Nama : MUHAMMAD SHIDDIQ

NPM : 5220411288

Database :

A screenshot of a computer

Description automatically generated

tb\_produk:

A screenshot of a menu

Description automatically generated

tb\_transaksi:

A screenshot of a computer

Description automatically generated

tb\_karyawan:

A screenshot of a computer

Description automatically generated

CRUD

Create:

Untuk tb\_produk:

A screenshot of a computer

Description automatically generated

Untuk tb\_karyawan

A screenshot of a computer

Description automatically generated

Untuk tb\_transaksi dan tb\_detail\_transaksi

A screenshot of a computer

Description automatically generated

Read:

Untuk tb\_produk:

A screenshot of a computer

Description automatically generated

Untuk tb\_karyawan :

A screenshot of a computer

Description automatically generated

Update:

Untuk tb\_produk:

A screenshot of a computer

Description automatically generated

Untuk tb\_karyawan:

A screenshot of a computer

Description automatically generated

Delete:

A screenshot of a computer

Description automatically generated

Setelah di delete:

A screenshot of a computer

Description automatically generated

Code Program:

import mysql.connector

from datetime import datetime

class Posh\_Coffe:

    def \_\_init\_\_(*self*) -> None:

        self.localhost = "localhost"

        self.root = "root"

        self.password = ""

        self.db\_name = "5220411288"

        self.con = self.connection()

        self.cursor = self.db.cursor()

    def connection(*self*):

        db = mysql.connector.connect(

*host* = self.localhost,

*user* = self.root,

*password* = self.password,

*database* = self.db\_name

        )

        self.db = db

class Karyawan(Posh\_Coffe):

    def \_\_init\_\_(*self*) -> None:

        super().\_\_init\_\_()

    def tambah(*self*):

        header()

        nama\_karyawan = input('Masukkan Nama Karyawan   : ')

        no\_hp = int(input('Masukkan No HP    :'))

        query = "INSERT INTO tb\_karyawan (nama\_karyawan, no\_hp) VALUES (%s, %s)"

        values = (nama\_karyawan, no\_hp)

        self.cursor.execute(query, values)

        self.db.commit()

        print(f"Karyawan {nama\_karyawan} berhasil ditambahkan.")

        pilih = input('Ingin tambah karyawan lagi?(y/t)')

        if pilih == 'y':

            self.tambah()

        else:

            menu\_karyawan()

    def read(*self*):

        header()

        query = f"SELECT \* FROM tb\_karyawan"

        self.cursor.execute(query)

        all\_data = self.cursor.fetchall()

        if all\_data:

            for row in all\_data:

                print(f"ID Karyawan: {row[0]}")

                print(f"Nama Karyawan: {row[1]}")

                print(f"No HP: {row[2]}")

    def edit(*self*, *id\_karyawan*):

        query = f"SELECT \* FROM tb\_karyawan"

        self.cursor.execute(query)

        all\_data = self.cursor.fetchall()

        for row in all\_data:

            if row[0] == id\_karyawan :

                new\_nama\_karyawan = input("Nama Karyawan   : ")

                new\_no\_hp = input("No HP        : ")

                query = "UPDATE tb\_karyawan SET nama\_karyawan = %s, no\_hp = %s WHERE id\_karyawan = %s"

                values = (new\_nama\_karyawan, new\_no\_hp, id\_karyawan)

                self.cursor.execute(query, values)

                self.db.commit()

                print(f"Karyawan dengan ID {id\_karyawan} berhasil diperbarui.")

                break

        if row[0] != id\_karyawan:

            print('id Karyawan tidak ada')

            coba = input('Ingin pilih karyawan yang lain?(y/t)')

            if coba == 'y':

                id\_karyawan = int(input('Pilih ID karyawan yang ingin di edit  : '))

                untuk\_edit\_karyawan(id\_karyawan)

            else:

                menu\_karyawan()

        coba = input('Ingin pilih karyawan yang lain?(y/t)')

        if coba == 'y':

            id\_karyawan = int(input('Pilih ID karyawan yang ingin di edit  : '))

            untuk\_edit\_karyawan(id\_karyawan)

        else:

            menu\_karyawan()

class Produk(Posh\_Coffe):

    def \_\_init\_\_(*self*) -> None:

        super().\_\_init\_\_()

    def tambah(*self*):

        header()

        nama\_menu = input('Masukkan Nama Menu   : ')

        stok = input('Masukkan Stok     :')

        harga = input('Masukkan Harga(satuan)   : ')

        query = "INSERT INTO tb\_produk (nama\_menu, stok, harga) VALUES (%s, %s, %s)"

        values = (nama\_menu, stok, harga)

        self.cursor.execute(query, values)

        self.db.commit()

        print(f"Menu {nama\_menu} berhasil ditambahkan.")

        pilih = input('Ingin tambah menu lagi?(y/t)')

        if pilih == 'y':

            self.tambah()

        else:

            menu\_produk()

    def read(*self*):

        header()

        query = f"SELECT \* FROM tb\_produk"

        self.cursor.execute(query)

        all\_data = self.cursor.fetchall()

        if all\_data:

            for row in all\_data:

                print(f"ID Menu     : {row[0]}")

                print(f"Nama Produk : {row[1]}")

                print(f"Stok        : {row[2]}")

                print(f"Harga       : {row[3]}")

                garis()

    def read\_by\_index(*self*, *id\_menu*):

        query = f"SELECT \* FROM tb\_produk WHERE id\_menu = {id\_menu}"

        self.cursor.execute(query)

        result = self.cursor.fetchone()

        return result

    def edit(*self*, *id\_menu*):

        query = f"SELECT \* FROM tb\_produk"

        self.cursor.execute(query)

        all\_data = self.cursor.fetchall()

        for row in all\_data:

            if row[0] == id\_menu :

                new\_nama\_menu = input('Masukkan Nama Menu   : ')

                new\_stok = input('Masukkan Stok     :')

                new\_harga = input('Masukkan Harga(satuan)   : ')

                query = "UPDATE tb\_produk SET nama\_menu = %s, stok = %s, harga = %s WHERE id\_menu = %s"

                values = (new\_nama\_menu, new\_stok, new\_harga, id\_menu)

                self.cursor.execute(query, values)

                self.db.commit()

                print(f"Menu dengan ID {id\_menu} berhasil diperbarui.")

                break

        if row[0] != id\_menu:

            print('id menu tidak ada')

            print(garis())

            coba = input('Ingin pilih id menu yang lain?(y/t)')

            if coba == 'y':

                id\_menu = int(input('Pilih ID produk yang ingin di edit  : '))

                untuk\_edit\_produk(id\_menu)

            else:

                menu\_produk()

        coba = input('Ingin pilih id menu yang lain?(y/t)')

        if coba == 'y':

            id\_menu = int(input('Pilih ID produk yang ingin di edit  : '))

            untuk\_edit\_produk(id\_menu)

        else:

            menu\_produk()

    def hapus(*self*, *id\_menu*):

        query = f"SELECT \* FROM tb\_produk"

        self.cursor.execute(query)

        all\_data = self.cursor.fetchall()

        for row in all\_data:

            if row[0] == id\_menu :

                query = "DELETE FROM tb\_produk WHERE id\_menu = %s"

                self.cursor.execute(query, (id\_menu,))

                self.db.commit()

                print(f"Menu dengan ID {id\_menu} berhasil dihapus.")

                break

        else:

            print('id produk tidak ada')

            print(garis())

            coba = input('Ingin hapus produk yang lain?(y/t)')

            if coba == 'y':

                untuk\_edit\_produk()

            else:

                menu\_produk()

        coba = input('Ingin hapus produk yang lain?(y/t)')

        if coba == 'y':

            untuk\_hapus\_produk()

        else:

            menu\_produk()

class Transaksi(Posh\_Coffe):

    def \_\_init\_\_(*self*) -> None:

        super().\_\_init\_\_()

    def tambah(*self*):

        header()

        menu\_jumlah\_list = []

        cur\_harga = 0

        while True:

            produk.read()

            id\_menu = int(input("Masukkan ID Produk yang dibeli : "))

            if id\_menu == 0:

                break

            jumlah\_beli = int(input("Masukkan Jumlah Beli: "))

            produk\_info = produk.read\_by\_index(id\_menu)

            menu, harga, stok = produk\_info[1], produk\_info[3], produk\_info[2]

            if jumlah\_beli > stok:

                return "Stok tidak mencukupi untuk melakukan pembelian."

            new\_stok = stok - jumlah\_beli

            total\_harga = harga \* jumlah\_beli

            harga\_for\_struk = cur\_harga + total\_harga

            stok = new\_stok

            query\_edit\_stok = "UPDATE tb\_produk SET nama\_menu=%s , stok=%s , harga = %s WHERE id\_menu=%s"

            value = (menu, stok, harga, id\_menu)

            self.cursor.execute(query\_edit\_stok,value)

            self.db.commit()

            menu\_jumlah\_list.append((id\_menu, jumlah\_beli))

            piliih = input("Tambah Produk yang di beli?y/t")

            cur\_harga = total\_harga

            if piliih == 't':

                break

        query\_karyawan = "SELECT \* FROM tb\_karyawan LIMIT 1"

        self.cursor.execute(query\_karyawan)

        karyawan\_info = self.cursor.fetchone()

        id\_karyawan, nama\_karyawan = karyawan\_info[0], karyawan\_info[1]

        struk = f"===== Struk Pembelian =====\nKasir: {nama\_karyawan}\n"

        query\_transaksi = "INSERT INTO tb\_transaksi (id\_karyawan, tanggal) VALUES (%s, %s)"

        values\_transaksi = (id\_karyawan, datetime.now())

        self.cursor.execute(query\_transaksi, values\_transaksi)

        self.db.commit()

        id\_transaksi = self.cursor.lastrowid

        for id\_menu, jumlah\_beli in menu\_jumlah\_list:

            query\_detail\_transaksi = "INSERT INTO tb\_detail\_transaksi (id\_transaksi, id\_menu, jumlah\_beli, harga, total\_harga) VALUES (%s, %s, %s, %s, %s)"

            values\_detail\_transaksi = (id\_transaksi, id\_menu, jumlah\_beli, harga, harga\_for\_struk)

            self.cursor.execute(query\_detail\_transaksi, values\_detail\_transaksi)

            struk += f"\nMenu: {menu}\nJumlah Beli: {jumlah\_beli}\nHarga Satuan: {harga}\n"

        struk += f"\nTotal Harga: {harga\_for\_struk}\n============================="

        self.db.commit()

        print(struk)

        pil = input('ingin lakukan transaksi lagi?(y/t) ')

        if pil == 'y':

            transaksi.tambah()

        else:

            menu\_transaksi()

karyawan = Karyawan()

produk = Produk()

transaksi = Transaksi()

def header():

    garis()

    print(''\*5,'Posh Coffe')

    garis()

    print('\n')

def garis():

    return print('='\*20)

def menu():

    header()

    print('1. Produk')

    print('2. Karyawan')

    print('3. Transaksi')

    print('\n')

    garis()

    pilih = int(input('Pilihan  : '))

    garis()

    if pilih == 1:

        menu\_produk()

    elif pilih == 2:

        menu\_karyawan()

    elif pilih == 3:

        menu\_transaksi()

def menu\_produk():

    header()

    print('1. Tambah Data Produk')

    print('2. Lihat Data Produk')

    print('3. Edit Data Produk')

    print('4. Hapus Data Produk')

    print('5. Kembali ke menu utama')

    print('\n')

    garis()

    pilih = int(input('Pilihan  : '))

    garis()

    if pilih == 1:

        produk.tambah()

    elif pilih == 2:

        produk.read()

        coba = input('Kembali ke menu sebelumnya?(y/t)')

        if coba == 'y':

            menu\_produk()

    elif pilih == 3:

        produk.read()

        id\_menu = int(input('Pilih ID menu yang ingin di edit  : '))

        untuk\_edit\_produk(id\_menu)

    elif pilih == 4:

        produk.read()

        id\_menu = int(input('Pilih ID menu yang ingin di hapus  : '))

        untuk\_hapus\_produk(id\_menu)

    else:

        menu()

def menu\_karyawan():

    header()

    print('1. Tambah Data Karyawan')

    print('2. Lihat Data Karyawan')

    print('3. Edit Data Karyawan')

    print('4. Kembali ke menu utama')

    print('\n')

    garis()

    pilih = int(input('Pilihan  : '))

    garis()

    if pilih == 1:

        karyawan.tambah()

    elif pilih == 2:

        karyawan.read()

        coba = input('Kembali ke menu sebelumnya?(y/t)')

        if coba == 'y':

            menu\_karyawan()

    elif pilih == 3:

        karyawan.read()

        id\_karyawan = int(input('Pilih ID karyawan yang ingin di edit  : '))

        untuk\_edit\_karyawan(id\_karyawan)

    else:

        menu()

def menu\_transaksi():

    header()

    print('1. Tambah Transaksi')

    print('2. Kembali ke menu utama')

    print('\n')

    garis()

    pilih = int(input('Pilihan  : '))

    garis()

    if pilih == 1:

        transaksi.tambah()

    else:

        menu()

def untuk\_edit\_produk(*id\_menu*):

    produk.edit(id\_menu)

def untuk\_edit\_karyawan(*id\_karyawan*):

    karyawan.edit(id\_karyawan)

def untuk\_hapus\_produk(*id\_menu*):

    produk.hapus(id\_menu)

def main():

    menu()

main()