**PROJECT DATABASE**

Laporan ini disusun untuk melengkapi Tugas Project Database

Mata Kuliah Pemrograman Berorientasi Objek (VI)



**Dosen Pengampu :**

Rodhiyah Mardhiyyah, S.Kom., M.Kom.

**Disusun oleh :**

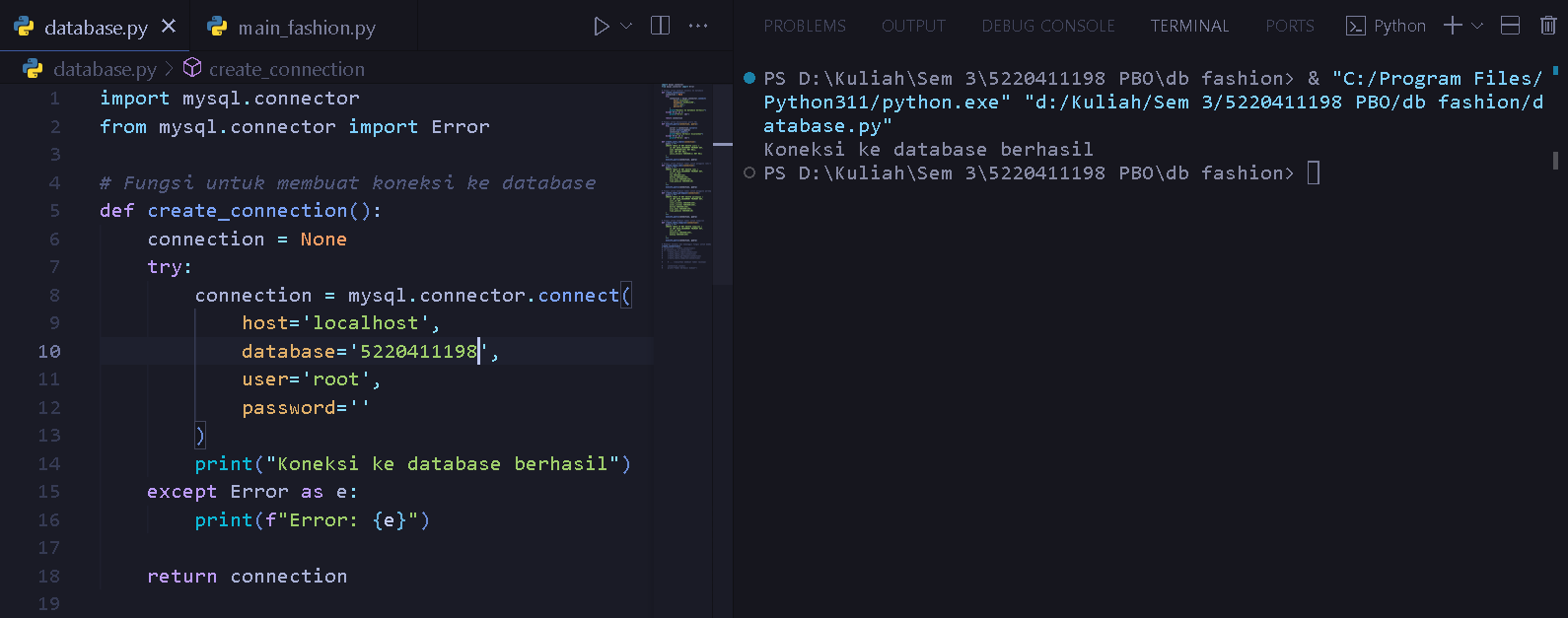
Raka Esa Rasendriya (5220411198)

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

Jl. Siliwangi Jl. Ring Road Utara, Jombor Lor, Sendangadi,

Kec. Mlati, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55285

1. **Database**
2. **Koneksi Database**

****

Memanggil fungsi create\_connection() untuk koneksi ke database.

1. **Membuat Tabel dengan Query**

Source Code & run :

import mysql.connector

from mysql.connector import Error

def create\_connection():

    connection = None

    try:

        connection = mysql.connector.connect(

            host='localhost',

            database='5220411198',

            user='root',

            password=''

        )

        print("Koneksi ke database berhasil")

    except Error as e:

        print(f"Error: {e}")

    return connection

def execute\_query(connection, query):

    try:

        cursor = connection.cursor()

        cursor.execute(query)

        connection.commit()

        print("Query berhasil dijalankan")

    except Error as e:

        print(f"Error: {e}")

def create\_users\_table(connection):

    query = """

    CREATE TABLE IF NOT EXISTS users (

        id INT AUTO\_INCREMENT PRIMARY KEY,

        nama VARCHAR(255) NOT NULL,

        umur INT NOT NULL,

        jenis\_kelamin VARCHAR(1) NOT NULL

    )

    """

    execute\_query(connection, query)

def create\_table\_laki(connection):

    query = """

    CREATE TABLE IF NOT EXISTS laki (

        id INT AUTO\_INCREMENT PRIMARY KEY,

        user\_id INT,

        top VARCHAR(255),

        bottom VARCHAR(255),

        alas\_kaki VARCHAR(255),

        tipe\_pakaian VARCHAR(10)

    )

    """

    execute\_query(connection, query)

def create\_table\_perempuan(connection):

    query = """

    CREATE TABLE IF NOT EXISTS perempuan (

        id INT AUTO\_INCREMENT PRIMARY KEY,

        user\_id INT,

        inner\_clothes VARCHAR(255),

        outer\_clothes VARCHAR(255),

        bottom VARCHAR(255),

        alas\_kaki VARCHAR(255),

        tipe\_pakaian VARCHAR(10)

    )

    """

    execute\_query(connection, query)

def create\_table\_tampilan(connection):

    query = """

    CREATE TABLE IF NOT EXISTS tampilan (

        id INT AUTO\_INCREMENT PRIMARY KEY,

        user\_id INT,

        aksesoris VARCHAR(255),

        makeup VARCHAR(255)

    )

    """

    execute\_query(connection, query)

create\_connection()

connection = create\_connection()

if connection.is\_connected():

    create\_users\_table(connection)

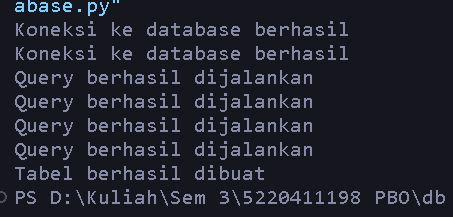
    create\_table\_laki(connection)

    create\_table\_perempuan(connection)

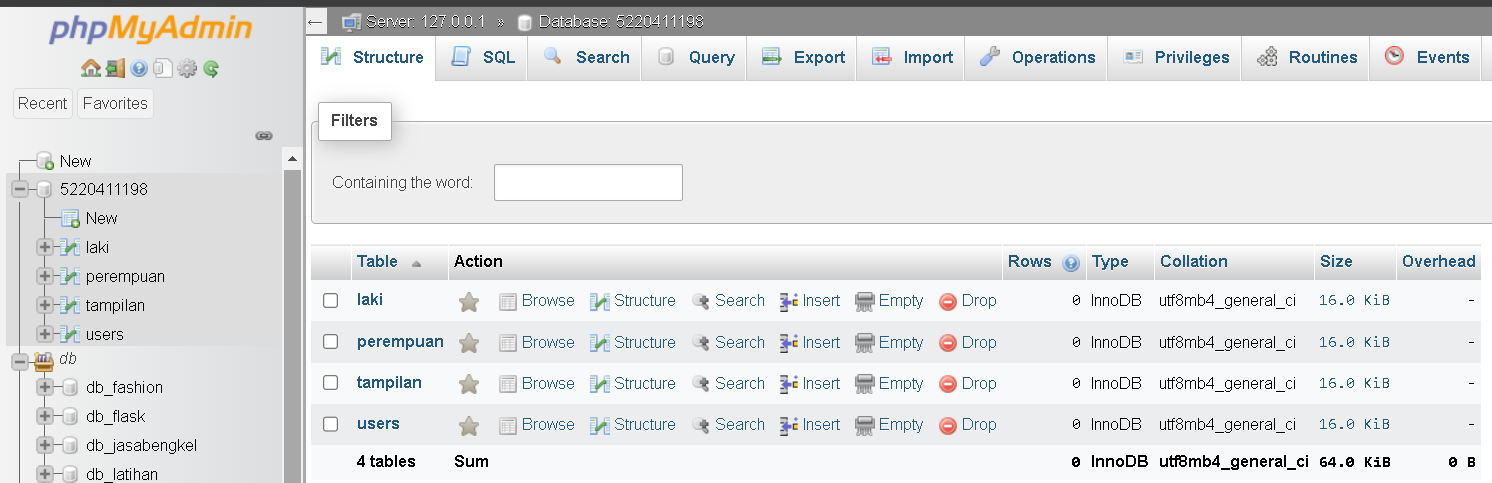
    create\_table\_tampilan(connection)

    connection.close()

    print("Tabel berhasil dibuat")



Hasil :



1. **Kode Program**

**Source Code Program Utama :**

import mysql.connector

from mysql.connector import Error

import os

def create\_connection():

    connection = None

    try:

        connection = mysql.connector.connect(

            host='localhost',

            database='5220411198',

            user='root',

            password=''

        )

        print("Koneksi ke database berhasil")

    except Error as e:

        print(f"Error: {e}")

    return connection

def execute\_query(connection, query):

    try:

        cursor = connection.cursor()

        cursor.execute(query)

        connection.commit()

        print("Query berhasil dijalankan")

    except Error as e:

        print(f"Error: {e}")

class Pakaian:

    def \_\_init\_\_(self, nama, umur, jenis\_kelamin):

        self.nama = nama

        self.\_umur = umur

        self.\_\_jenis\_kelamin = jenis\_kelamin

    def display\_info(self):

        print(f"\nInformasi Pengguna:")

        print(f"Nama            : {self.nama}")

        print(f"Umur            : {self.\_umur} tahun")

        print(f"Jenis Kelamin   : {self.\_\_jenis\_kelamin}")

class Laki(Pakaian):

    casual = {

        'atasan': ['T-shirt ', 'Sweater', 'Hoodie'],

        'bawahan': ['Jeans', 'Chino shorts', 'Joggers'],

        'alas\_kaki': ['Shoes', 'Sneakers', 'Sandals']

    }

    formal = {

        'atasan': ['Suit Jacket', 'Tuxedo shirt', 'Polo shirt'],

        'bawahan': ['Suit trousers', 'Chinos', 'Tuxedo pants'],

        'alas\_kaki': ['Oxford shoes', 'Loafers', 'Brogues']

    }

    def \_\_init\_\_(self, nama, umur):

        super().\_\_init\_\_(nama, umur, "Laki-laki")

        self.top = None

        self.bottom = None

        self.alas\_kaki = None

        self.tipe\_pakaian = None

    def show\_menu(self, kategori, clothes):

        print(f"Pilihan {kategori}:")

        for idx, item in enumerate(clothes[kategori], start=1):

            print(f"{idx}. {item}")

        choice = int(input(f"Pilih nomor {kategori}: "))

        return clothes[kategori][choice - 1]

    def pilih\_pakaian(self, kategori, clothes):

        print(f"\nMemilih Pakaian untuk {self.nama} ({kategori}):")

        selected\_item = self.show\_menu(kategori, clothes)

        return selected\_item

    def set\_tipe\_pakaian(self, tipe):

        print(f"\nMemilih Tipe Pakaian {tipe} untuk {self.nama}:")

        if tipe == 'casual':

            self.top = self.pilih\_pakaian('atasan', self.casual)

            self.bottom = self.pilih\_pakaian('bawahan', self.casual)

            self.alas\_kaki = self.pilih\_pakaian('alas\_kaki', self.casual)

        elif tipe == 'formal':

            self.top = self.pilih\_pakaian('atasan', self.formal)

            self.bottom = self.pilih\_pakaian('bawahan', self.formal)

            self.alas\_kaki = self.pilih\_pakaian('alas\_kaki', self.formal)

        self.tipe\_pakaian = tipe

    def display\_pakaianL(self):

        super().display\_info()

        print(f"\nDetail Pakaian untuk {self.nama}:")

        print(f"Tipe Pakaian    : {self.tipe\_pakaian if self.tipe\_pakaian else 'Belum dipilih'}")

        print(f"Atasan          : {self.top if self.top else 'Belum dipilih'}")

        print(f"Bawahan         : {self.bottom if self.bottom else 'Belum dipilih'}")

        print(f"Alas Kaki       : {self.alas\_kaki if self.alas\_kaki else 'Belum dipilih'}")

class Perempuan(Pakaian):

    casual = {

        'dalam': ['Blouse', 'Sweater', 'T-Shirt'],

        'luar': ['Vest', 'Cardigan', 'Denim Jacket'],

        'bawahan': ['Skirt', 'Pants', 'Shorts'],

        'alas\_kaki': ['Shoes', 'Sneakers', 'Sandals']

    }

    formal = {

        'dalam': ['Dress shirt', 'Blouse', 'Kemeja Putih'],

        'luar': ['Long Cardigan', 'Blazer', 'Vest'],

        'bawahan': ['Dress pants', 'Pencil skirt', 'Trousers'],

        'alas\_kaki': ['Heels', 'Pumps', 'Slingbacks']

    }

    def \_\_init\_\_(self, nama, umur):

        super().\_\_init\_\_(nama, umur, "Perempuan")

        self.inner = None

        self.outer = None

        self.bottom = None

        self.alas\_kaki = None

        self.tipe\_pakaian = None

    def show\_menu(self, kategori, clothes):

        print(f"Pilihan {kategori}:")

        for idx, item in enumerate(clothes[kategori], start=1):

            print(f"{idx}. {item}")

        choice = int(input(f"Pilih nomor {kategori}: "))

        return clothes[kategori][choice - 1]

    def pilih\_pakaian(self, kategori, clothes):

        print(f"\nMemilih Pakaian untuk {self.nama} ({kategori}):")

        selected\_item = self.show\_menu(kategori, clothes)

        return selected\_item

    def set\_tipe\_pakaian(self, tipe):

        print(f"\nMemilih Tipe Pakaian {tipe} untuk {self.nama}:")

        if tipe == 'casual':

            self.inner = self.pilih\_pakaian('dalam', self.casual)

            self.outer = self.pilih\_pakaian('luar', self.casual)

            self.bottom = self.pilih\_pakaian('bawahan', self.casual)

            self.alas\_kaki = self.pilih\_pakaian('alas\_kaki', self.casual)

        elif tipe == 'formal':

            self.inner = self.pilih\_pakaian('dalam', self.formal)

            self.outer = self.pilih\_pakaian('luar', self.formal)

            self.bottom = self.pilih\_pakaian('bawahan', self.formal)

            self.alas\_kaki = self.pilih\_pakaian('alas\_kaki', self.formal)

        self.tipe\_pakaian = tipe

    def display\_pakaianP(self):

        super().display\_info()

        print(f"\nDetail Pakaian untuk {self.nama}:")

        print(f"Tipe Pakaian    : {self.tipe\_pakaian if self.tipe\_pakaian else 'Belum dipilih'}")

        print(f"Inner           : {self.inner if self.inner else 'Belum dipilih'}")

        print(f"Outer           : {self.outer if self.outer else 'Belum dipilih'}")

        print(f"Bawahan         : {self.bottom if self.bottom else 'Belum dipilih'}")

        print(f"Alas Kaki       : {self.alas\_kaki if self.alas\_kaki else 'Belum dipilih'}")

class Tampilan(Perempuan):

    pilihan = {

        'aksesoris': ['Anting', 'Kalung', 'Gelang', 'Cincin'],

        'makeup': ['Base', 'Eye', 'Lip', 'Cheek', 'Brow', 'Setting Spray', 'Body', 'Mineral', 'Skincare-Infused']

    }

    def \_\_init\_\_(self, nama, umur):

        super().\_\_init\_\_(nama, umur)

        self.aksesoris = []

        self.makeup = []

    def tambah\_aksesoris(self):

        selected\_tampilan = self.show\_menu('aksesoris', self.pilihan)

        if selected\_tampilan not in self.aksesoris:

            self.aksesoris.append(selected\_tampilan)

            print(f"Aksesoris {selected\_tampilan} dipilih.")

    def tambah\_makeup(self):

        selected\_tampilan = self.show\_menu('makeup', self.pilihan)

        if selected\_tampilan not in self.makeup:

            self.makeup.append(selected\_tampilan)

            print(f"Make Up {selected\_tampilan} dipilih.")

    def display\_tampilan(self):

        super().display\_pakaianP()

        print(f"Make Up         : {', '.join(self.makeup) if self.makeup else 'Belum dipilih'}")

        print(f"Aksesoris       : {', '.join(self.aksesoris) if self.aksesoris else 'Belum dipilih'}")

*#Penerapan CRUD, INSERT, SELECT dan lainnya yang dibutuhkan untuk mengelola database*

def list\_pengguna(connection):

    connection = create\_connection()

    if connection.is\_connected():

        try:

            cursor = connection.cursor(dictionary=True)

            query = "SELECT \* FROM users"

            cursor.execute(query)

            users = cursor.fetchall()

            while True:

                print("\n==============================")

                print("\nList Pengguna:")

                for idx, user in enumerate(users, start=1):

                    print(f"{idx}. {user['nama']} ({user['jenis\_kelamin']})")

                choice\_detail = input("\nPilih nomor pengguna untuk melihat atau tambah detail (0 untuk kembali): ")

                if choice\_detail.isdigit():

                    choice\_detail = int(choice\_detail)

                    if 0 < choice\_detail <= len(users):

                        pengguna\_terpilih = users[choice\_detail - 1]

                        while True:

                            os.system("cls")

                            print("\n==============================")

                            print(f"Detail Pengguna {pengguna\_terpilih['nama']} ({pengguna\_terpilih['jenis\_kelamin']}):")

                            print("1. Tampilkan Pakaian")

                            print("2. Tambah Pakaian")

                            print("3. Hapus Pengguna")

                            print("4. Update Data Pengguna")

                            print("0. Kembali")

                            choice\_detail\_menu = input("\nPilih Menu : ")

                            if choice\_detail\_menu == "1":

                                display\_pakaian\_pengguna(connection, pengguna\_terpilih)

                            elif choice\_detail\_menu == "2":

                                tambah\_pakaian\_pengguna(connection, pengguna\_terpilih)

                            elif choice\_detail\_menu == "3":

                                hapus\_pengguna(connection, pengguna\_terpilih)

                            elif choice\_detail\_menu == "4":

                                update\_data\_pengguna(connection, pengguna\_terpilih)

                            elif choice\_detail\_menu == "0":

                                break

                            else:

                                print("Pilihan tidak valid. Silakan pilih antara 0-4.")

                            os.system("pause")

                    elif choice\_detail == 0:

                        break

                    else:

                        print("Nomor pengguna tidak valid.")

                else:

                    print("Masukkan angka yang valid.")

                os.system("pause")

        except Error as e:

            print(f"Error: {e}")

        finally:

            cursor.close()

            connection.close()

    else:

        print("Tidak dapat terhubung ke database.")

def display\_pakaian\_pengguna(connection, pengguna\_terpilih):

    cursor = connection.cursor(dictionary=True)

    if pengguna\_terpilih['jenis\_kelamin'] == 'L':

        pengguna = Laki(pengguna\_terpilih['nama'], pengguna\_terpilih['umur'])

        query\_pakaian = f"SELECT \* FROM laki WHERE user\_id = {pengguna\_terpilih['id']}"

        cursor.execute(query\_pakaian)

        pakaian = cursor.fetchone()

        if pakaian:

            pengguna.top = pakaian['top']

            pengguna.bottom = pakaian['bottom']

            pengguna.alas\_kaki = pakaian['alas\_kaki']

            pengguna.tipe\_pakaian = pakaian['tipe\_pakaian']

        pengguna.display\_pakaianL()

    elif pengguna\_terpilih['jenis\_kelamin'] == 'P':

        pengguna = Tampilan(pengguna\_terpilih['nama'], pengguna\_terpilih['umur'])

        query\_pakaian = f"SELECT \* FROM perempuan WHERE user\_id = {pengguna\_terpilih['id']}"

        cursor.execute(query\_pakaian)

        pakaian = cursor.fetchone()

        if pakaian:

            pengguna.inner = pakaian['inner\_clothes']

            pengguna.outer = pakaian['outer\_clothes']

            pengguna.bottom = pakaian['bottom']

            pengguna.alas\_kaki = pakaian['alas\_kaki']

            pengguna.tipe\_pakaian = pakaian['tipe\_pakaian']

        query\_tampilan = f"SELECT \* FROM tampilan WHERE user\_id = {pengguna\_terpilih['id']}"

        cursor.execute(query\_tampilan)

        tampilan = cursor.fetchone()

        if tampilan:

            pengguna.aksesoris = tampilan['aksesoris'].split(', ') if tampilan['aksesoris'] else []

            pengguna.makeup = tampilan['makeup'].split(', ') if tampilan['makeup'] else []

        pengguna.display\_tampilan()

def hapus\_pengguna(connection, pengguna\_terpilih):

    user\_id = pengguna\_terpilih['id']

    jenis\_kelamin = pengguna\_terpilih['jenis\_kelamin']

    confirm\_delete = input(f"Apakah Anda yakin ingin menghapus pengguna {pengguna\_terpilih['nama']}? (Y/N): ")

    if confirm\_delete.upper() == 'Y':

        try:

            cursor = connection.cursor()

            if jenis\_kelamin == 'L':

                query\_delete\_pakaian = f"DELETE FROM laki WHERE user\_id = {user\_id}"

                execute\_query(connection, query\_delete\_pakaian)

            elif jenis\_kelamin == 'P':

                query\_delete\_pakaian\_perempuan = f"DELETE FROM perempuan WHERE user\_id = {user\_id}"

                query\_delete\_tampilan = f"DELETE FROM tampilan WHERE user\_id = {user\_id}"

                execute\_query(connection, query\_delete\_pakaian\_perempuan)

                execute\_query(connection, query\_delete\_tampilan)

            query\_delete\_user = f"DELETE FROM users WHERE id = {user\_id}"

            execute\_query(connection, query\_delete\_user)

            print(f"Pengguna dengan ID {user\_id} berhasil dihapus.")

            os.system("pause")

        except Error as e:

            print(f"Error: {e}")

        finally:

            cursor.close()

            tampil()

    else:

        print(f"Pengguna dengan ID {user\_id} tidak dihapus.")

def update\_data\_pengguna(connection, pengguna\_terpilih):

    user\_id = pengguna\_terpilih['id']

    try:

        cursor = connection.cursor(dictionary=True)

        query\_get\_user = f"SELECT \* FROM users WHERE id = {user\_id}"

        cursor.execute(query\_get\_user)

        user\_data = cursor.fetchone()

        if user\_data:

            print(f"\nData Pengguna yang akan diupdate:")

            print(f"Nama          : {user\_data['nama']}")

            print(f"Umur          : {user\_data['umur']}")

            print(f"Jenis Kelamin : {user\_data['jenis\_kelamin']}")

            baru\_nama = input("\nMasukkan Nama Baru: ")

            baru\_umur = int(input("Masukkan Umur Baru: "))

            baru\_jk = input("Masukkan Jenis Kelamin Baru (L/P): ")

            if baru\_jk.upper() in ['L', 'P']:

                query\_update\_user = f"UPDATE users SET nama = '{baru\_nama}', umur = {baru\_umur}, jenis\_kelamin = '{baru\_jk}' WHERE id = {user\_id}"

                execute\_query(connection, query\_update\_user)

                print("Data pengguna berhasil diupdate.")

                os.system("pause")

                tampil()

            else:

                print("Jenis kelamin tidak valid.")

                os.system("pause")

        else:

            print(f"Pengguna dengan ID {user\_id} tidak ditemukan.")

            os.system("pause")

    except Error as e:

        print(f"Error: {e}")

    finally:

        cursor.close()

    os.system("pause")

def tambah\_pakaian\_pengguna(connection, pengguna\_terpilih):

    cursor = connection.cursor()

    if pengguna\_terpilih['jenis\_kelamin'] == 'L':

        pengguna = Laki(pengguna\_terpilih['nama'], pengguna\_terpilih['umur'])

        query\_pakaian = f"SELECT \* FROM laki WHERE user\_id = {pengguna\_terpilih['id']}"

        cursor.execute(query\_pakaian)

        pakaian = cursor.fetchone()

        if pakaian:

            pengguna.top = pakaian[1]

            pengguna.bottom = pakaian[2]

            pengguna.alas\_kaki = pakaian[3]

            pengguna.tipe\_pakaian = pakaian[4]

        pengguna.display\_pakaianL()

        while True:

            os.system("cls")

            print("\n==============================")

            print("\nTambah Pakaian:")

            print("1. Casual")

            print("2. Formal")

            print("0. Kembali")

            choice\_tambah\_pakaian = input("\nPilih Tipe Pakaian (0-2): ")

            if choice\_tambah\_pakaian == "1":

                pengguna.set\_tipe\_pakaian('casual')

                update\_query = f"""

                UPDATE laki

                SET top = '{pengguna.top}', bottom = '{pengguna.bottom}', alas\_kaki = '{pengguna.alas\_kaki}', tipe\_pakaian = 'casual'

                WHERE user\_id = {pengguna\_terpilih['id']}

                """

                execute\_query(connection, update\_query)

                print("Pakaian casual berhasil ditambahkan.")

            elif choice\_tambah\_pakaian == "2":

                pengguna.set\_tipe\_pakaian('formal')

                update\_query = f"""

                UPDATE laki

                SET top = '{pengguna.top}', bottom = '{pengguna.bottom}', alas\_kaki = '{pengguna.alas\_kaki}', tipe\_pakaian = 'formal'

                WHERE user\_id = {pengguna\_terpilih['id']}

                """

                execute\_query(connection, update\_query)

                print("Pakaian formal berhasil ditambahkan.")

            elif choice\_tambah\_pakaian == "0":

                break

            else:

                print("Pilihan tidak valid. Silakan pilih antara 0-2.")

            os.system("pause")

    elif pengguna\_terpilih['jenis\_kelamin'] == 'P':

        pengguna = Tampilan(pengguna\_terpilih['nama'], pengguna\_terpilih['umur'])

        query\_pakaian = f"SELECT \* FROM perempuan WHERE user\_id = {pengguna\_terpilih['id']}"

        cursor.execute(query\_pakaian)

        pakaian = cursor.fetchone()

        if pakaian:

            pengguna.inner = pakaian[1]

            pengguna.outer = pakaian[2]

            pengguna.bottom = pakaian[3]

            pengguna.alas\_kaki = pakaian[4]

            pengguna.tipe\_pakaian = pakaian[5]

        pengguna.display\_tampilan()

        while True:

            os.system("cls")

            print("\n==============================")

            print("\nTambah Pakaian:")

            print("1. Casual")

            print("2. Formal")

            print("3. Tambah Aksesoris")

            print("4. Tambah MakeUp")

            print("0. Kembali")

            choice\_tambah\_pakaian = input("\nPilih Tipe Pakaian (0-3): ")

            if choice\_tambah\_pakaian == "1":

                pengguna.set\_tipe\_pakaian('casual')

                update\_query = f"""

                UPDATE perempuan

                SET inner\_clothes = '{pengguna.inner}', outer\_clothes = '{pengguna.outer}',

                bottom = '{pengguna.bottom}', alas\_kaki = '{pengguna.alas\_kaki}', tipe\_pakaian = 'casual'

                WHERE user\_id = {pengguna\_terpilih['id']}

                """

                execute\_query(connection, update\_query)

                print("Pakaian casual berhasil ditambahkan.")

            elif choice\_tambah\_pakaian == "2":

                pengguna.set\_tipe\_pakaian('formal')

                update\_query = f"""

                UPDATE perempuan

                SET inner\_clothes = '{pengguna.inner}', outer\_clothes = '{pengguna.outer}',

                bottom = '{pengguna.bottom}', alas\_kaki = '{pengguna.alas\_kaki}', tipe\_pakaian = 'formal'

                WHERE user\_id = {pengguna\_terpilih['id']}

                """

                execute\_query(connection, update\_query)

                print("Pakaian formal berhasil ditambahkan.")

            elif choice\_tambah\_pakaian == "3":

                pengguna.tambah\_aksesoris()

                update\_query = f"""

                UPDATE tampilan

                SET aksesoris = '{', '.join(pengguna.aksesoris)}'

                WHERE user\_id = {pengguna\_terpilih['id']}

                """

                execute\_query(connection, update\_query)

                print("Aksesoris berhasil ditambahkan.")

            elif choice\_tambah\_pakaian == "4":

                pengguna.tambah\_makeup()

                update\_query = f"""

                UPDATE tampilan

                SET makeup = '{', '.join(pengguna.makeup)}'

                WHERE user\_id = {pengguna\_terpilih['id']}

                """

                execute\_query(connection, update\_query)

                print("MakeUp berhasil ditambahkan.")

            elif choice\_tambah\_pakaian == "0":

                break

            else:

                print("Pilihan tidak valid. Silakan pilih antara 0-3.")

            os.system("pause")

    else:

        print("Jenis kelamin tidak valid.")

*#Main Program*

def tampil():

    connection = create\_connection()

    while True:

        os.system("cls")

        print("\n==============================")

        print("Menu Utama :")

        print("1. Tambah Pengguna")

        print("2. List Pengguna")

        print("3. Keluar")

        choice\_menu = input("\nPilih Menu : ")

        if choice\_menu == "1":

            while True:

                nama = input("\nMasukkan Nama Pengguna: ")

                umur = int(input("Masukkan Umur Pengguna: "))

                jenis\_kelamin = input("Masukkan Jenis Kelamin Pengguna (L/P): ")

                if jenis\_kelamin.upper() in ['L', 'P']:

                    break

                else:

                    print("Jenis kelamin tidak valid. Silahkan isi ulang data diri Anda.")

                os.system("pause")

            if jenis\_kelamin.upper() == 'L':

                pengguna = Laki(nama, umur)

                query\_insert = f"INSERT INTO users (nama, umur, jenis\_kelamin) VALUES ('{nama}', {umur}, 'L')"

                execute\_query(connection, query\_insert)

                query\_get\_id = f"SELECT id FROM users WHERE nama = '{nama}'"

                cursor = connection.cursor()

                cursor.execute(query\_get\_id)

                user\_id = cursor.fetchone()[0]

                query\_insert\_laki = f"INSERT INTO laki (user\_id, top, bottom, alas\_kaki, tipe\_pakaian) VALUES ({user\_id}, '', '', '', '')"

                execute\_query(connection, query\_insert\_laki)

            elif jenis\_kelamin.upper() == 'P':

                pengguna = Tampilan(nama, umur)

                query\_insert = f"INSERT INTO users (nama, umur, jenis\_kelamin) VALUES ('{nama}', {umur}, 'P')"

                execute\_query(connection, query\_insert)

                query\_get\_id = f"SELECT id FROM users WHERE nama = '{nama}'"

                cursor = connection.cursor()

                cursor.execute(query\_get\_id)

                user\_id = cursor.fetchone()[0]

                query\_insert\_perempuan = f"INSERT INTO perempuan (user\_id, inner\_clothes, outer\_clothes, bottom, alas\_kaki, tipe\_pakaian) VALUES ({user\_id}, '', '', '', '', '')"

                execute\_query(connection, query\_insert\_perempuan)

                query\_insert\_tampilan = f"INSERT INTO tampilan (user\_id, aksesoris, makeup) VALUES ({user\_id}, '', '')"

                execute\_query(connection, query\_insert\_tampilan)

        elif choice\_menu == "2":

            list\_pengguna(connection)

        elif choice\_menu == "3":

            print("Program selesai. Sampai jumpa!")

            break

        else:

            print("Pilihan tidak valid. Silakan pilih antara 1-4.")

        os.system("pause")

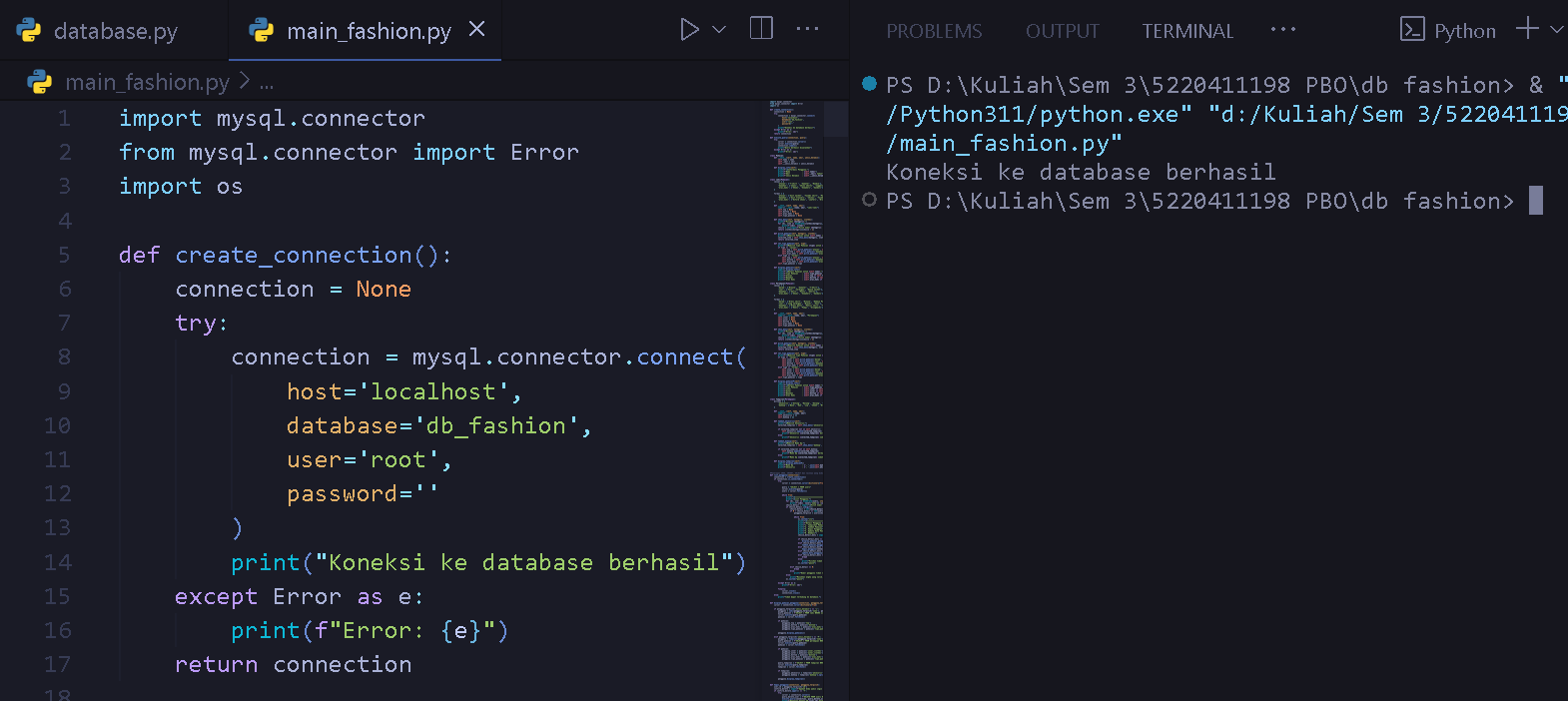
if \_\_name\_\_ == "\_\_main\_\_":

*# create\_connection()*

    tampil()

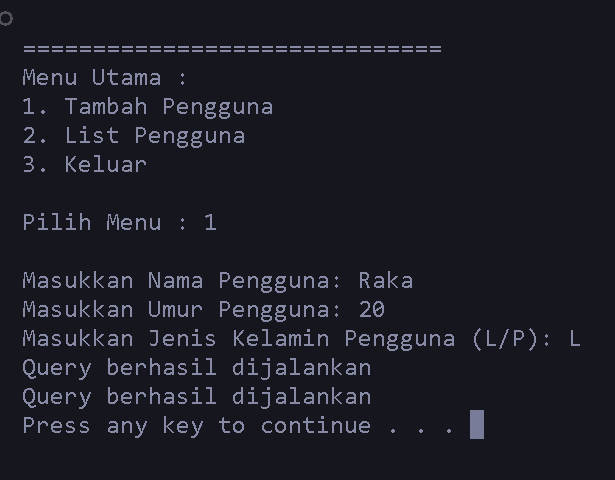
1. **Hasil Running**
2. **Create**

Membuat koneksi dengan Database :

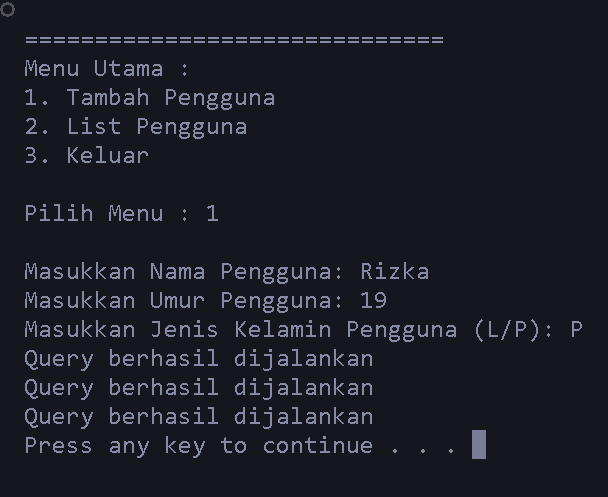


1. **Insert**

Mengisi data tabel user dan mengisi data pakaiannya dengan kekosongan (sementara) :

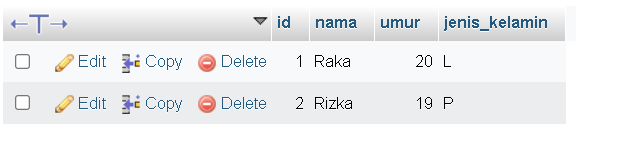


Terdapat dua query yang berhasil dijalankan karena query pertama memasukkan data user sedangkan query kedua memasukkan data pakaian laki—laki yang sementara dikosongkan.

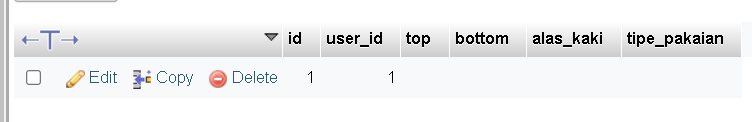


Sama halnya seperti data laki—laki, namun pada data perempuan terdapat query tambahan untuk mengisi data tampilan (aksesoris dan makeup) yang kosong sementara.

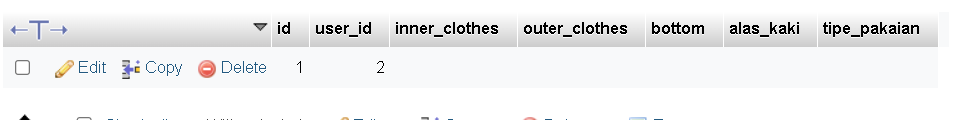
Tabel user :



Tabel laki :



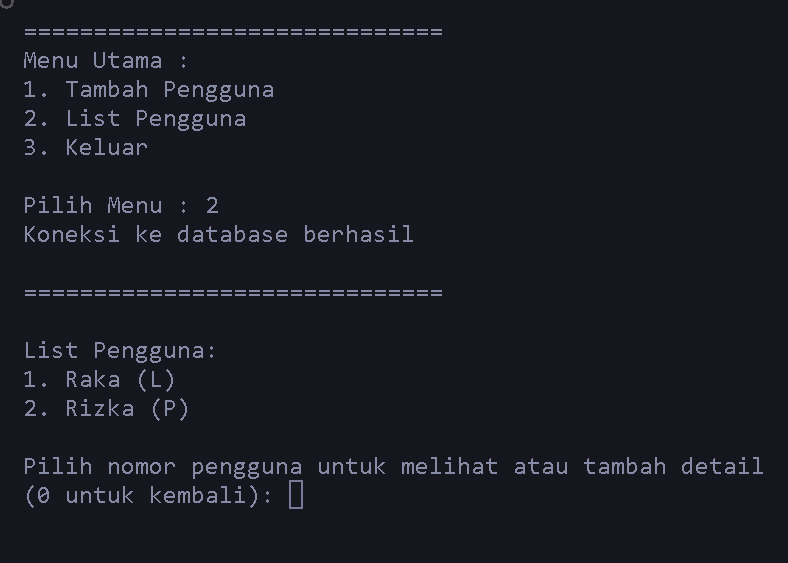
Tabel perempuan :



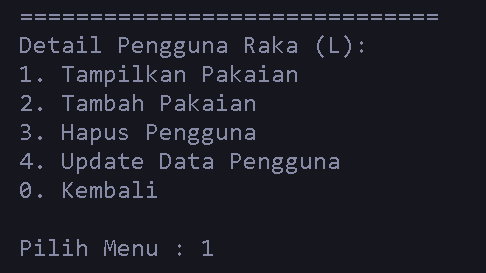
Tabel tampilan (perempuan) :

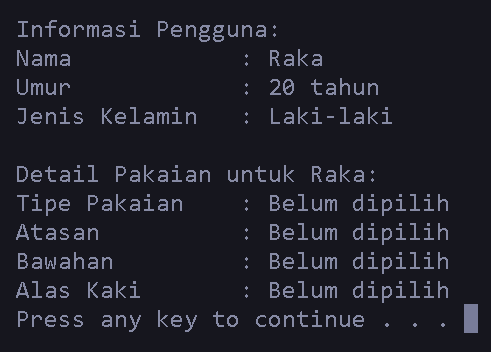


1. **Read & Select**
2. Membaca/menampilkan isi tabel user

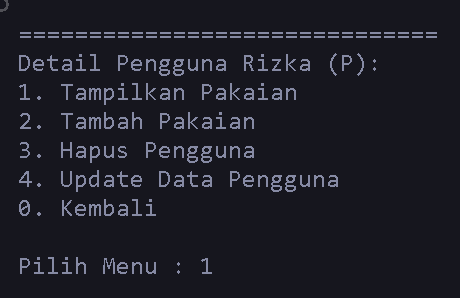


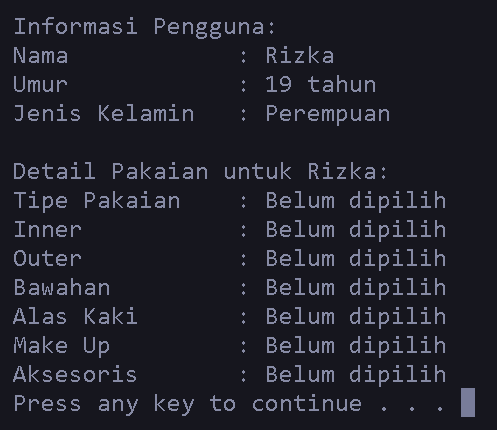
1. Membaca/menampilkan isi tabel laki ketika memilih user laki—laki



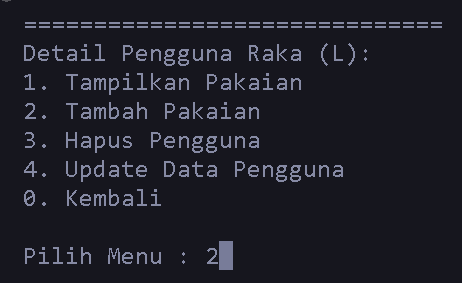


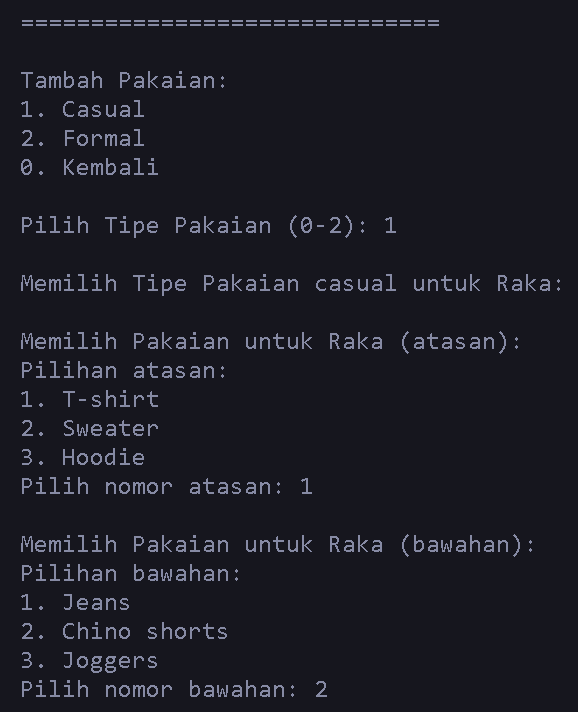
1. Membaca/menampilkan isi tabel perempuan dan tampilan ketika memilih user perempuan

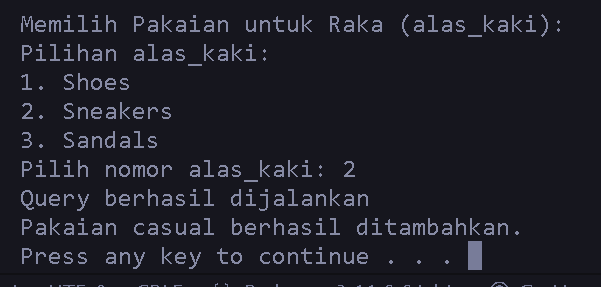


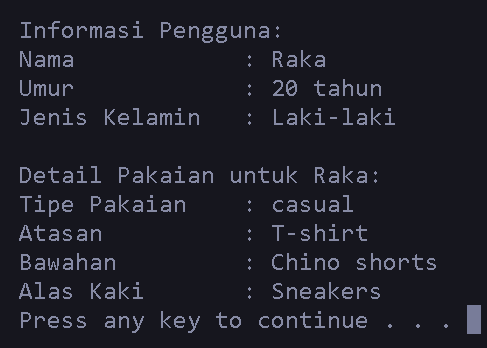


1. **Update**
2. Melakukan pembaruan pada data tabel pakaian laki—laki



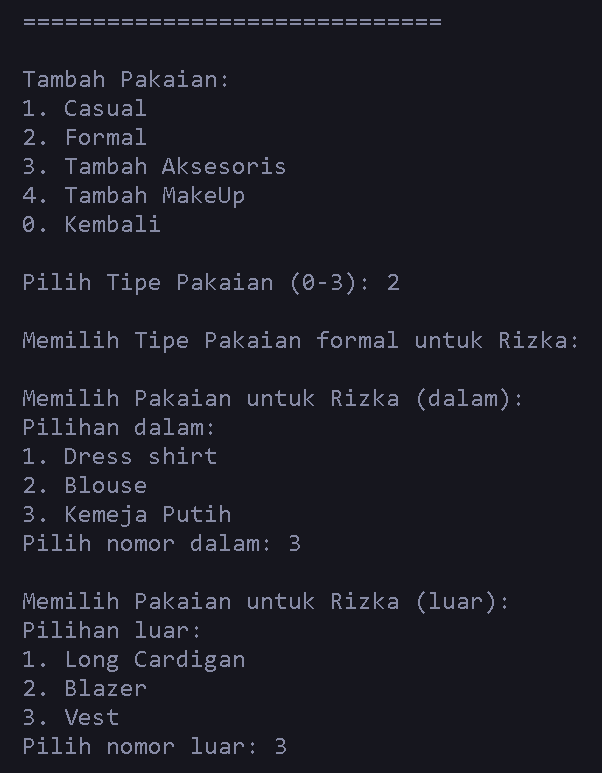


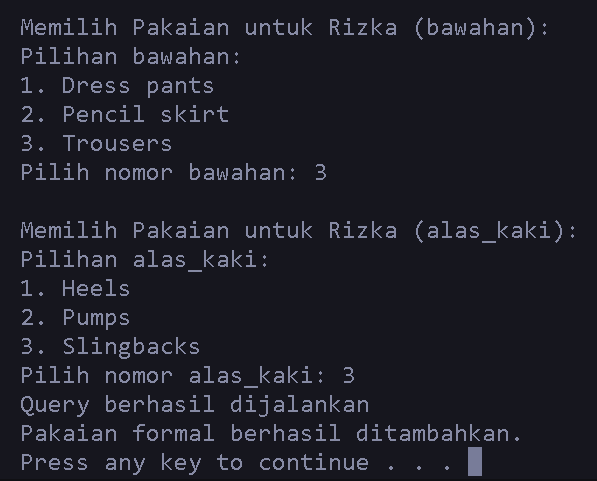






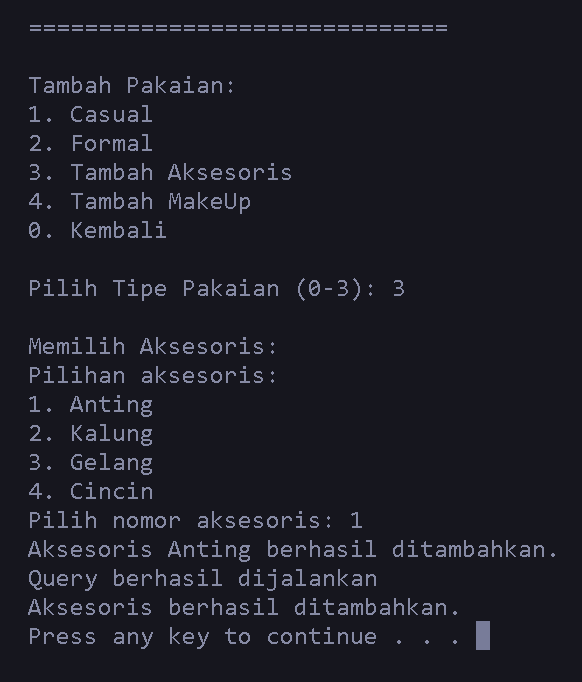
1. Melakukan pembaruan pada data tabel pakaian perempuan

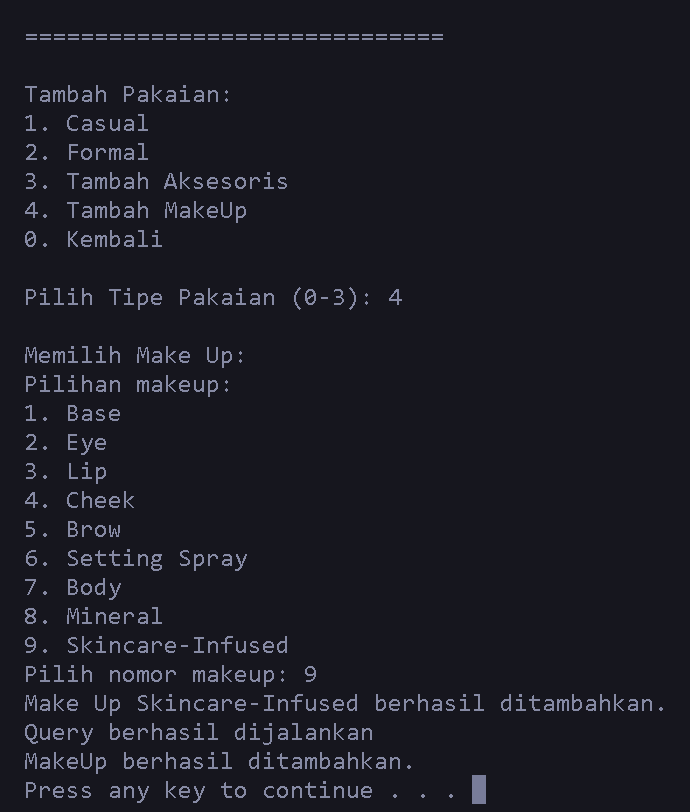


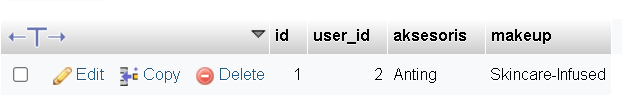




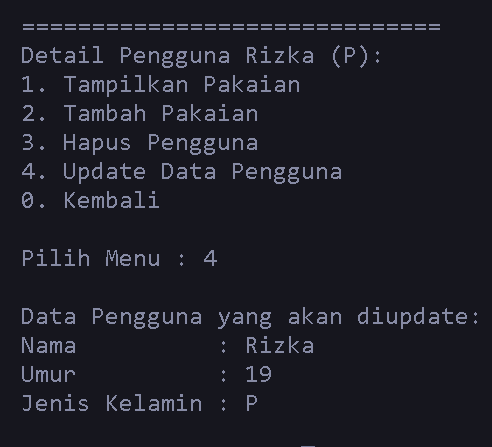
1. Melakukan pembaruan pada data tabel tampilan (aksesoris dan makeup) perempuan

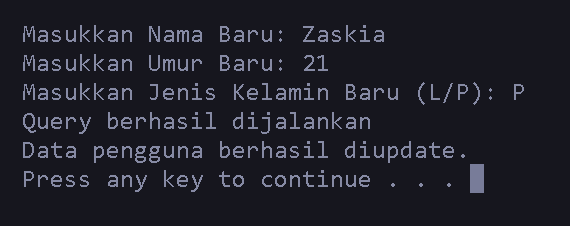






1. Melakukan pembaruan pada data salah satu user yang ada pada tabel user



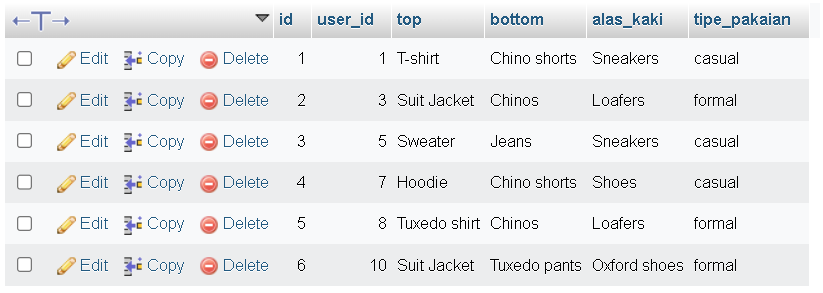




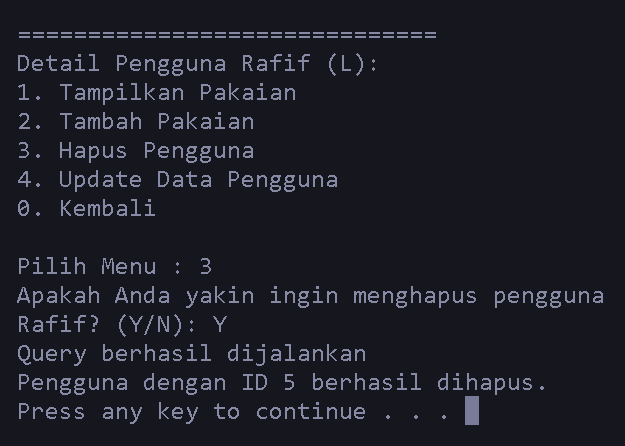
1. **Delete**
2. **Menghapus data user laki—laki**

* Data awal pada tabel user dan laki :

****

****

* Menghapus salah satu user laki—laki :

****

* Data tabel setelah menghapus salah satu user :



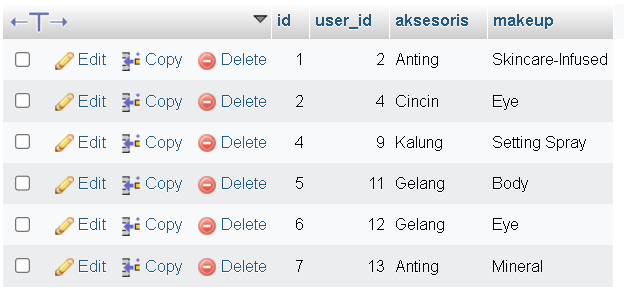
****

1. **Menghapus data user perempuan :**

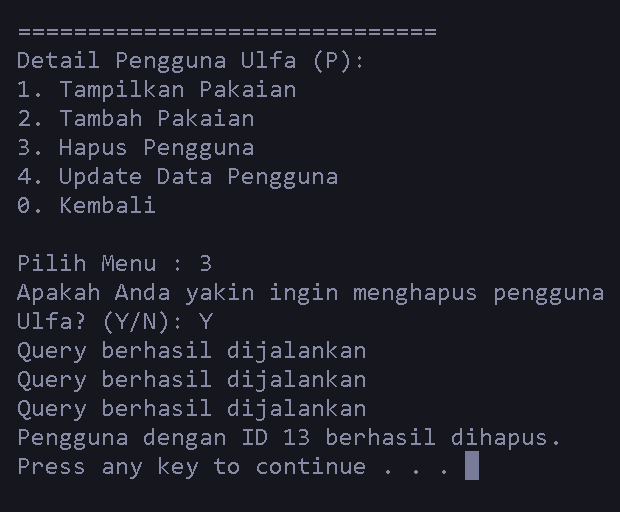
* Data awal pada tabel user, perempuan dan tampilan :

****

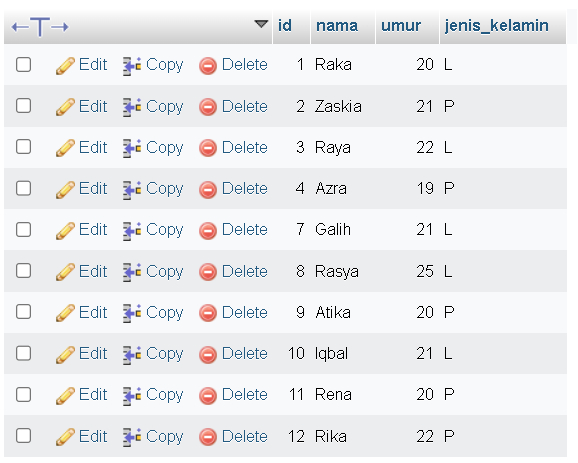
****

****

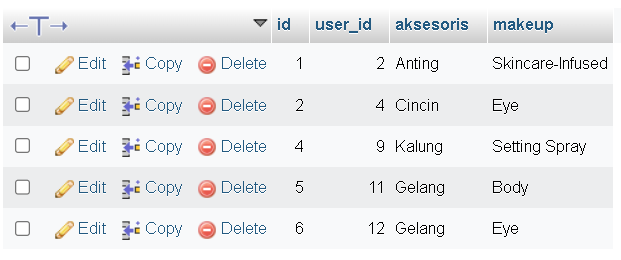
* Menghapus salah satu user perempuan :

****

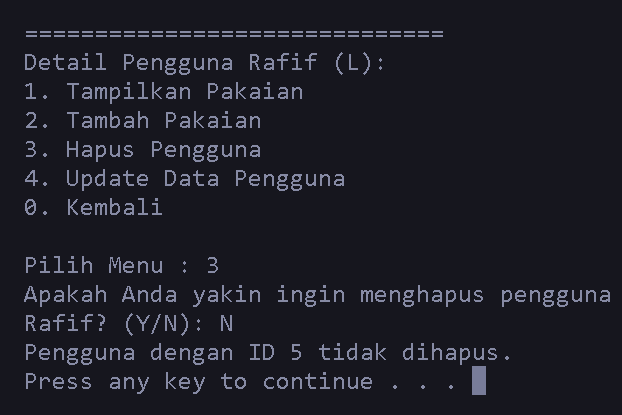
* Data tabel setelah menghapus salah satu user :

****

****

****

1. **Contoh jika tidak jadi menghapus :**

****

1. **Link Program**

* **Google Drive**

<https://drive.google.com/drive/folders/1y3lor9gsE_VO64kflDvnPGBr8jcOzXD8?usp=sharing>

Alternatif :

<https://drive.google.com/file/d/1yC-C2bt7nx2t9A8lIDEqnCK-kmvt-KjG/view?usp=sharing>

* **Github**

<https://github.com/RakaRasendriya/PBOVI_5220411198_Raka-Esa-Rasendriya_Project-Database>