):
            messagebox.showinfo("showinfo", "Data Ditemukan")
            self.TampilkanData()
            self.ditemukan = True
        else:
            messagebox.showwarning("showwarning", "Data Tidak Ditemukan")
            self.ditemukan = False
            self.txtJudul.focus()
        return res

    def TampilkanData(self, event=None):
        kodebuku = self.txtKodebuku.get()
        bk = Buku()
        res = bk.getByNIM(kodebuku)
        self.txtJudul.delete(0,END)
        self.txtJudul.insert(END,bk.judul)
        self.txtPenulis.delete(0,END)
        self.txtPenulis.insert(END,bk.penulis)
        self.txtPenerbit.delete(0,END)
        self.txtPenerbit.insert(END,bk.penerbit)
        self.txtTahun.delete(0,END)
        self.txtTahun.insert(END,bk.tahun)

        self.txtKategori.set(bk.kategori)
        self.btnSimpan.config(text="Update")

    def onSimpan(self, event=None):
        kodebuku = self.txtKodebuku.get()
        judul = self.txtJudul.get()
        penulis = self.txtPenulis.get()
        penerbit = self.txtPenerbit.get()
        tahun = self.txtTahun.get()
        kategori = self.txtKategori.get()

        bk = Buku()
        bk.kodebuku = kodebuku
        bk.judul = judul
        bk.penulis = penulis
        bk.penerbit = penerbit
        bk.tahun = tahun
        bk.kategori = kategori
        if(self.ditemukan==True):
            res = bk.updateByNIM(kodebuku)
            ket = 'Diperbarui'
        else:
            res = bk.simpan()
            ket = 'Disimpan'
        rec = bk.affected
        if(rec>0):
            messagebox.showinfo("showinfo", "Data Berhasil "+ket)
        else:
            messagebox.showwarning("showwarning", "Data Gagal "+ket)
        self.onClear()
        return rec

    def onDelete(self, event=None):
        kodebuku = self.txtKodebuku.get()
        bk = Buku()
        bk.kodebuku = kodebuku
        if(self.ditemukan==True):
            res = bk.deleteByNIM(kodebuku)
            rec = bk.affected
        else:
            messagebox.showinfo("showinfo", "Data harus ditemukan dulu sebelum
dihapus")
            rec = 0

        if(rec>0):
            messagebox.showinfo("showinfo", "Data Berhasil dihapus")

        self.onClear()

    def onKeluar(self, event=None):
        # memberikan perintah menutup aplikasi
        self.parent.destroy()

if __name__ == '__main__':
    root2 = tk.Tk()
    aplikasi = FrmBuku(root2, "Aplikasi Data Buku")
    root2.mainloop()
```

Tampilan Dasboard:

```
import tkinter as tk
from tkinter import Menu
from FrmPerpustakaan import *
from FrmKalori import *
from FrmSuhu import *
from FrmBuku import *
from FrmBio import *
```

```
# root window
root = tk.Tk()
root.title('Tugas Kelompok')
#root.attributes('-fullscreen', True)
root.geometry("500x400")
# create a menubar
menubar = Menu(root)
root.config(menu=menubar)

# create a menu
file_menu = Menu(menubar)
app_menu = Menu(menubar)
data_menu = Menu(menubar)

# Menu File
file_menu.add_command(
    label='BiodataKu', command= lambda: new_window("Data Pengembang", FrmBio)
)

file_menu.add_command(
    label='Exit', command=root.destroy
)

# Menu App
app_menu.add_command(
    label='App Suhu', command= lambda: new_window("Mengkonversi Suhu", FormSuhu)
)
app_menu.add_command(
    label='App Kalori', command= lambda: new_window("Mengecek Kalori",
FormKalori)
)

# Menu Data
data_menu.add_command(
    label='Data Anggota Perpustakaan & Peminjaman', command= lambda:
new_window("Data Perpustakaan Universitas Muhammadiyah Cirebon", FrmPerpustakaan)
)
data_menu.add_command(
    label='Data Buku Perpustakaan', command= lambda: new_window("Data Buku
Perpustakaan Universitas Muhammadiyah Cirebon", FrmBuku)
)
```

```python
def new_window( number, _class):
    new = tk.Toplevel()
    new.transient()
    new.grab_set()
    _class(new, number)

# add the File menu to the menubar
menubar.add_cascade(
    label="File", menu=file_menu
)
menubar.add_cascade(
    label="App", menu=app_menu
)
menubar.add_cascade(
    label="Data", menu=data_menu
)

# Menu Data is Disabled
# menubar.entryconfig('Data',state="disabled")

menubar.entryconfig('Data',state="normal")

root.mainloop()
```

Tampilan Tambahan1:

```python
from tkinter import Frame,Label,YES,BOTH,Tk,W

class FrmBio:
    def __init__(self, parent, title):
        self.parent = parent

        self.parent.title(title)
        self.parent.protocol("WM_DELETE_WINDOW")
        self.aturKomponen()

    def aturKomponen(self):
        mainFrame = Frame(self.parent, bd=10)
        mainFrame.pack(fill=BOTH, expand=YES)
#=================================================================================
=============
        Label(mainFrame, text="Program Perpustakaan").grid(row=0, column=1,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text=" ").grid(row=1, column=1,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Nama                 : ').grid(row=2, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Rifki Fadilah (210511011)').grid(row=2, column=1,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Kelas                  : ").grid(row=5, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Reguler 1 (TI21A)").grid(row=5, column=1,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Semester           : ").grid(row=6, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="3 (Tiga)").grid(row=6, column=1,
            sticky=W, padx=5, pady=5)

        Label(mainFrame, text="="*20).grid(row=8, column=1,
            sticky=W, padx=5, pady=5)




if __name__ == '__main__':
    root = Tk()
    aplikasi = FrmBio(root, "Biodataku")
    root.mainloop()
```

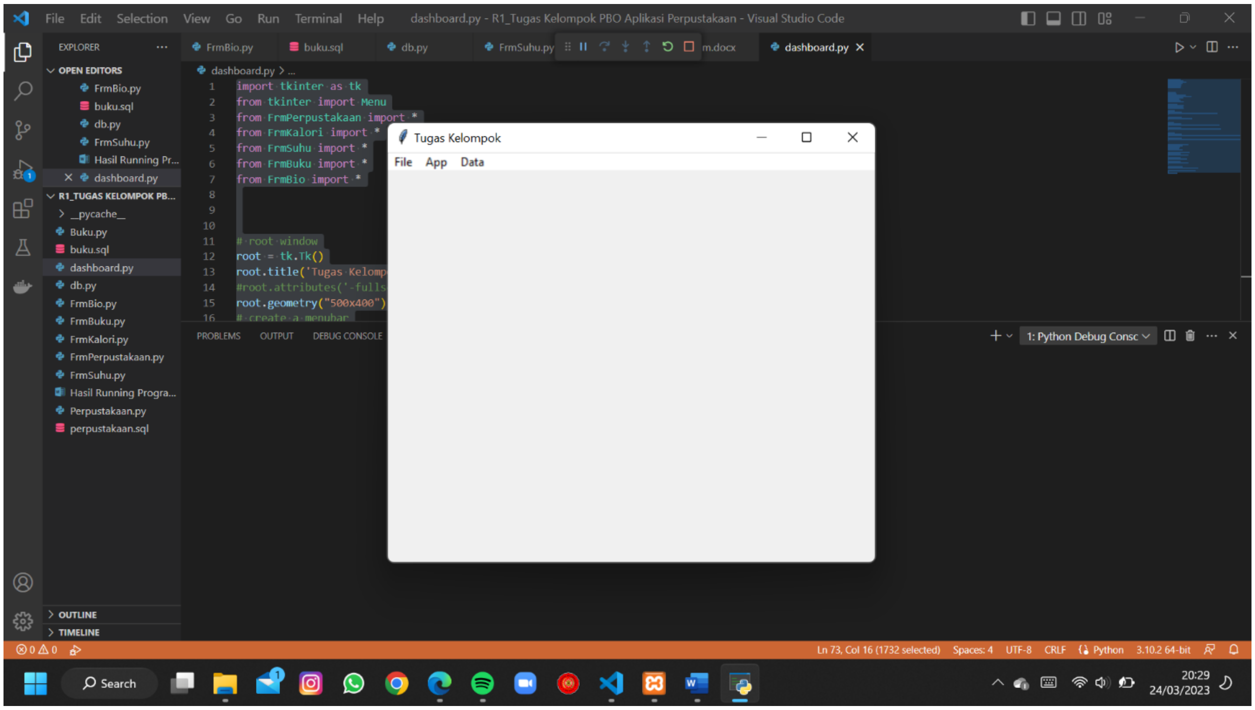
Fungsi dan Tampilan Tambahan 1:

```
'''
Nama    : Rifki Fadilah
Kelas   : R1
NIM     : 210511011
'''

from tkinter import Frame,Label,Entry,Button,YES,BOTH,END,Tk,W
from tkinter.messagebox import NO

class FormKalori:
    def __init__(self, parent, title):
        self.parent = parent
        self.parent.title(title)
        self.parent.protocol("WM_DELETE_WINDOW", self.onKeluar)
        self.aturKomponen()

    def aturKomponen(self):
        mainFrame = Frame(self.parent, bd=5)
        mainFrame.pack(fill=BOTH, expand=NO)
#================================================================================
=============
        Label(mainFrame, text="Masukkan data dengan benar!").grid(row=0,
column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Berat Badan:').grid(row=1, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Tinggi Badan:").grid(row=3, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Usia Saat Ini:").grid(row=5, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="="*40).grid(row=7, column=0,
            sticky=W, padx=0, pady=0)
        Label(mainFrame, text="Kebutuhan Kalori=").grid(row=9, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Keterangan=").grid(row=10, column=0,
            sticky=W, padx=5, pady=5)

#================================================================================
=============

        self.txtBerat = Entry(mainFrame)
        self.txtBerat.grid(row=1, column=1, padx=5, pady=5)
        self.txtTinggi = Entry(mainFrame)
        self.txtTinggi.grid(row=3, column=1, padx=5, pady=5)
        self.txtUsia = Entry(mainFrame)
        self.txtUsia.grid(row=5, column=1, padx=5, pady=5)
        self.txtKalori = Entry(mainFrame)
        self.txtKalori.grid(row=9, column=1, padx=5, pady=5)
        self.txtKeterangan = Entry(mainFrame)
        self.txtKeterangan.grid(row=10, column=1, padx=5, pady=5)

#================================================================================
=============

        self.btnHitung = Button(mainFrame, text='Cek Sebagai Pria',  fg= "white",
bg="blue",
            command=self.onHitung)
        self.btnHitung.grid(row=11, column=0, padx=5, pady=5)

        self.btnHitung1 = Button(mainFrame, text='Cek Sebagai Wanita', fg=
"white", bg="red",
            command=self.onHitung1)
        self.btnHitung1.grid(row=11, column=1, padx=5, pady=5)



#================================================================================
=============

    def onHitung(self, event=None):
        berat = float(self.txtBerat.get())
        tinggi = float(self.txtTinggi.get())
        usia = float(self.txtUsia.get())
        kalorip = (88.4 + 13.4 * berat) + (4.8 * tinggi) - (5.68 * usia)
        if (kalorip>=2500):
            keterangan='Ideal'
        elif (kalorip>=1500):
            keterangan='Cukup'
        elif (kalorip>=1300):
            keterangan='Kurang'
        elif (kalorip>=1100):
            keterangan='Sangat Kurang'
        else :
            keterangan='Anda Harus Ke Dokter'
        self.txtKalori.delete(0,END)
        self.txtKalori.insert(END,str(kalorip))
        self.txtKeterangan.delete(0,END)
        self.txtKeterangan.insert(END,str(keterangan))


    def onHitung1(self, event=None):
        berat = float(self.txtBerat.get())
        tinggi = float(self.txtTinggi.get())
        usia = float(self.txtUsia.get())
        kaloriw = (447.6 + 9.25 * berat) + (3.10 * tinggi) - (4.33 * usia)
        if (kaloriw>=2000):
            keterangan='Ideal'
        elif (kaloriw>=1400):
            keterangan='Cukup'
        elif (kaloriw>=1200):
            keterangan='Kurang'
        elif (kaloriw>=900):
            keterangan='Sangat Kurang'
        else :
            keterangan='Anda Harus Ke Dokter'
        self.txtKalori.delete(0,END)
        self.txtKalori.insert(END,str(kaloriw))
        self.txtKeterangan.delete(0,END)
        self.txtKeterangan.insert(END,str(keterangan))

#================================================================================
=============
    def onKeluar(self, event=None):

        self.parent.destroy()

if __name__ == '__main__':
    root = Tk()
    aplikasi = FormKalori(root, "Menghitung Kebutuhan Kalori Harian")
    root.mainloop()
```

Fungsi dan Tampilan Tambahan 2:

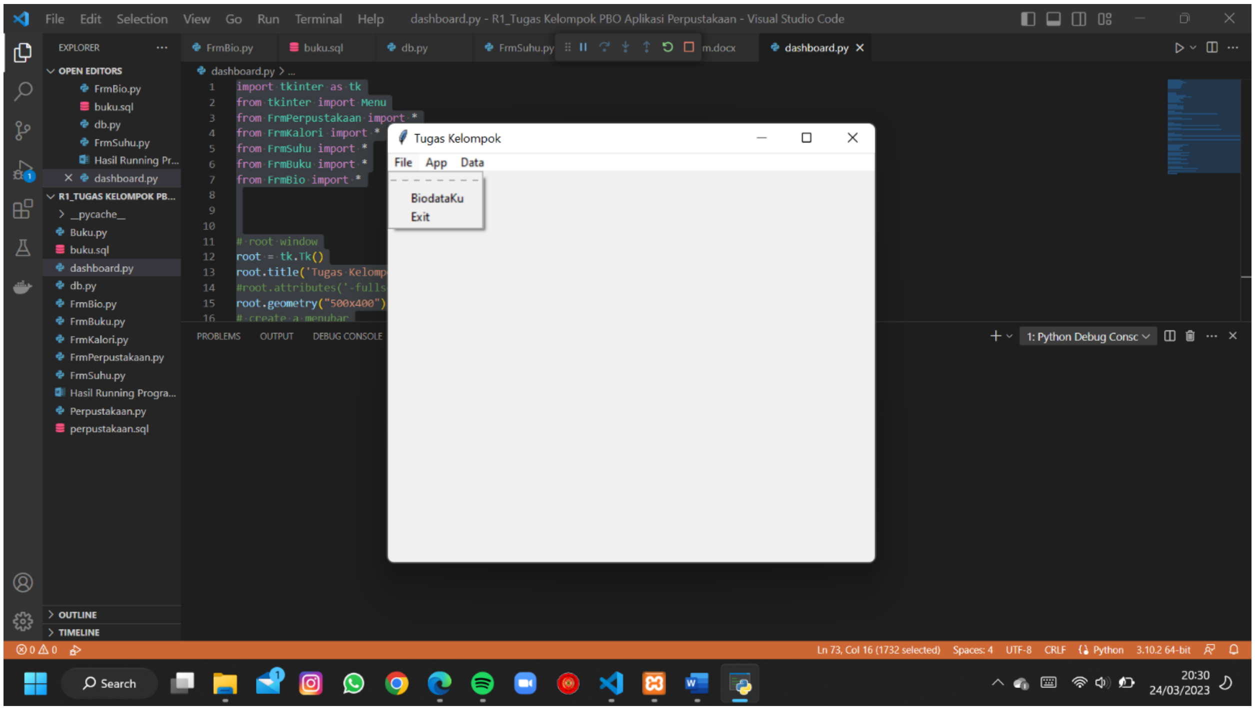
```
'''
Nama    : Rifki Fadilah
Kelas   : R1
NIM     : 210511011
'''

from tkinter import Frame,Label,Entry,Button,YES,BOTH,END,Tk,W

class FormSuhu:
    def __init__(self, parent, title):
        self.parent = parent

        self.parent.title(title)
        self.parent.protocol("WM_DELETE_WINDOW", self.onKeluar)
        self.aturKomponen()

    def aturKomponen(self):
        mainFrame = Frame(self.parent, bd=10)
        mainFrame.pack(fill=BOTH, expand=YES)
#================================================================================
=============
        Label(mainFrame, text="Mari Mengkonversi Suhu💗").grid(row=0, column=1,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Celcius :').grid(row=1, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text='Farenheit  :').grid(row=3, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Kelvin :").grid(row=5, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Reamur :").grid(row=7, column=0,
            sticky=W, padx=5, pady=5)

        Label(mainFrame, text="="*20).grid(row=8, column=0,
            sticky=W, padx=5, pady=5)

        Label(mainFrame, text="Suhu Dalam Celcius =").grid(row=9, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Suhu Dalam Reamur =").grid(row=10, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Suhu Dalam Farenheit =").grid(row=11, column=0,
            sticky=W, padx=5, pady=5)
        Label(mainFrame, text="Suhu Dalam Kelvin =").grid(row=12, column=0,
            sticky=W, padx=5, pady=5)

#================================================================================
=============
        self.txtCelcius = Entry(mainFrame)
        self.txtCelcius.grid(row=1, column=1, padx=5, pady=5)

        self.txtFarenheit = Entry(mainFrame)
        self.txtFarenheit.grid(row=3, column=1, padx=5, pady=5)

        self.txtKelvin = Entry(mainFrame)
        self.txtKelvin.grid(row=5, column=1, padx=5, pady=5)

        self.txtReamur = Entry(mainFrame)
        self.txtReamur.grid(row=7, column=1, padx=5, pady=5)

        self.txtDalamCelcius = Entry(mainFrame)
        self.txtDalamCelcius.grid(row=9, column=1, padx=5, pady=5)

        self.txtDalamReamur = Entry(mainFrame)
        self.txtDalamReamur.grid(row=10, column=1, padx=5, pady=5)

        self.txtDalamFarenheit = Entry(mainFrame)
        self.txtDalamFarenheit.grid(row=11, column=1, padx=5, pady=5)
        self.txtDalamKelvin = Entry(mainFrame)
        self.txtDalamKelvin.grid(row=12, column=1, padx=5, pady=5)

#================================================================================
=============

        self.btnHitung = Button(mainFrame, text='Konversikan Suhu Celcius',  fg=
"white", bg="blue",
            command=self.onHitung3)
        self.btnHitung.grid(row=7, column=2, padx=5, pady=5)

        self.btnHitung = Button(mainFrame, text='Konversikan Suhu Farenheit', fg=
"black", bg="red",
            command=self.onHitung)
        self.btnHitung.grid(row=8, column=2, padx=5, pady=5)

        self.btnHitung = Button(mainFrame, text='Konversikan Suhu Kelvin',  fg=
"white", bg="green",
            command=self.onHitung1)
        self.btnHitung.grid(row=9, column=2, padx=5, pady=5)

        self.btnHitung = Button(mainFrame, text='Konversikan Suhu Reamur',  fg=
"black", bg="yellow",
            command=self.onHitung2)
        self.btnHitung.grid(row=10, column=2, padx=5, pady=5)


#================================================================================
=============

    def onHitung(self, event=None):
        farenheit = float(self.txtFarenheit.get())

        celcius1 = 5/9 * (farenheit - 32)
        kelvin1 = 5/9 * (farenheit - 32) +273
        reamur1 = 4/9 * (farenheit - 32)
        self.txtDalamCelcius.delete(0,END)
        self.txtDalamCelcius.insert(END,str(celcius1))
        self.txtDalamKelvin.delete(0,END)
        self.txtDalamKelvin.insert(END,str(kelvin1))
        self.txtDalamReamur.delete(0,END)
        self.txtDalamReamur.insert(END,str(reamur1))

    def onHitung1(self, event=None):
        kelvin = float(self.txtKelvin.get())

        celcius2 = kelvin - 273
        farenheit1 = 9/5 * (kelvin - 273) + 32
        reamur2 = 4/5 * (kelvin - 273)
        self.txtDalamCelcius.delete(0,END)
        self.txtDalamCelcius.insert(END,str(celcius2))
        self.txtDalamFarenheit.delete(0,END)
        self.txtDalamFarenheit.insert(END,str(farenheit1))
        self.txtDalamReamur.delete(0,END)
        self.txtDalamReamur.insert(END,str(reamur2))

    def onHitung2(self, event=None):
        reamur = float(self.txtReamur.get())

        celcius3 = 5/4 * reamur
        farenheit2 = (9/4 * reamur) + 32
        kelvin2 = (5/4 * reamur) + 273
        self.txtDalamCelcius.delete(0,END)
        self.txtDalamCelcius.insert(END,str(celcius3))
        self.txtDalamFarenheit.delete(0,END)
        self.txtDalamFarenheit.insert(END,str(farenheit2))
        self.txtDalamKelvin.delete(0,END)
        self.txtDalamKelvin.insert(END,str(kelvin2,))

    def onHitung3(self, event=None):
        celcius = float(self.txtCelcius.get())

        farenheit3 = (9/5 * celcius) + 32
        kelvin3 = celcius + 273
        reamur3 = 4/5 * celcius
        self.txtDalamFarenheit.delete(0,END)
        self.txtDalamFarenheit.insert(END,str(farenheit3))
        self.txtDalamKelvin.delete(0,END)
        self.txtDalamKelvin.insert(END,str(kelvin3))
        self.txtDalamReamur.delete(0,END)
        self.txtDalamReamur.insert(END,str(reamur3))

#================================================================================
=============
    def onKeluar(self, event=None):

        self.parent.destroy()

if __name__ == '__main__':
    root = Tk()
    aplikasi = FormSuhu(root, "Aplikasi Mengkonversi Suhu by Rifki Fadilah 
210511011")
    root.mainloop()
```

Hasil Running Program:

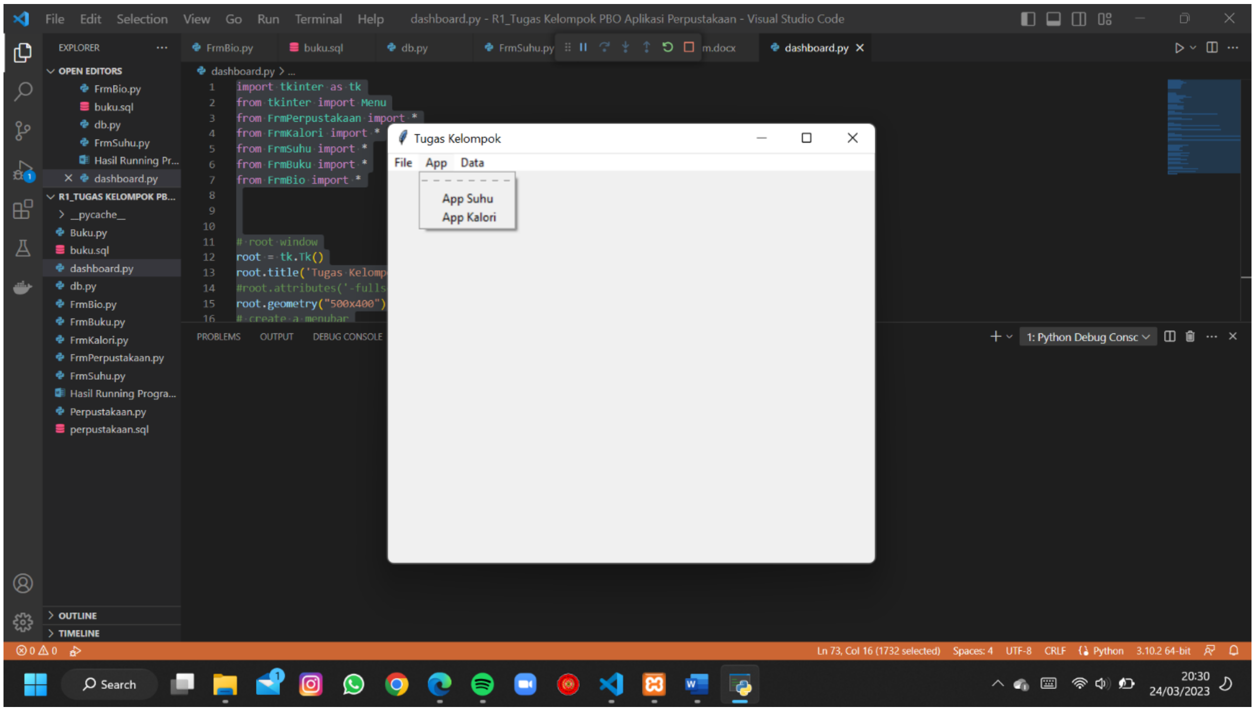


```
import tkinter as tk
from tkinter import Menu
from FrmPerpustakaan import *
from FrmKalori import *
from FrmSuhu import *
from FrmBuku import *
from FrmBio import *



# root window
root = tk.Tk()
root.title('Tugas Kelomp
#root.attributes('-fulls
root.geometry("500x400")
# create a menubar
```

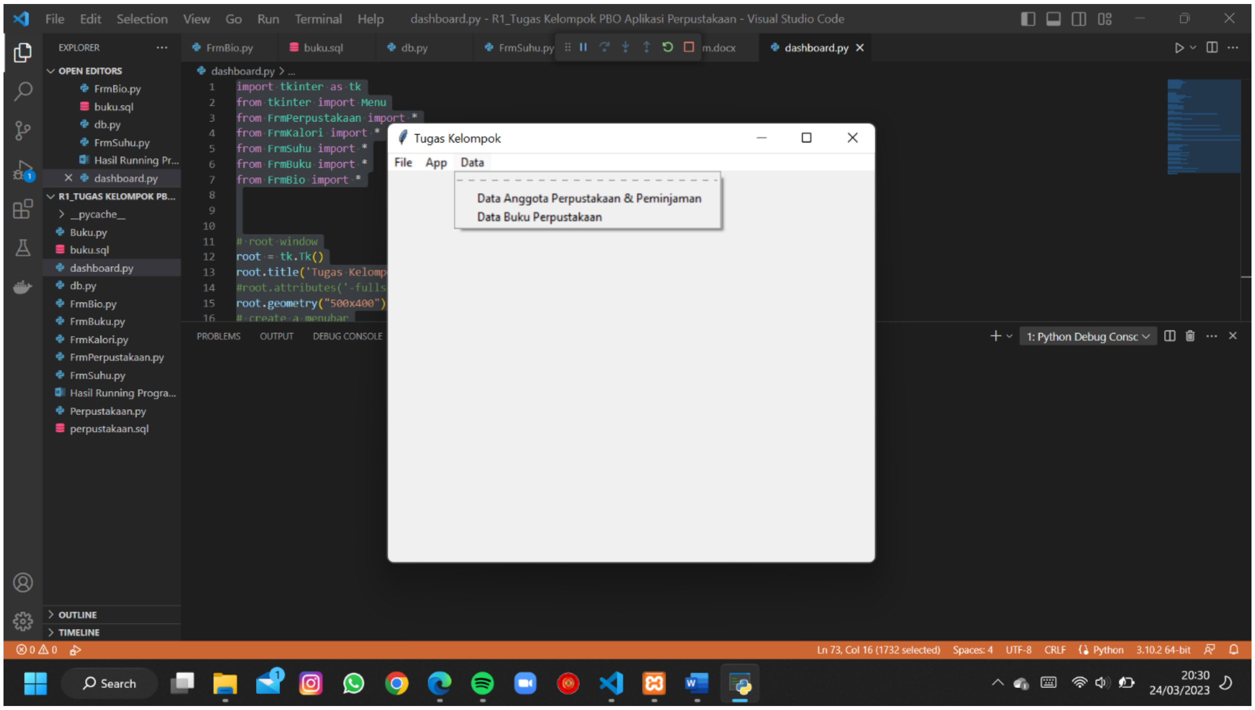
dashboard.py - R1_Tugas Kelompok PBO Aplikasi Perpustakaan - Visual Studio Code
Tugas Kelompok
File
App
Data



```
import tkinter as tk
from tkinter import Menu
from FrmPerpustakaan import *
from FrmKalori import *
from FrmSuhu import *
from FrmBuku import *
from FrmBio import *



# root window
root = tk.Tk()
root.title('Tugas Kelomp
#root.attributes('-fulls
root.geometry("500x400")
# create a menubar
```

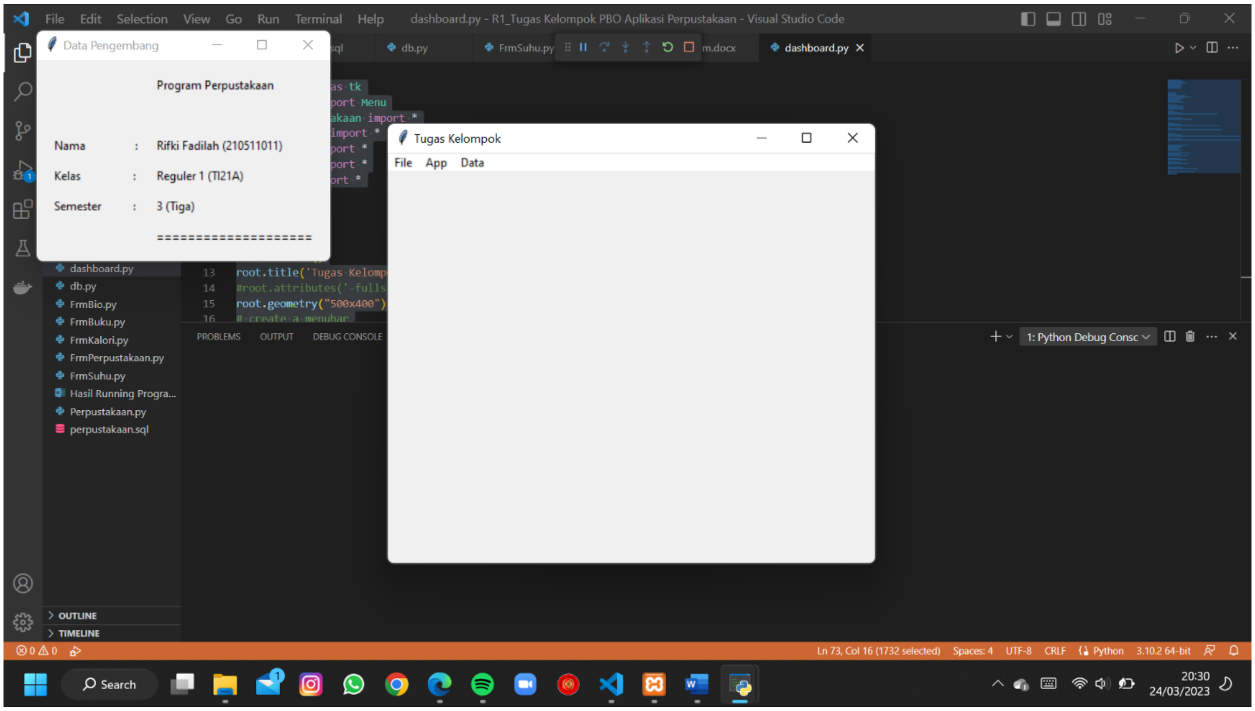


```
import tkinter as tk
from tkinter import Menu
from FrmPerpustakaan import *
from FrmKalori import *
from FrmSuhu import *
from FrmBuku import *
from FrmBio import *



# root window
root = tk.Tk()
root.title('Tugas Kelomp
#root.attributes('-fulls
root.geometry("500x400")
# create a menubar
```


dashboard.py - R1_Tugas Kelompok PBO Aplikasi Perpustakaan - Visual Studio Code
FrmBio.py
buku.sql
db.py
FrmSuhu.py
dashboard.py
Tugas Kelompok
File
App
Data

```
import tkinter as tk
from tkinter import Menu
from FrmPerpustakaan import *
from FrmKalori import *
from FrmSuhu import *
from FrmBuku import *
from FrmBio import *



# root window
root = tk.Tk()
root.title('Tugas Kelomp
#root.attributes('-fulls
root.geometry("500x400")
# create a menubar
```

Data Anggota Perpustakaan & Peminjaman
Data Buku Perpustakaan


dashboard.py - R1_Tugas Kelompok PBO Aplikasi Perpustakaan - Visual Studio Code
Data Pengembang
Tugas Kelompok
File
App
Data

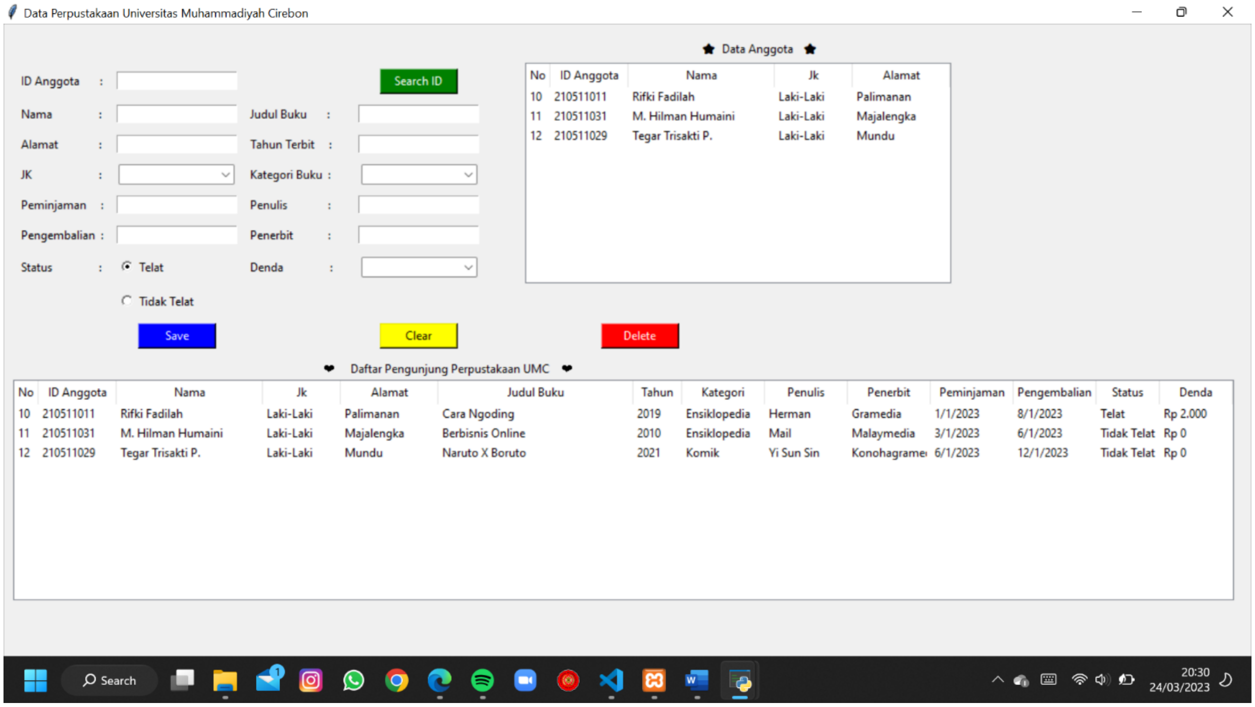
Program Perpustakaan

Nama : Rifki Fadilah (210511011)

Kelas : Reguler 1 (TI21A)

Semester : 3 (Tiga)

====================



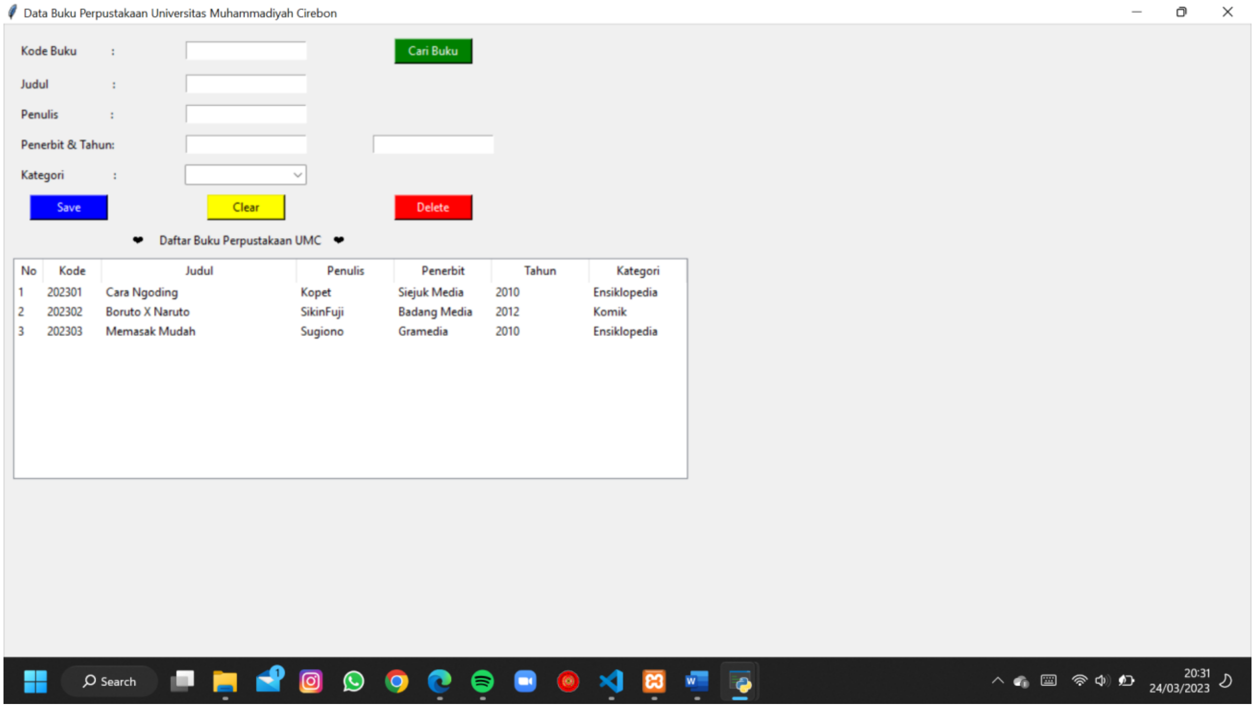
Status : ● Telat ○ Tidak Telat

★ Data Anggota ★

No	ID Anggota	Nama	Jk	Alamat
10	210511011	Rifki Fadilah	Laki-Laki	Palimanan
11	210511031	M. Hilman Humaini	Laki-Laki	Majalengka
12	210511029	Tegar Trisakti P.	Laki-Laki	Mundu

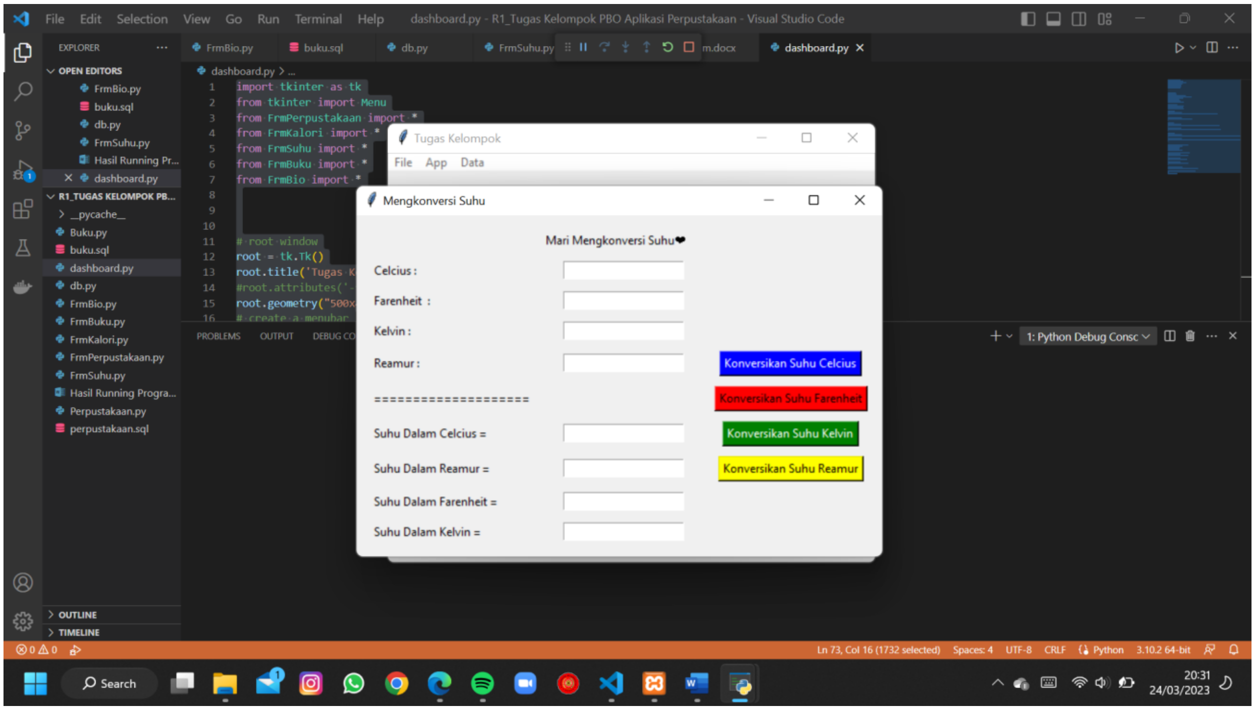
❤ Daftar Pengunjung Perpustakaan UMC ❤

No	ID Anggota	Nama	Jk	Alamat	Judul Buku	Tahun	Kategori	Penulis	Penerbit	Peminjaman	Pengembalian	Status	Denda
10	210511011	Rifki Fadilah	Laki-Laki	Palimanan	Cara Ngoding	2019	Ensiklopedia	Herman	Gramedia	1/1/2023	8/1/2023	Telat	Rp 2.000
11	210511031	M. Hilman Humaini	Laki-Laki	Majalengka	Berbisnis Online	2010	Ensiklopedia	Mail	Malaymedia	3/1/2023	6/1/2023	Tidak Telat	Rp 0
12	210511029	Tegar Trisakti P.	Laki-Laki	Mundu	Naruto X Boruto	2021	Komik	Yi Sun Sin	Konohagrame	6/1/2023	12/1/2023	Tidak Telat	Rp 0



❤ Daftar Buku Perpustakaan UMC ❤

No	Kode	Judul	Penulis	Penerbit	Tahun	Kategori
1	202301	Cara Ngoding	Kopet	Siejuk Media	2010	Ensiklopedia
2	202302	Boruto X Naruto	SikinFuji	Badang Media	2012	Komik
3	202303	Memasak Mudah	Sugiono	Gramedia	2010	Ensiklopedia



```
import tkinter as tk
from tkinter import Menu
from FrmPerpustakaan import *
from FrmKalori import *
from FrmSuhu import *
from FrmBuku import *
from FrmBio import *



# root window
root = tk.Tk()
root.title('Tugas K
#root.attributes('-
root.geometry("500x
# create a menubar
```

Mengkonversi Suhu

Mari Mengkonversi Suhu❤

Celcius :

Farenheit :

Kelvin :

Reamur :

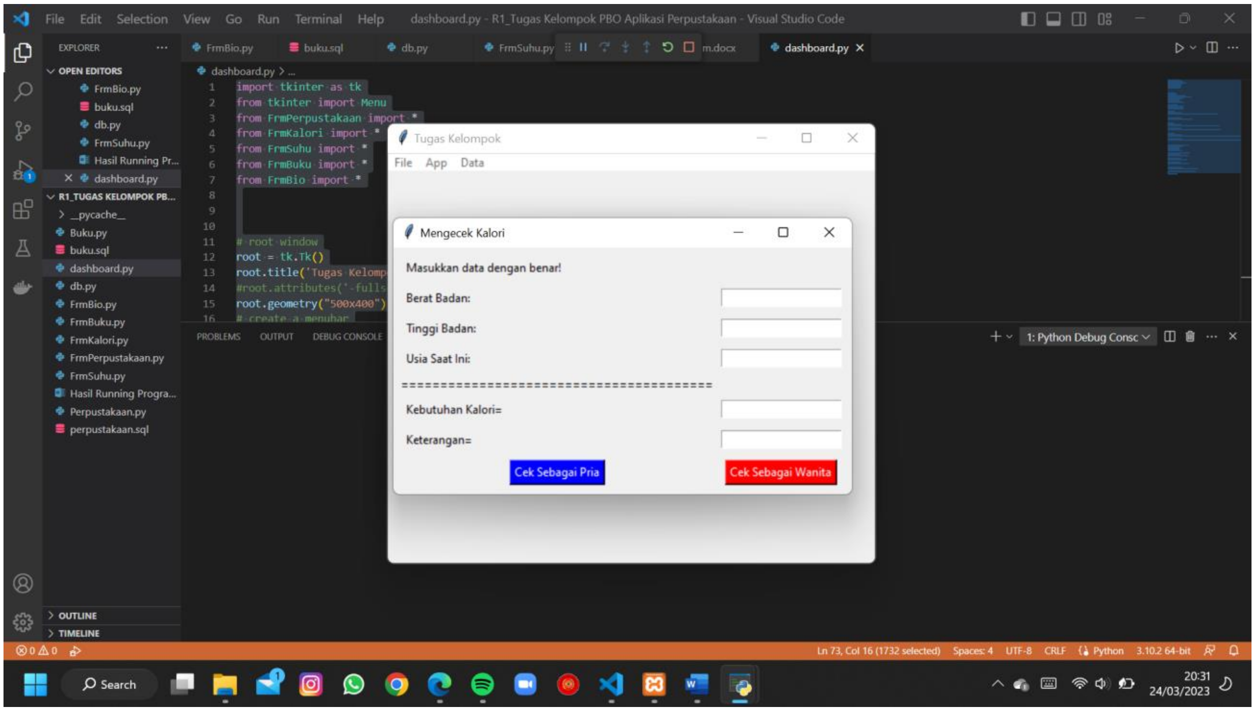
===================

Suhu Dalam Celcius =

Suhu Dalam Reamur =

Suhu Dalam Farenheit =

Suhu Dalam Kelvin =



```
import tkinter as tk
from tkinter import Menu
from FrmPerpustakaan import *
from FrmKalori import *
from FrmSuhu import *
from FrmBuku import *
from FrmBio import *



# root window
root = tk.Tk()
root.title('Tugas Kelomp
#root.attributes('-fulls
root.geometry("500x400")
# create a menubar
```

Mengecek Kalori

Masukkan data dengan benar!

Berat Badan:

Tinggi Badan:

Usia Saat Ini:

==

Kebutuhan Kalori=

Keterangan=