

LAPORAN TUGAS PERTEMUAN 10

T. A. Semester Genap 2023/2024



Nama : Alfina Dwi Rachmawati (5230411288)

Kelas : VIII

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024**

I. Tujuan

Program ini dirancang untuk mengelola data transaksi penjualan, yang mencakup pengelolaan pegawai, produk, transaksi, dan detail transaksi. Pengguna dapat menambahkan pegawai, produk, melakukan transaksi, dan mencetak struk transaksi yang dilakukan.

II. Kode Program

A. Koneksi ke Database

```
import mysql.connector

conn = mysql.connector.connect (
    host="localhost",
    user="root",
    password="",
    database="penjualan"
)
cursor = conn.cursor()
```

Pada bagian ini, program melakukan koneksi ke database MySQL yang berada di localhost menggunakan modul mysql.connector. Variabel conn menyimpan koneksi ke database, sementara cursor digunakan untuk menjalankan query SQL.

B. Membuat Tabel

```
cursor.execute("""
CREATE TABLE IF NOT EXISTS Pegawai (
    nik VARCHAR(20) PRIMARY KEY,
    nama VARCHAR(255),
    alamat TEXT
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS Produk (
    kode_produk INT PRIMARY KEY AUTO_INCREMENT,
    nama_produk VARCHAR(255),
    jenis_produk VARCHAR(50),
    harga DECIMAL(10, 2)
)
""")
```

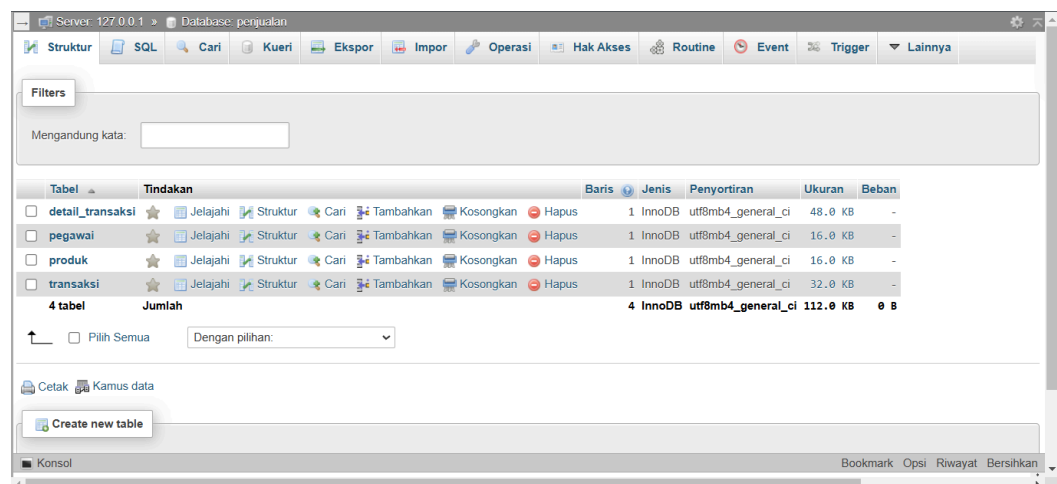
```

cursor.execute("""
CREATE TABLE IF NOT EXISTS Transaksi (
    no_transaksi INT PRIMARY KEY AUTO_INCREMENT,
    nik VARCHAR(20),
    tanggal DATE,
    FOREIGN KEY (nik) REFERENCES Pegawai(nik)
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS Detail_Transaksi (
    id_detail INT PRIMARY KEY AUTO_INCREMENT,
    no_transaksi INT,
    kode_produk INT,
    jumlah INT,
    total_harga DECIMAL(10, 2),
    FOREIGN KEY (no_transaksi) REFERENCES
Transaksi(no_transaksi),
    FOREIGN KEY (kode_produk) REFERENCES Produk(kode_produk)
)
""")

```

- **Pegawai:** Menyimpan data pegawai dengan kolom nik, nama, dan alamat.
- **Produk:** Menyimpan data produk yang terdiri dari kode_produk, nama_produk, jenis_produk, dan harga.
- **Transaksi:** Menyimpan data transaksi yang mencatat nomor transaksi dan pegawai yang melakukan transaksi.
- **Detail_Transaksi:** Menyimpan detail produk yang dibeli dalam setiap transaksi, termasuk kode_produk, jumlah, dan total_harga.



C. Menu Utama

```
while True:
    print("\nMenu:")
    print("1. Tambah Pegawai")
    print("2. Tambah Produk")
    print("3. Lakukan Transaksi")
    print("4. Cetak Struk")
    print("5. Keluar")

    pilihan = input("Pilih menu: ")
```

Bagian ini adalah tampilan menu utama yang memberikan pilihan kepada pengguna untuk memilih aksi yang ingin dilakukan. Menu ini diulang dalam sebuah loop while hingga pengguna memilih untuk keluar.

D. Tambah Pegawai

```
if pilihan == "1":
    nik = input("Masukkan NIK Pegawai: ")
    nama = input("Masukkan Nama Pegawai: ")
    alamat = input("Masukkan Alamat Pegawai: ")
    cursor.execute("INSERT INTO Pegawai (nik, nama, alamat) VALUES (%s, %s, %s)", (nik, nama, alamat))
    conn.commit()
    print("Pegawai berhasil ditambahkan.")
```

Jika pengguna memilih untuk menambahkan pegawai, program akan meminta input untuk nik, nama, dan alamat pegawai. Data ini kemudian dimasukkan ke dalam tabel Pegawai menggunakan query INSERT. Setelah itu, perubahan disimpan dengan conn.commit().

Menambahkan Pegawai :

```
Menu:
1. Tambah Pegawai
2. Tambah Produk
3. Lakukan Transaksi
4. Cetak Struk
5. Keluar
Pilih menu: 1
Masukkan NIK Pegawai: 2222
Masukkan Nama Pegawai: seungmin
Masukkan Alamat Pegawai: jl. stay
Pegawai berhasil ditambahkan.
```

E. Tambah Produk

```
elif pilihan == "2":
    print("\nJenis Produk:")
```

```

print("1. Snack")
print("2. Makanan")
print("3. Minuman")
jenis_produk = input("Pilih jenis produk: ")
nama_produk = input("Masukkan Nama Produk: ")
harga = float(input("Masukkan Harga Produk: "))

if jenis_produk == "1":
    jenis = "Snack"
elif jenis_produk == "2":
    jenis = "Makanan"
elif jenis_produk == "3":
    jenis = "Minuman"
else:
    print("Jenis produk tidak valid.")
    continue

cursor.execute("INSERT INTO Produk (nama_produk,
jenis_produk, harga) VALUES (%s, %s, %s)", (nama_produk,
jenis, harga))

conn.commit()

print("Produk berhasil ditambahkan.")

```

Bagian ini memungkinkan pengguna untuk menambahkan produk ke dalam sistem. Pengguna memilih jenis produk (snack, makanan, atau minuman), kemudian memasukkan nama produk dan harga produk. Data ini akan dimasukkan ke dalam tabel Produk.

Menambahkan Produk :

```

Menu:
1. Tambah Pegawai
2. Tambah Produk
3. Lakukan Transaksi
4. Cetak Struk
5. Keluar
Pilih menu: 2

Jenis Produk:
1. Snack
2. Makanan
3. Minuman
Pilih jenis produk: 1
Masukkan Nama Produk: Sari Roti Coklat
Masukkan Harga Produk: 6000
Produk berhasil ditambahkan.

```

Tersimpan di Database :

Server: 127.0.0.1 » Database: perputakan » Tabel: produk

Jelajahi

Struktur

SQL

Cari

Tambahkan

Ekspor

Impor

Hak Akses

Operasi

Pelacakan

Trigger

☐ Profil

[Edit kotak]

[Ubah]

[Jelaskan SQL]

[Buat kode PHP]

[Segarkan]

☐ Tampilkan semua

Jumlah baris: 25

Saring baris: Cari di tabel ini

Sort by key: Tidak ada

Extra options

← →

▼ kode_produk

nama_produk

jenis_produk

harga

☐

Ubah

Salin

Hapus

1

lays

Snack

5000.00

☐

Ubah

Salin

Hapus

2

Sari Roti Coklat

Snack

6000.00

☐

Ubah

Salin

Hapus

3

Air Mineral

Minuman

8000.00

☐

Ubah

Salin

Hapus

4

Nasi Rendang

Makanan

30000.00

☐

Ubah

Salin

Hapus

5

Nasi Ayam

Makanan

25000.00

⬆

☐ Pilih Semua

Dengan pilihan:

Ubah

Salin

Hapus

Ekspor

☐ Tampilkan semua

Jumlah baris: 25

Saring baris: Cari di tabel ini

Sort by key: Tidak ada

Operasi hasil kueri

Konsol

Bookmark Opsi Riwayat Bersihkan

F. Transaksi

```
elif pilihan == "3":
    nik = input("Masukkan NIK Pegawai yang melakukan
transaksi: ")

    cursor.execute("INSERT INTO Transaksi (nik, tanggal)
VALUES (%s, CURDATE())", (nik,))
    conn.commit()
    no_transaksi = cursor.lastrowid

    while True:
        print("\nPilih Produk (Ketik 's' untuk
selesai):")

        cursor.execute("SELECT * FROM Produk")
        produk_list = cursor.fetchall()
        for p in produk_list:
            print(f"{p[0]}. {p[1]} - {p[2]} - Rp{p[3]}")

        produk_input = input("Pilih kode produk atau 's'
untuk selesai: ")

        if produk_input.lower() == 's':
            break

        try:
            kode_produk = int(produk_input)
            jumlah = int(input("Masukkan Jumlah Produk:
"))

            cursor.execute("SELECT harga FROM Produk
WHERE kode_produk = %s", (kode_produk,))
            harga = cursor.fetchone()[0]
            total_harga = harga * jumlah
```

```

        cursor.execute("INSERT INTO Detail_Transaksi
(no_transaksi, kode_produk, jumlah, total_harga) VALUES (%s,
%s, %s, %s)", (no_transaksi, kode_produk, jumlah,
total_harga))

        conn.commit()

        print(f"Produk dengan kode {kode_produk}
sebanyak {jumlah} berhasil ditambahkan ke transaksi.")

    except (ValueError, TypeError):
        print("Input tidak valid.")

```

Untuk melakukan transaksi, pengguna memasukkan nik pegawai yang melakukan transaksi. Program kemudian membuat transaksi baru dan mencatat tanggal transaksi menggunakan CURDATE(). Nomor transaksi yang baru dibuat disimpan dalam no_transaksi.

```

Menu:
1. Tambah Pegawai
2. Tambah Produk
3. Lakukan Transaksi
4. Cetak Struk
5. Keluar
Pilih menu: 3
Masukkan NIK Pegawai yang melakukan transaksi: 2222

Pilih Produk (Ketik 's' untuk selesai):
1. lays - Snack - Rp5000.00
2. Sari Roti Coklat - Snack - Rp6000.00
3. Air Mineral - Minuman - Rp8000.00
4. Nasi Rendang - Makanan - Rp30000.00
5. Nasi Ayam - Makanan - Rp25000.00
Pilih kode produk atau 's' untuk selesai: 3
Masukkan Jumlah Produk: 1
Produk dengan kode 3 sebanyak 1 berhasil ditambahkan ke transaksi.

```

Transaksi tersimpan di Database:

The screenshot shows a database management interface with the 'Detail_Transaksi' table selected. The table contains the following data:

id_detail	no_transaksi	kode_produk	jumlah	total_harga
1	1	1	3	15000.00
2	2	2	1	8000.00
3	2	4	1	30000.00

G. Cetak Struk

```

elif pilihan == "4":

    no_transaksi = input("Masukkan No Transaksi yang
ingin dicetak: ")

```

```

        cursor.execute("""
            SELECT t.no_transaksi, t.tanggal, p.nama, d.jumlah,
pr.nama_produk, pr.jenis_produk, d.total_harga
            FROM Transaksi t
            JOIN Detail_Transaksi d ON t.no_transaksi =
d.no_transaksi
            JOIN Pegawai p ON t.nik = p.nik
            JOIN Produk pr ON d.kode_produk = pr.kode_produk
            WHERE t.no_transaksi = %s
        """, (no_transaksi,))

    transaksi = cursor.fetchall()
    if not transaksi:
        print("Transaksi tidak ditemukan.")
        continue

    print("\n--- Struk Transaksi ---")
    print(f"No Transaksi: {transaksi[0][0]}")
    print(f"Tanggal: {transaksi[0][1]}")
    print(f"Pegawai: {transaksi[0][2]}")
    print("Produk yang dibeli:")
    for t in transaksi:
        print(f"- {t[4]} ({t[5]}), Jumlah: {t[3]}, Total
Harga: Rp{t[6]}")
    print("-----")

```

Untuk mencetak struk transaksi, program meminta no_transaksi yang ingin dicetak. Kemudian, program mengambil data transaksi dari beberapa tabel yang terhubung (Transaksi, Detail_Transaksi, Pegawai, Produk) dan menampilkan detail transaksi termasuk nama pegawai, produk yang dibeli, jumlah, dan total harga.

```

Menu:
1. Tambah Pegawai
2. Tambah Produk
3. Lakukan Transaksi
4. Cetak Struk
5. Keluar
Pilih menu: 4
Masukkan No Transaksi yang ingin dicetak: 2

--- Struk Transaksi ---
No Transaksi: 2
Tanggal: 2024-12-12
Pegawai: seungmin
Produk yang dibeli:
- Air Mineral (Minuman), Jumlah: 1, Total Harga: Rp8000.00
- Nasi Rendang (Makanan), Jumlah: 1, Total Harga: Rp30000.00
-----

```


III. Kesimpulan

Program ini memungkinkan pengguna untuk mengelola data penjualan dengan menyimpan informasi pegawai, produk, dan transaksi. Setiap transaksi yang dilakukan dapat dicatat dengan detail produk yang dibeli. Selain itu, pengguna juga dapat mencetak struk transaksi untuk melihat rincian pembelian yang dilakukan. Program ini memanfaatkan database MySQL untuk menyimpan data, dan menggunakan query SQL untuk berinteraksi dengan database.