

**LAPORAN TUGAS DATABASE
PROGRAM KASIR PENJUALAN**

PEMROGRAMAN BERBASIS OBJEK



5230411327

Muhammad Irfan Baihaqi

Kelas VIII

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

2024

PEMBAHASAN

1. KODE SQL DDL UNTUK MEMBUAT STRUKTUR TABEL

1.1. Tabel Pegawai

```
mysql> SHOW CREATE TABLE Pegawai;
+-----+-----+
| Table | Create Table |
+-----+-----+
| Pegawai | CREATE TABLE `pegawai` (
  `nik_pegawai` bigint NOT NULL,
  `nama_pegawai` varchar(250) NOT NULL,
  `alamat_pegawai` varchar(250) NOT NULL,
  PRIMARY KEY (`nik_pegawai`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)
```

Struktur Tabel Pegawai:

```
mysql> DESC Pegawai;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nik_pegawai | bigint | NO | PRI | NULL | |
| nama_pegawai | varchar(250) | NO | | NULL | |
| alamat_pegawai | varchar(250) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

1.2. Membuat Tabel Produk

```
mysql> CREATE TABLE Produk(
  -> id_produk CHAR(4) NOT NULL PRIMARY KEY,
  -> nama_produk VARCHAR(250) NOT NULL,
  -> jenis_produk ENUM("Snack", "Minuman", "Makanan") NOT NULL,
  -> harga INT NOT NULL
  -> );
Query OK, 0 rows affected (0.02 sec)
```

Struktur Tabel Produk:

```
mysql> DESC Produk;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_produk | char(4) | NO | PRI | NULL | |
| nama_produk | varchar(250) | NO | | NULL | |
| jenis_produk | enum('Snack', 'Minuman', 'Makanan') | NO | | NULL | |
| harga | int | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

1.3. Membuat Tabel Transaksi

```
mysql> CREATE TABLE Transaksi(
-> id_transaksi CHAR(4) NOT NULL PRIMARY KEY,
-> detail_transaksi VARCHAR(250) NOT NULL
-> );
Query OK, 0 rows affected (0.02 sec)
```

Struktur Tabel Transaksi:

```
mysql> DESC Transaksi;
```

Field	Type	Null	Key	Default	Extra
id_transaksi	char(4)	NO	PRI	NULL	
detail_transaksi	varchar(250)	NO		NULL	

```
2 rows in set (0.00 sec)
```

1.4. Tabel Struk

```
mysql> SHOW CREATE TABLE Struk;
+-----+
| Table | Create Table
+-----+
| Struk | CREATE TABLE `struk` (
  `id_struk` char(4) NOT NULL,
  `id_transaksi` char(4) NOT NULL,
  `id_produk` char(4) NOT NULL,
  `nik_pegawai` bigint DEFAULT NULL,
  `jumlah_produk` int NOT NULL,
  `waktu_transaksi` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_struk`,`id_transaksi`),
  KEY `fk_id_transaksi` (`id_transaksi`),
  KEY `fk_id_produk` (`id_produk`),
  KEY `fk_nik_pegawai` (`nik_pegawai`),
  CONSTRAINT `fk_id_produk` FOREIGN KEY (`id_produk`) REFERENCES `produk` (`id_produk`),
  CONSTRAINT `fk_id_transaksi` FOREIGN KEY (`id_transaksi`) REFERENCES `transaksi` (`id_transaksi`),
  CONSTRAINT `fk_nik_pegawai` FOREIGN KEY (`nik_pegawai`) REFERENCES `pegawai` (`nik_pegawai`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)
```

Struktur Tabel Struk:

```
mysql> DESC Struk;
```

Field	Type	Null	Key	Default	Extra
id_struk	char(4)	NO	PRI	NULL	
id_transaksi	char(4)	NO	PRI	NULL	
id_produk	char(4)	NO	MUL	NULL	
nik_pegawai	bigint	YES	MUL	NULL	
jumlah_produk	int	NO		NULL	
waktu_transaksi	datetime	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED

6 rows in set (0.00 sec)

1.5. Membuat Tabel Pembayaran (Tambahan)

```
mysql> CREATE TABLE Pembayaran(
  -> id_pembayaran CHAR(4) NOT NULL PRIMARY KEY,
  -> id_struk CHAR(4) NOT NULL,
  -> metode_bayar ENUM("TUNAI","DANA","OVO","GOPAY","LINKAJA","SHOPEEPAY") NOT NULL,
  -> membayar INT NOT NULL,
  -> CONSTRAINT fk_id_struk FOREIGN KEY (id_struk) REFERENCES Struk(id_struk)
  -> );
Query OK, 0 rows affected (0.07 sec)
```

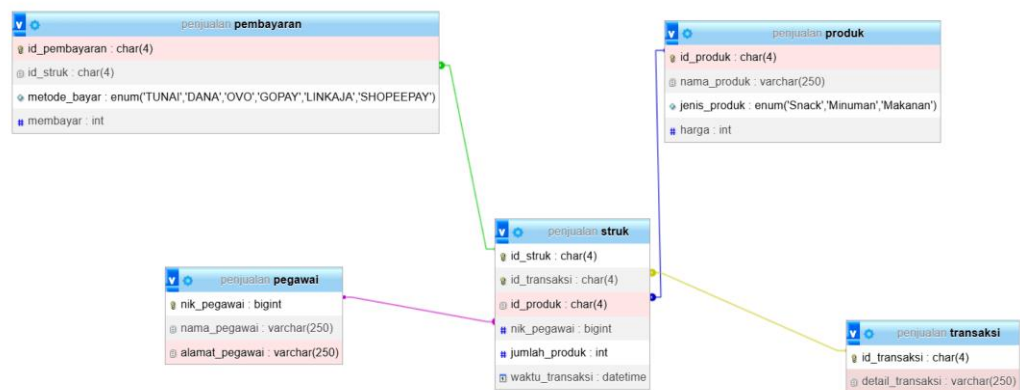
Struktur Tabel Pembayaran:

```
mysql> DESC Pembayaran;
```

Field	Type	Null	Key	Default	Extra
id_pembayaran	char(4)	NO	PRI	NULL	
id_struk	char(4)	NO	MUL	NULL	
metode_bayar	enum('TUNAI','DANA','OVO','GOPAY','LINKAJA','SHOPEEPAY')	NO		NULL	
membayar	int	NO		NULL	

4 rows in set (0.00 sec)

2. Diagram Relasi Antar Tabel



3. Kode Program

3.1 Modul Yang dibutuhkan

```
import mysql.connector  
from mysql.connector import Error
```

```
from tabulate import tabulate  
import string  
import random  
import os  
from reportlab.lib.pagesizes import letter  
from reportlab.pdfgen import canvas
```

from tabulate import tabulate import string import random import os from reportlab.lib.pagesizes import letter from reportlab.pdfgen import canvas jelaskan kode di atas dan kegunaan modul-modul nya dalam 1 paragraf . Kode tersebut mengimpor beberapa modul yang berguna untuk berbagai tugas dalam pemrograman. Modul tabulate digunakan untuk menyajikan data dalam bentuk tabel yang rapi, baik di konsol maupun dalam file. string dan random digunakan bersama-sama untuk menghasilkan string acak, seperti ID unik atau kata sandi. os berfungsi untuk berinteraksi dengan sistem operasi, misalnya untuk menghapus layar atau menjalankan perintah sistem lainnya. Terakhir, reportlab.lib.pagesizes dan reportlab.pdfgen.canvas adalah modul dari ReportLab yang memungkinkan pembuatan dan pengaturan halaman serta menghasilkan dokumen PDF dengan tata letak yang kompleks, seperti faktur atau

laporan. Kombinasi modul-modul ini memungkinkan pembuatan aplikasi yang dapat menampilkan data dengan rapi, berinteraksi dengan sistem, dan menghasilkan dokumen PDF yang professional

3.2 Membuat Fungsi Koneksi ke database

```
def koneksi():  
    try:  
        # Koneksi ke database  
        connection = mysql.connector.connect(  
            host='localhost',  
            database='penjualan',  
            user='root',  
            password=''  
        )  
        return connection  
  
    except Error as e:  
        print(  
            "Error saat mencoba menghubungkan ke MySQL", e)
```

Fungsi koneksi() bertujuan untuk menghubungkan aplikasi dengan database MySQL. Fungsi ini menggunakan modul mysql.connector untuk membuat koneksi ke database dengan parameter seperti host, database, user, dan password. Jika koneksi berhasil, fungsi akan mengembalikan objek koneksi tersebut. Jika terjadi kesalahan saat mencoba menghubungkan ke MySQL, blok except akan menangkap kesalahan tersebut dan mencetak pesan error yang sesuai menggunakan print("Error saat mencoba menghubungkan ke MySQL", e)

3.3 Method untuk menambahkan data pegawai ke database

```
@classmethod
def tambahpegawai(cls):
    while True:
        nikstr = input("Masukkan NIK pegawai: ")
        cek = nikstr.isdigit()
        if cek == True:
            nik = int(nikstr)
            nama = input(
                "Masukkan Nama pegawai: ")
            alamat = input(
                "Masukkan alamat pegawai: ")
            conn = koneksi()
            cur = conn.cursor()
            cur.execute(
                "INSERT INTO pegawai VALUES(%s,%s,%s)", (nik,nama,
                alamat))
            conn.commit()
            conn.close()

            # cls.daftarpegawai.append(pegawaibaru) #objek dimasukan ke dalam list class
            os.system('cls')
            print(
                "PEGAWAI BERHASIL DITAMBAHKAN...")
            os.system('pause')
            return True
        else:
            print("NIK HARUS BERUPA ANGKA")
```

Metode tambahpegawai adalah method class yang digunakan untuk menambahkan data pegawai ke dalam database. Proses ini dilakukan dalam loop while True sehingga pengguna dapat terus memasukkan data sampai NIK yang valid diberikan. Metode ini dimulai dengan meminta input NIK dari pengguna dan memeriksa apakah input tersebut berupa angka menggunakan metode isdigit(). Jika benar, input diubah menjadi integer dan nama serta alamat pegawai diminta. Kemudian, koneksi ke database dibuat menggunakan fungsi koneksi(). Selanjutnya, sebuah cursor dibuat dan eksekusi query INSERT INTO pegawai VALUES(%s,%s,%s) dilakukan untuk menambahkan data pegawai ke

dalam database. Setelah perubahan dikomit, koneksi ditutup dan layar dibersihkan dengan `os.system('cls')`. Jika NIK yang dimasukkan bukan angka, pesan error ditampilkan untuk memberitahu pengguna bahwa NIK harus berupa angka

3.4 Method untuk menampilkan seluruh pegawai

```
@classmethod
def tampilsemuapegawai(cls):
    conn = koneksi()
    if conn is None:
        print("Koneksi ke database gagal.")
        return
    cur = conn.cursor()
    cur.execute("SELECT * FROM pegawai")
    data = cur.fetchall()
    conn.close()
    header = ["NIK", "Nama Pegawai", "Alamat Pegawai"]
    # Mengakses elemen tuple menggunakan indeks
    dataUser = [[data[0], data[1], data[2]] for data in data]
    print(tabulate(dataUser, headers=header, tablefmt="grid"))
```

Metode `tampilsemuapegawai` adalah method class yang digunakan untuk menampilkan semua data pegawai dari database dalam bentuk tabel yang rapi. Pertama, koneksi ke database dibuat menggunakan fungsi `koneksi()`. Jika koneksi gagal, pesan error ditampilkan dan eksekusi dihentikan. Kemudian, cursor dibuat dan query `SELECT * FROM pegawai` dijalankan untuk mengambil semua data pegawai. Hasil query tersebut disimpan dalam variabel `data`, dan koneksi ke database ditutup. Selanjutnya, elemen-elemen tuple dalam `data` diakses menggunakan indeks untuk membuat daftar `dataUser`. Akhirnya, data pegawai ditampilkan dalam bentuk tabel menggunakan fungsi `tabulate`, dengan header kolom "NIK", "Nama Pegawai", dan "Alamat Pegawai".

3.5 Method untuk menghapus pegawai

```
@classmethod
def hapuspegawai(cls):
    try:
        conn = koneksi()
        if conn is None:
            print("Koneksi ke database gagal.")
            return
        cur = conn.cursor()
        Pegawai.tampilsemuapegawai()
        try:
            nik_hapus = int(input(
                "Masukkan NIK pegawai yang akan dihapus: "))
        except ValueError:
            print("NIK harus berupa angka.")
            return
        cur.execute(
            "SELECT * FROM pegawai WHERE nik_pegawai = %s", (
                nik_hapus,))
        pegawai = cur.fetchone()
        if pegawai:
            cur.execute(
                "DELETE FROM pegawai WHERE nik_pegawai = %s", (
                    nik_hapus,))
            conn.commit()
            conn.close()
            print("Pegawai berhasil dihapus.")
            return True
        else:
            print("Pegawai tidak ditemukan.")
            return
    except mysql.connector.Error as e:
        #pegawai sudah pernah melakukan transaksi
        print("Error saat menghapus data pegawai:", e
    )
    return
```

Metode hapuspegawai adalah method class yang digunakan untuk menghapus data pegawai dari database. Pertama, fungsi mencoba menghubungkan ke database menggunakan koneksi(). Jika koneksi gagal, pesan error akan dicetak dan proses dihentikan. Setelah berhasil terhubung, method tampilsemuapegawai dipanggil untuk menampilkan semua data pegawai. Kemudian, pengguna diminta untuk memasukkan NIK pegawai yang akan dihapus. Jika NIK bukan angka, pesan error akan ditampilkan dan proses dihentikan. Setelah validasi, query dijalankan untuk memeriksa apakah pegawai dengan NIK tersebut ada di database. Jika ada, data pegawai akan dihapus dengan query DELETE. Jika tidak ada, pesan bahwa pegawai tidak ditemukan akan dicetak. Jika terjadi

kesalahan selama penghapusan, khususnya jika pegawai tersebut berhubungan dengan transaksi lain, pesan error akan ditampilkan

3.6 Membuat id dengan 4 karakter

```
@staticmethod
def generate_id(): #METHOD UNTUK MEMBUAT ID
    characters = string.ascii_letters + string.digits
    unique_id = ''.join(random.choices(characters, k=4))
    return unique_id
```

Metode `generate_id` adalah static method yang digunakan untuk membuat ID unik. Metode ini menggabungkan huruf ASCII (baik huruf besar maupun kecil) dan angka untuk membentuk kumpulan karakter (`characters`). Kemudian, menggunakan fungsi `random.choices`, metode ini memilih secara acak empat karakter dari kumpulan tersebut untuk membentuk `unique_id`. Hasilnya adalah string acak sepanjang empat karakter yang terdiri dari campuran huruf dan angka, yang kemudian dikembalikan oleh metode ini. Metode ini sangat berguna untuk menghasilkan ID unik yang bisa digunakan sebagai pengenal dalam berbagai konteks, seperti ID transaksi atau kode unik lainnya.

3.7 Menambahkan Transaksi

```
@classmethod
def tambahtransaksi(cls):
    try:
        totalBayar = 0
        cls.tampilsemuapegawai()
        conn = koneksi()
        cur = conn.cursor()
        nik_pegawai = input("Masukkan kode pegawai yang membuat transaksi: ")
        cur.execute("SELECT * FROM pegawai WHERE nik_pegawai = %s", (nik_pegawai,))
        pegawai = cur.fetchone()
        if pegawai:
            id_struk = cls.generate_id()

# customer melakukan transaksi pembelian beberapa barang dalam 1 waktu > maka akan menghasilkan 1 struk

# setiap pembelian barang menghasilkan id transaksi , tetapi masih 1 id struk yang sama JIKA MASIH UNTUK CUSTOMER ITU DAN DALAM WAKTU ITU.
        while True:
            Produk.tampilkanproduk()
            produkbeli = input("SILAHKAN MASUKKAN KODE PRODUK: ")
            cur.execute("SELECT * FROM produk WHERE id_produk = %s", (produkbeli,))
            produk = cur.fetchone()
            if produk:
                id_transaksi = cls.generate_id()
                try:
                    jumlah_produk = int(input("Masukkan jumlah produk yang dibeli: "))
                    if jumlah_produk > 0:
                        Transaksi.createTransaksi(id_transaksi)
                        # print("P berhasil membuat transaksi") #debug
                        Struk.createStruk(id_struk, id_transaksi, produkbeli, nik_pegawai,
                        jumlah_produk)

                        totalBayar += produk[3] * jumlah_produk
                        sudah = input("Apakah TRANSAKSI SUDAH SELESAI (y/n)? ")
                        if sudah.lower() == "y":
                            cls.pembayaran(id_struk, totalBayar)
                            Struk.buat_struk_pdf_dari_db(id_struk)

#akan membuat pdf file untuk struk kali ini
                            return True
                        else:
                            continue
                    else: #debug
                        print("gagal menambahkan produk beli, jumlah beli harus lebih dari 0")
                        os.system('pause')
                except ValueError:
                    print("JUMLAH PRODUK HARUS BERUPA ANGKA")
                    print("gagal menambahkan produk beli")
                    os.system('pause')
                    continue
            else:
                print("PRODUK TIDAK ADA")
                os.system('pause')
                os.system('cls')
                continue
        else:
            print("Pegawai Tidak Ada , GAGAL MELAKUKAN TRANSAKSI...")
            os.system('pause')
            return True
    except Exception as e:
        print("Error:", e)
```

Metode tambahtransaksi adalah method class yang memungkinkan penambahan transaksi baru di database. Proses dimulai dengan menampilkan semua pegawai yang tersedia dan menghubungkan ke database. Pengguna kemudian diminta untuk memasukkan kode pegawai yang membuat transaksi, dan sistem memverifikasi keberadaan pegawai tersebut di database. Jika pegawai ditemukan, sebuah ID struk unik

dihasilkan. Loop while True digunakan untuk memungkinkan pembelian beberapa barang dalam satu transaksi. Setiap barang yang dibeli dicari di database berdasarkan kode produk, dan jika ditemukan, transaksi dibuat dengan ID transaksi unik dan barang tersebut ditambahkan ke struk dengan detail jumlah produk dan harga. Total pembayaran dihitung dan pengguna diminta untuk menyelesaikan transaksi atau melanjutkan pembelian barang lainnya.

Jika transaksi selesai, sistem memanggil method pembayaran untuk memproses pembayaran dan membuat file PDF struk untuk transaksi tersebut. Metode ini juga menangani berbagai skenario kesalahan, seperti NIK pegawai yang tidak valid, produk tidak ditemukan, atau input jumlah produk yang bukan angka, dengan memberikan pesan kesalahan yang relevan kepada pengguna. Jika terjadi kesalahan umum lainnya, pesan error ditampilkan. Proses ini memastikan bahwa data transaksi dimasukkan dengan benar ke dalam database dan diakhiri dengan pembuatan struk PDF yang berisi detail transaksi yang lengkap

3.8 Membuat data pembayaran

```
@classmethod
def pembayaran(cls, id_struk, totalBayar):
    conn = koneksi()
    cur = conn.cursor()
    metodeBayar = {
        "1" : "TUNAI",
        "2" : "DANA",
        "3" : "OVO",
        "4" : "GOPAY",
        "5" : "LINKAJA",
        "6" : "SHOPEEPAY"
    }
    while True:
        for key, value in metodeBayar.items():
            print(f"{key}. {value}")
        metode_bayar = input("Pilih metode pembayaran (1-6): ")
        membayar = input("Masukkan jumlah uang yang dibayarkan: ")
        cek = membayar.isdigit()
        if metode_bayar in metodeBayar and cek == True:
            membayar = int(membayar)
            if membayar < totalBayar:
                print("Uang tidak cukup.")
                os.system('pause')
                continue
            else:
                conn = koneksi()
                cur = conn.cursor()
                cur.execute("INSERT INTO Pembayaran(id_pembayaran, id_struk, metode_bayar, membayar) VALUES (
%s, %s, %s, %s)", (cls.generate_id(), id_struk, metode_bayar, membayar))
                conn.commit()
                print("Pembayaran berhasil.")
                os.system('pause')
                return True
        else:
            print("Metode pembayaran tidak valid. Silakan pilih metode yang benar.")
```

Metode pembayaran adalah method class yang menangani proses pembayaran untuk suatu transaksi tertentu yang diidentifikasi dengan id_struk dan totalBayar. Pertama, metode ini menghubungkan ke database dan mendefinisikan berbagai metode pembayaran yang tersedia. Dalam loop while True, metode pembayaran yang tersedia ditampilkan, dan pengguna diminta untuk memilih salah satu metode serta memasukkan jumlah uang yang dibayarkan. Validasi dilakukan untuk memastikan bahwa metode pembayaran yang dipilih valid dan jumlah yang dibayarkan berupa angka. Jika jumlah uang yang dibayarkan kurang dari total yang harus dibayar, pesan "Uang tidak cukup" ditampilkan dan pengguna diminta untuk mencoba lagi. Jika valid, informasi pembayaran dimasukkan ke dalam database, transaksi dikonfirmasi berhasil, dan loop dihentikan. Jika metode pembayaran tidak valid, pesan error ditampilkan dan pengguna diminta untuk memilih metode yang benar.

3.9 Membuat Struk dalam bentuk pdf pada id struk yang dipilih

```
def printStruckToPDFbyID(cls, id_struk):  
    Struk.tampilsemuastruk()  
    id_struk = input("Masukkan ID Struk: ")  
    conn = koneksi()  
    cur = conn.cursor()  
    cur.execute(  
        "SELECT * FROM struk WHERE id_struk = %s", (  
            id_struk,))  
    struk = cur.fetchone()  
    if struk:  
        Struk.buat_struk_pdf_dari_db(id_struk)  
        return True  
    print("Struk tidak ditemukan.")  
    os.system('pause')  
    os.system('cls')  
    return False
```

Metode printStruckToPDFbyID bertujuan untuk mencetak struk ke dalam format PDF berdasarkan ID struk yang diberikan. Pertama, metode Struk.tampilsemuastruk() dipanggil untuk menampilkan semua struk yang ada. Kemudian, pengguna diminta untuk memasukkan ID struk yang diinginkan. Fungsi ini kemudian membuka koneksi ke database dan menjalankan query untuk mencari struk dengan ID yang diberikan. Jika struk ditemukan, metode Struk.buat_struk_pdf_dari_db(id_struk) dipanggil untuk membuat file PDF struk tersebut, dan fungsi mengembalikan True. Jika struk tidak ditemukan, pesan "Struk tidak ditemukan" ditampilkan, layar dibersihkan, dan fungsi mengembalikan False. Fungsi ini mempermudah pencetakan struk dalam bentuk PDF secara dinamis berdasarkan ID yang dimasukkan oleh pengguna.

3.10 Class Transaksi dengan methodnya createtransaksi untuk membuat / memasukkan data transaksi per barang ke database

```
class Transaksi:

    def __init__(self) -> None:
        pass

    @classmethod
    def createTransaksi(cls, id_transaksi): # METHOD UNTUK MEMBUAT TRANSAKSI BARU
        conn = koneksi()
        if conn is None:
            print("Koneksi ke database gagal.")
            return
        cur = conn.cursor()

        try:
            masukan_catatan = input("Apakah ingin Masukkan catatan untuk transaksi barang ini? (y/n): ")
            if masukan_catatan.lower() == "y":
                catatan = input("Masukkan catatan: ")
                detail_transaksi = catatan
                cur.execute("INSERT INTO transaksi (id_transaksi, detail_transaksi) VALUES (%s, %s)", (
                    id_transaksi, detail_transaksi))
            else:
                detail_transaksi = ""
                cur.execute("INSERT INTO transaksi (id_transaksi, detail_transaksi) VALUES (%s, %s)", (
                    id_transaksi, detail_transaksi))

            conn.commit()
            # print("berhasil dibuat.")
        except mysql.connector.Error as e:
            print("Error saat memasukkan data ke tabel transaksi:", e)
        except Exception as ex:
            print("Unexpected error:", ex)
        finally:
            if cur:
                cur.close()
            if conn:
                conn.close()
```

Kelas Transaksi adalah sebuah blueprint untuk membuat dan mengelola transaksi dalam aplikasi. Kelas ini memiliki konstruktor `__init__` yang tidak melakukan apa-apa (didefinisikan dengan `pass`), namun tetap ada untuk memenuhi persyaratan inisialisasi dalam Python. Kelas ini juga memiliki method kelas `createTransaksi` yang bertujuan untuk membuat transaksi baru di dalam database. Method ini pertama-tama mencoba membuat koneksi ke database menggunakan fungsi `koneksi()`. Jika koneksi gagal, pesan kesalahan akan ditampilkan. Kemudian, metode ini membuat cursor dan meminta pengguna untuk memasukkan catatan untuk transaksi. Berdasarkan input pengguna, catatan disimpan dalam variabel `detail_transaksi` dan kemudian dimasukkan ke dalam tabel transaksi di database dengan ID transaksi yang diberikan. Semua perubahan dikomit ke database.

Dalam blok try, method ini menangani input pengguna dan query database dengan eksekusi INSERT INTO untuk menambahkan catatan transaksi. Jika terjadi kesalahan, blok except menangkap error dari mysql.connector atau error umum lainnya dan mencetak pesan kesalahan yang relevan. Blok finally memastikan bahwa cursor dan koneksi ke database selalu ditutup, terlepas dari apakah operasi berhasil atau terjadi kesalahan. Metode ini memastikan bahwa transaksi baru dapat dibuat dengan informasi lengkap dan sistem menangani setiap potensi kesalahan dengan cara yang terstruktur. Dengan demikian, metode ini berfungsi sebagai bagian penting dalam pengelolaan transaksi di dalam aplikasi.

3.11 Pendefinisian Class Struk untuk menampung method – method yang terkait dengan struk.

```
class Struk:  
  
    def __init__(self) -> None:  
        pass
```


3.12 Method untuk membuat struk

```
@classmethod
def createStruk(cls, id_struk, id_transaksi, produkbeli, nik_pegawai, jumlah_produk):
    try:

        conn = koneksi()
        if conn is None:
            print("Koneksi ke database gagal.")
            return

        cur = conn.cursor()
        query = ("INSERT INTO struk (id_struk, id_transaksi, id_produk, nik_pegawai, jumlah_produk) "
                "VALUES (%s, %s, %s, %s, %s)")
        values = (id_struk, id_transaksi, produkbeli, nik_pegawai, jumlah_produk)

        cur.execute(query, values)
        conn.commit()
        conn.close()

    except mysql.connector.Error as e:
        print("Error saat memasukkan data ke tabel struk:", e)
    except Exception as ex:
        print("Unexpected error:", ex)
    finally:
        if conn.is_connected():
            cur.close()
            conn.close()
```

Metode createStruk adalah method class yang digunakan untuk membuat dan menyimpan struk baru dalam database. Metode ini pertama-tama mencoba untuk menghubungkan ke database menggunakan fungsi koneksi(). Jika koneksi gagal, pesan error ditampilkan dan eksekusi dihentikan. Jika koneksi berhasil, cursor dibuat dan query SQL disiapkan untuk memasukkan data struk baru ke dalam tabel struk dengan nilai yang diberikan (ID struk, ID transaksi, produk yang dibeli, NIK pegawai, dan jumlah produk). Query ini kemudian dieksekusi, perubahan dikomit, dan koneksi ditutup.

Jika terjadi kesalahan saat memasukkan data ke dalam tabel, blok except menangkap error dari mysql.connector dan mencetak pesan kesalahan yang relevan. Blok finally memastikan bahwa cursor dan koneksi selalu ditutup jika koneksi masih aktif, menjaga integritas dan kebersihan proses.

3.13 Method untuk Menampilkan Semua Riwayat Struk

```
@classmethod
def tampilsemuastruk(cls):
    conn = koneksi()
    cur = conn.cursor()
    cur.execute(
        "SELECT s.id_struk, s.id_transaksi, p.nama_produk, s.jumlah_produk, p.harga, s.waktu_transaksi, pg.nama_pegawai
        FROM struk s JOIN produk p ON s.id_produk = p.id_produk JOIN pegawai pg ON s.nik_pegawai = pg.nik_pegawai"
    )
    data = cur.fetchall()
    header = ["ID Struk", "ID Transaksi", "Nama Produk", "Nama Pegawai", "Jumlah Produk", "Harga Produk",
        "Total Harga", "Waktu Transaksi"]
    datastruk = [[o[0], o[1], o[2], o[6], o[3], o[4], o[3] * o[4], o[5]] for o in data]
    conn.close()
    print(tabulate(datastruk, headers=header, tablefmt="grid"))
    os.system('pause')
```

Metode `tampilsemuastruk` adalah method class yang digunakan untuk menampilkan semua data struk dari database dalam bentuk tabel. Pertama, koneksi ke database dibuat menggunakan fungsi `koneksi()`. Setelah itu, cursor dibuat dan query SQL dijalankan untuk mengambil semua data struk yang melibatkan join antara tabel struk, produk, dan pegawai untuk mendapatkan informasi lengkap tentang struk, produk, dan pegawai yang terlibat. Hasil query disimpan dalam variabel `data` dan kemudian diolah ke dalam list `datastruk` yang menyusun ulang data dengan menambahkan kolom "Total Harga" yang merupakan hasil perkalian jumlah produk dengan harga produk. Setelah koneksi ditutup, data struk ditampilkan dalam bentuk tabel menggunakan fungsi `tabulate` dengan header yang sudah ditentukan dan format tabel yang rapi. Terakhir, fungsi `os.system('pause')` digunakan untuk menjeda tampilan agar pengguna dapat membaca data yang ditampilkan.

3.14 Method untuk menampilkan Semua Struk di hari tertentu.

```
@classmethod
def tampilstrukbywaktu(cls):
    try:
        tanggal = input("Masukkan tanggal (YYYY-MM-DD): ")
        conn = koneksi()
        cur = conn.cursor()
        cur.execute(
            "SELECT s.id_struk, s.id_transaksi, p.nama_produk, s.jumlah_produk, p.harga, s.waktu_transaksi, pg.nama_pegawa
            i FROM struk s JOIN produk p ON s.id_produk = p.id_produk JOIN pegawai pg ON s.nik_pegawai = pg.nik_pegawai WH
            ERE DATE(s.waktu_transaksi) =
            %s", (tanggal,))
        data = cur.fetchall()
        header = ["ID Struk", "ID Transaksi", "Nama Produk", "Nama Pegawai", "Jumlah Produk", "Harga Produk",
            "Total Harga", "Waktu Transaksi"]
        datastruk = [[o[0], o[1], o[2], o[6], o[3], o[4], o[3] * o[4], o[5]] for o in data]
        conn.close()
        if datastruk:
            print(tabulate(datastruk, headers=header, tablefmt="grid"))
        else:
            print("Tidak ada data struk pada tanggal tersebut.")
            os.system('pause')
        os.system('pause')
    except Exception as e:
        print("Error:", e)
        os.system('pause')
```

Metode `tampilstrukbywaktu` adalah method class yang digunakan untuk menampilkan semua struk yang terjadi pada tanggal tertentu. Pertama, pengguna diminta untuk memasukkan tanggal dalam format YYYY-MM-DD. Kemudian, koneksi ke database dibuat menggunakan fungsi `koneksi()`, dan cursor digunakan untuk menjalankan query yang mencari semua struk yang sesuai dengan tanggal tersebut, melakukan join antara tabel struk, produk, dan pegawai untuk mendapatkan informasi lengkap. Hasil query disimpan dalam variabel `data` dan diolah ke dalam list `datastruk` dengan menambahkan kolom "Total Harga". Setelah koneksi ditutup, jika ada data struk yang ditemukan, data ditampilkan dalam bentuk tabel menggunakan fungsi `tabulate`. Jika tidak ada data ditemukan, pesan "Tidak ada data struk pada tanggal tersebut" ditampilkan. Metode ini juga menangani kesalahan dengan mencetak pesan error jika terjadi masalah selama proses.

3.15 Method untuk membuat pdf atau mencetak Struk

```
@classmethod
def buat_struk_pdf_dari_db(cls, id_struk="0000", nama_file="struk.pdf"):
    # Koneksi ke database
    conn = koneksi()
    if conn is None:
        print("Koneksi ke database gagal.")
        return
    cur = conn.cursor(dictionary=True)

    try:
        # Query untuk header dan item transaksi
        cur.execute(
            """
            SELECT
                struk.id_transaksi,
                struk.waktu_transaksi,
                pegawai.nama_pegawai AS kasir,
                produk.nama_produk AS nama,
                produk.harga AS harga_satuan,
                struk.jumlah_produk,
                (struk.jumlah_produk * produk.harga) AS subtotal,
                pembayaran.membayar AS Bayar,
                pembayaran.metode_bayar AS Metode_Bayar
            FROM struk
            JOIN produk ON struk.id_produk = produk.id_produk
            JOIN pegawai ON struk.nik_pegawai = pegawai.nik_pegawai
            JOIN pembayaran ON struk.id_struk = pembayaran.id_struk
            WHERE struk.id_struk = %s
            """
            ,
            (id_struk,)
        )
        print("Query berhasil dijalankan.")

        data_items = cur.fetchall()
        # print(f"Data items berhasil diambil: {data_items}") # Debugging print statement

        if not data_items:
            print("Tidak ada data transaksi dengan ID tersebut.")
            os.system("pause")
            return

        # Debugging print statement
        # print(f"Isi Data items setelah pengecekan: {data_items}")

        # Data untuk header
        data_struk = {
            'id_transaksi': data_items[0]['id_transaksi'],
            'tanggal': data_items[0]['waktu_transaksi'],
            'kasir': data_items[0]['kasir'],
            'item': [
                {
                    'nama': item['nama'],
                    'qty': item['jumlah_produk'],
                    'harga_satuan': item['harga_satuan'],
                    'subtotal': item['subtotal'],
                }
                for item in data_items
            ],
            'total': sum(item['subtotal'] for item in data_items),
            'pembayaran': data_items[0]['Bayar'], # Sesuaikan dengan data pembayaran di database
            'kembalian': data_items[0]['Bayar'] - sum(item['subtotal'] for item in data_items),
        }

    except mysql.connector.Error as e:
        print("Error saat mengambil data dari database:", e)
        return
    except Exception as e:
        print("Unexpected error:", e)
        return
    finally:
        # Tutup koneksi database
        cur.close()
        conn.close()
```

```

nama_file = f"Struk_{id_struk}.pdf"
print(f"Menyimpan file PDF dengan nama: {nama_file}") # Debugging
c = canvas.Canvas(nama_file, pagesize=letter)
width, height = letter

try:
    # Header Struk
    c.setFont("Helvetica-Bold", 14)
    c.drawCentredString(width / 2, height - 50, "Warung Bebas Galau")
    c.setFont("Helvetica", 10)
    c.drawCentredString(width / 2, height - 65, "Jl. Kebahagiaan No. 42, Yogyakarta")
    c.drawCentredString(width / 2, height - 80, "Telp: 0812-3456-7890")

    # Informasi Transaksi
    c.setFont("Helvetica", 10)
    y_position = height - 120
    c.drawString(50, y_position, f"ID Transaksi: {data_struk['id_transaksi']}")
    c.drawString(50, y_position - 15, f"Tanggal : {data_struk['tanggal']}")
    c.drawString(50, y_position - 30, f"Kasir : {data_struk['kasir']}")

    # Garis pemisah
    c.line(50, y_position - 40, width - 50, y_position - 40)

    # Header Item
    y_position -= 60
    c.drawString(50, y_position, "Item")
    c.drawString(200, y_position, "Qty")
    c.drawString(250, y_position, "Harga Satuan")
    c.drawString(350, y_position, "Subtotal")

    # Isi Item
    y_position -= 20
    for item in data_struk['item']:
        c.drawString(50, y_position, item['nama'])
        c.drawString(200, y_position, str(item['qty']))
        c.drawString(250, y_position, f"Rp {item['harga_satuan']:,}")
        c.drawString(350, y_position, f"Rp {item['subtotal']:,}")
        y_position -= 15

    # Garis pemisah kedua
    c.line(50, y_position - 5, width - 50, y_position - 5)

    # Total dan Pembayaran
    y_position -= 30
    c.drawString(250, y_position, "Total:")
    c.drawString(350, y_position, f"Rp {data_struk['total']:,}")

    y_position -= 15
    c.drawString(250, y_position, "Pembayaran:")
    c.drawString(350, y_position, f"Rp {data_struk['pembayaran']:,}")

    y_position -= 15
    c.drawString(250, y_position, "Kembalian:")
    c.drawString(350, y_position, f"Rp {data_struk['kembalian']:,}")

    # Footer Struk
    y_position -= 40
    c.setFont("Helvetica", 10)
    c.drawCentredString(width / 2, y_position, "Terima kasih telah berbelanja di Warung Bebas Galau")

    # Simpan file PDF
    c.save()
    print("File PDF berhasil disimpan.") # Debugging
except Exception as e:
    print("Error saat membuat PDF:", e)

```

Metode `buat_struk_pdf_dari_db` adalah method class yang digunakan untuk menghasilkan file PDF struk berdasarkan data dari database. Pertama, koneksi ke database dibuat menggunakan fungsi `koneksi()`. Jika koneksi gagal, pesan error akan ditampilkan dan eksekusi dihentikan. Jika koneksi berhasil, cursor dibuat dalam mode kamus dan

query dijalankan untuk mengambil data transaksi yang sesuai dengan `id_struk` yang diberikan. Data yang diambil kemudian diproses menjadi sebuah dictionary `data_struk` yang menyimpan informasi seperti ID transaksi, tanggal, kasir, detail item, total, pembayaran, dan kembalian. Metode ini juga menangani kesalahan dengan blok `except` yang mencetak pesan error jika ada masalah saat mengambil data dari database atau saat membuat PDF.

Setelah data struk berhasil diambil, file PDF dibuat dengan menggunakan modul `reportlab`. Header struk, informasi transaksi, dan detail item ditulis ke dalam PDF dengan menggunakan berbagai metode dari objek `canvas`. Total pembayaran, jumlah yang dibayarkan, dan kembalian juga ditampilkan di bagian bawah struk. Jika ada kesalahan selama pembuatan PDF, pesan error akan ditampilkan. Akhirnya, file PDF disimpan dengan nama yang sesuai dengan ID struk, dan pesan konfirmasi keberhasilan ditampilkan. Metode ini memastikan bahwa struk transaksi dapat diambil dari database dan disajikan dalam format PDF yang rapi dan profesional.

3.16 Membuat Class Produk

```
class Produk:
    def __init__(self):
        pass
```

Kelas `Produk` adalah template dasar untuk membuat dan mengelola informasi tentang produk dalam aplikasi. Konstruktor `__init__` pada kelas ini tidak memiliki inisialisasi khusus dan hanya berfungsi sebagai placeholder menggunakan `pass`. Ini memberikan struktur dasar yang memungkinkan penambahan metode lain di dalam kelas untuk mengatur berbagai fungsi terkait produk, seperti menambahkan, menghapus

3.17 Method untuk menampilkan seluruh produk.

```
@classmethod
def tampilkanproduk(cls):
    conn = koneksi()
    if conn is None:
        print("Koneksi ke database gagal.")
        return
    cur = conn.cursor()
    cur.execute("SELECT * FROM produk")
    data = cur.fetchall()
    conn.close()
    header = ["Kode Produk", "Nama Produk", "Jenis", "Harga"]
    # Mengakses elemen tuple menggunakan indeks
    dataUser = [[o[0], o[1], o[2], o[3]] for o in data]
    print(tabulate(dataUser, headers=header, tablefmt="grid"))
```

Metode `tampilkanproduk` adalah method class yang digunakan untuk menampilkan semua data produk dari database dalam bentuk tabel yang rapi. Pertama, koneksi ke database dibuat menggunakan fungsi `koneksi()`. Jika koneksi gagal, pesan error ditampilkan dan eksekusi dihentikan. Setelah koneksi berhasil, cursor dibuat dan query `SELECT * FROM produk` dijalankan untuk mengambil semua data produk. Hasil query disimpan dalam variabel `data`, koneksi ditutup, dan data diolah ke dalam list `dataUser` dengan mengakses elemen tuple menggunakan indeks. Akhirnya, data produk ditampilkan dalam bentuk tabel menggunakan fungsi `tabulate`.

3.18 Method Untuk Menambahkan Produk

```
@classmethod
def tambahproduk(cls):
    try:
        nama_produk = input("Masukkan nama produk: ")
        kategori = {
            "1": "Snack",
            "2": "Minuman",
            "3": "Makanan"
        }
        while True:
            print("Pilih kategori produk:")
            for key, value in kategori.items():
                print(f"{key}. {value}")
            han = input("Masukkan pilihan Anda: ")
            if han in kategori:
                jenis_produk = kategori[han]
            else:
                print("Pilihan tidak valid. Silakan pilih Kategori kembali.")
                continue

            hargastr = input("Masukkan Harga Produk Satuan: ")
            cek = hargastr.isdigit()
            if cek == True:
                harga = int(hargastr)
                conn = koneksi()
                cur = conn.cursor()
                cur.execute("INSERT INTO produk VALUES (%s,%s,%s,%s)", (cls.generate_id(), nama_produk,
                jenis_produk, harga))
                conn.commit()
                conn.close()
                print(f"produk {nama_produk}, berhasil ditambahkan")
                return True
            else:
                print("harga harus angka")
                continue
    except ValueError as e:
        print(f"Terjadi kesalahan: {e}")
```

Metode tambahproduk adalah method class yang digunakan untuk menambahkan produk baru ke dalam database. Pertama, pengguna diminta untuk memasukkan nama produk dan memilih kategori produk dari daftar yang diberikan (Snack, Minuman, Makanan). Jika kategori yang dipilih valid, pengguna kemudian diminta untuk memasukkan harga produk, yang harus berupa angka. Setelah input validasi selesai, koneksi ke database dibuat, dan data produk baru dimasukkan ke dalam tabel produk menggunakan query SQL. Jika harga yang dimasukkan bukan angka, pesan error akan ditampilkan dan pengguna diminta untuk mengulangi input. Jika terjadi kesalahan selama proses, pesan error dicetak untuk membantu debugging.

3.19 Method untuk Menghapus Produk

```
@classmethod
def hapusProduk(cls):
    try:
        cls.tampilkanproduk()
        produk = input("Masukkan kode produk yang ingin dihapus: ")
        conn = koneksi()
        cur = conn.cursor()
        cur.execute("SELECT * FROM produk WHERE id_produk = %s", (produk,))
        data = cur.fetchone()
        if data:
            cur.execute("DELETE FROM produk WHERE id_produk = %s", (produk,))
            conn.commit()
            conn.close()
            print(f"Produk dengan kode {produk} berhasil dihapus.")
            return True
        else:
            print(f"Produk dengan kode {produk} tidak ditemukan.")
            return False
    except mysql.connector.Error as e:
        #produk sudah pernah dibeli. untuk menjaga integritas data > maka tidak bisa dihapus
        print(f"Terjadi kesalahan: {e}")
```

Metode `hapusProduk` adalah method class yang digunakan untuk menghapus produk dari database berdasarkan kode produk yang diberikan oleh pengguna. Pertama, metode ini menampilkan semua produk yang ada dengan memanggil `cls.tampilkanproduk()`. Setelah itu, pengguna diminta untuk memasukkan kode produk yang ingin dihapus. Koneksi ke database dibuat, dan cursor digunakan untuk menjalankan query yang mencari produk dengan kode yang dimasukkan. Jika produk ditemukan, query `DELETE` dijalankan untuk menghapus produk tersebut dari database, perubahan dikomit, dan koneksi ditutup. Jika produk tidak ditemukan, pesan "Produk dengan kode tidak ditemukan" ditampilkan. Jika terjadi kesalahan karena produk sudah pernah dibeli (untuk menjaga integritas data), pesan error akan ditampilkan. Metode ini memastikan bahwa produk dapat dihapus dengan aman dari database jika tidak ada relasi yang menghalanginya.

3.20 Fungsi untuk tampilan menu utama

```
def tampilanMenuUtama():  
    print("\n=====")  
    print("          MENU UTAMA          ")  
    print("=====")  
    print("1. Tambah Pegawai")  
    print("2. Tambah Produk")  
    print("3. Tampilkan Seluruh Product")  
    print("4. Hapus Produk")  
    print("5. Tampilkan Semua Pegawai")  
    print("6. Hapus Pegawai")  
    print("7. Buat Transaksi Baru")  
    print("8. Tampilkan Riwayat Seluruh Struk Transaksi")  
    print("9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu")  
    print("10. Print Struk Transaksi dengan ID")  
    print("0. Exit")  
    print("=====")
```

3.21 Fungsi Utama yang digunakan untuk menjalankan beberapa fungsional program ketika user menginputkan pilihan

```
def menu():
    while True:
        # os.system('cls')
        tampilanMenuUtama()
        pilih = input("PILIH MENU: ")
        if pilih == "1":
            Pegawai.tambahpegawai()
        elif pilih == "2":
            Produk.tambahproduk()
            os.system('pause')
        elif pilih == "3":
            Produk.tampilkanproduk()
            os.system('pause')
        elif pilih == "4":
            Produk.hapusProduk()
            os.system('pause')
        elif pilih == "5":
            Pegawai.tampilsemuapegawai()
            os.system('pause')
        elif pilih == "6":
            Pegawai.hapuspegawai()
            os.system('pause')
        elif pilih == "7":
            Pegawai.tambahtransaksi()
        elif pilih == "8":
            Struk.tampilsemuastruk()
        elif pilih == "9":
            Struk.tampilstrukbywaktu()
        elif pilih == "10":
            Struk.tampilsemuastruk()
            id_struk = input("Masukkan ID Struk: ")
            Struk.buat_struk_pdf_dari_db(id_struk)
        elif pilih == "0":
            print("Anda Keluar Program")
            return True
        else:
            print("Ulang ya")
            os.system('pause')
```

3.22 Kode untuk menjalankan fungsi menu utama program.

```
if __name__ == "__main__":  
    menu()
```

4. IMPLEMENTASI ATAU HASIL RUNNING

4.1 Menambahkan Pegawai

-Masuk Menu Utama

```
=====
                MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: █
```

-Masuk menu tambah pegawai dengan input 1

-Masukkan data yang diperlukan , seperti NIK (harus angka), nama pegawai dan alamat.

```
=====
PILIH MENU: 1
Masukkan NIK pegawai: 123321
Masukkan Nama pegawai: Megumin Bakudan
Masukkan alamat pegawai: Axell, Mamono City, Yogyakarta █
```

-Tekan Enter, jika berhasil dan data yang dimasukkan valid maka akan muncul pemberitahuan.

```
PEGAWAI BERHASIL DITAMBAHKAN...
Press any key to continue . . . █
```

4.2 Menambahkan Produk.

-Masuk ke menu utama

-Masuk ke menu tambah produk

```
=====
                        MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 2
```

-Masukkan data yang diperlukan

-Ketika semua data valid, maka akan muncul pemberitahuan sukses:

```
=====
PILIH MENU: 2
Masukkan nama produk: Gudeg Krecek
Pilih kategori produk:
1. Snack
2. Minuman
3. Makanan
Masukkan pilihan Anda: 3
Masukkan Harga Produk Satuan: 13000
produk Gudeg Krecek, berhasil ditambahkan
Press any key to continue . . .
```

4.3 Menampilkan Seluruh Product.

-Masuk ke menu utama

-Masuk ke menu Tampilkan seluruh product

```
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 3
```

-Maka akan tampil beberapa product yang telah tersimpan

```
=====
PILIH MENU: 3
+-----+-----+-----+-----+
| Kode Produk | Nama Produk      | Jenis  | Harga |
+=====+=====+=====+=====+
| DToL        | Es Teh Pengajian | Minuman | 5000  |
+-----+-----+-----+-----+
| un0a        | Gudeg Krecek     | Makanan | 13000 |
+-----+-----+-----+-----+
| v8eK        | Ikan Bakar + Nasi | Makanan | 20000 |
+-----+-----+-----+-----+
Press any key to continue . . .
```

4.4 Menghapus Produk

-Masuk Ke menu utama

-Masuk ke menu Hapus Produk

```
=====
                        MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 4
```

-Masukkan Kode produk yang ingin dihapus

```
=====
PILIH MENU: 4
+-----+-----+-----+-----+
| Kode Produk | Nama Produk          | Jenis  | Harga |
+-----+-----+-----+-----+
| 8EMo        | Telur Asin dengan Garam Doa | Makanan | 20000 |
+-----+-----+-----+-----+
| ARWv        | Nasi Padang Paru masak Rendang | Makanan | 16000 |
+-----+-----+-----+-----+
| CDIB        | Yakiniku              | Makanan | 20000 |
+-----+-----+-----+-----+
| DToL        | Es Teh Pengajian      | Minuman | 5000  |
+-----+-----+-----+-----+
| Jktv        | Takoyaki Crabs Stick + Cumi | Snack  | 23000 |
+-----+-----+-----+-----+
| N018        | Es Kopi Aren          | Minuman | 8000  |
+-----+-----+-----+-----+
| o8YW        | Takoyaki              | Snack  | 13000 |
+-----+-----+-----+-----+
| unOa        | Gudeg Krecek          | Makanan | 13000 |
+-----+-----+-----+-----+
| v8eK        | Ikan Bakar + Nasi     | Makanan | 20000 |
+-----+-----+-----+-----+
Masukkan kode produk yang ingin dihapus: 8EMo
```


-Jika berhasil dihapus, maka akan muncul pemberitahuan

```
+-----+
Masukkan kode produk yang ingin dihapus: 8EMo
Produk dengan kode 8EMo berhasil dihapus.
Press any key to continue . . .
```

**NB: TIDAK BISA MENGHAPUS PRODUK YANG SUDAH PERNAH
DIBELI CUSTOMER / YANG BERELASI DENGAN TABEL
LAINNYA.**

```
Masukkan kode produk yang ingin dihapus: v8ek
Terjadi kesalahan: 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails ('penjualan`.`struk`, CONSTRAINT `fk_id_produk` FOREIGN K
(`id_produk`) REFERENCES `produk` (`id_produk`))
Press any key to continue . . .
```

4.5 Menampilkan Semua Pegawai

-Masuk ke menu utama

-Masuk ke menu Tampil Semua Pegawai

```
=====
                        MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 5
```

-Maka akan tampil seluruh daftar pegawai

```
=====
PILIH MENU: 5
+-----+-----+-----+
|          NIK | Nama Pegawai   | Alamat Pegawai |
+-----+-----+-----+
|          12345 | ryu           | wates, kulon progo |
+-----+-----+-----+
|          123321 | Megumin Bakudan | Axell, Mamono City, Yogyakarta |
+-----+-----+-----+
|          123450 | Irfan Baihaqi  | pengasih, Kulon Progo, Yogyakarta |
+-----+-----+-----+
|          543210 | Himmel san     | Boukensa city, Liyue, Saranjana |
+-----+-----+-----+
|          3401123 | Cihuy         | Cihuy , Yogyakarta |
+-----+-----+-----+
| 3401022008097771 | IRfan si Ambis | Wates, Kulon Progo , Yogyakarta |
+-----+-----+-----+
Press any key to continue . . .
```

4.6 Hapus Pegawai

-Masuk ke menu utama

-Masuk ke menu hapus pegawai

```
=====
                        MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 6
```

-Maka akan muncul daftar pegawai, masukkan NIK Pegawai yang akan dihapus

NIK	Nama Pegawai	Alamat Pegawai
10101	Agus Sosuke	Hollow Castle, Isekai
12345	ryu	wates, kulon progo
123321	Megumin Bakudan	Axell, Mamono City, Yogyakarta
123450	Irfan Baihaqi	pengasih, Kulon Progo, Yogyakarta
543210	Himmel san	Boukensa city, Liyue, Saranjana
3401123	Cihuy	Cihuy , Yogyakarta
3401022008097771	Irfan si Ambis	Wates, Kulon Progo , Yogyakarta

Masukkan NIK pegawai yang akan dihapus: 10101

-Jika berhasil maka akan muncul pemberitahuan

```
Masukkan NIK pegawai yang akan dihapus: 10101
Pegawai berhasil dihapus.
Press any key to continue . . .
```

NB: TIDAK BISA MENGHAPUS PEGAWAI YANG SUDAH PERNAH MELAKUKAN TRANSAKSI.

```
Masukkan NIK pegawai yang akan dihapus: 3401123
Error saat menghapus data pegawai: 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails ('penjualan','struk', CONSTRAINT 'fk_ni
gawai' FOREIGN KEY ('nik_pegawai') REFERENCES 'pegawai' ('nik_pegawai'))
Press any key to continue . . .
```

4.7 Membuat Transaksi Baru

-Masuk ke menu utama

-Masuk ke menu Buat Transaksi Baru

```

=====
                        MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 7

```

- Maka akan muncul nama pegawai yang akan melayani pembeli
- Masukkan nama pegawai yang sedang melayani sekarang untuk pembelian dan customer kali ini.

```

+-----+-----+-----+
|      NIK | Nama Pegawai | Alamat Pegawai |
+-----+-----+-----+
|    123321 | Megumin Bakudan | Axell, Mamono City, Yogyakarta |
+-----+-----+-----+
|    123450 | Irfan Baihaqi   | pengasih, Kulon Progo, Yogyakarta |
+-----+-----+-----+
|    543210 | Himmel san      | Boukensa city, Liyue, Saranjana |
+-----+-----+-----+
|    3401123 | Cihuy           | Cihuy , Yogyakarta |
+-----+-----+-----+
| 3401022008097771 | IRfan si Ambis | Wates, Kulon Progo , Yogyakarta |
+-----+-----+-----+
Masukkan kode pegawai yang membuat transaksi: 543210

```

- Maka akan muncul tabel daftar nama produk
- Masukkan id produk dari produk yang dibeli pembeli

Kode Produk	Nama Produk	Jenis	Harga
ARWv	Nasi Padang Paru masak Rendang	Makanan	16000
CDIB	Yakiniku	Makanan	20000
DTOL	Es Teh Pengajian	Minuman	5000
Jktv	Takoyaki Crabs Stick + Cumi	Snack	23000
N018	Es Kopi Aren	Minuman	8000
o8YW	Takoyaki	Snack	13000
un0a	Gudeg Krecek	Makanan	13000
v8eK	Ikan Bakar + Nasi	Makanan	20000

SILAHKAN MASUKKAN KODE PRODUK: un0a

-Masukkan data-data lain. Untuk detail transaksi untuk produk pada pembelian ini sifatnya opsional. Masukkan Y , jika ingin menambahkan catatan sebagai detail transaksi.

```
SILAHKAN MASUKKAN KODE PRODUK: un0a
Masukkan jumlah produk yang dibeli: 2
Apakah ingin Masukkan catatan untuk transaksi barang ini? (y/n): y
Masukkan catatan: Gudeg nya dibanyakin.
```

-Akan muncul pertanyaan , apakah ingin lanjut untuk melakukan transaksi / menambahkan barang lagi untuk pembelian tersebut.

-Jika ingin menambahkan lagi , inputkan “n”

```
SILAHKAN MASUKKAN KODE PRODUK: un0a
Masukkan jumlah produk yang dibeli: 2
Apakah ingin Masukkan catatan untuk transaksi barang ini? (y/n): y
Masukkan catatan: Gudeg nya dibanyakin.
Apakah TRANSAKSI SUDAH SELESAI (y/n)?: y
```

Maka akan bisa menambahkan produk beberapa kali untuk pembelian ini.

```
+-----+
SILAHKAN MASUKKAN KODE PRODUK: n018
Masukkan jumlah produk yang dibeli: 2
Apakah ingin Masukkan catatan untuk transaksi barang ini? (y/n): n
Apakah TRANSAKSI SUDAH SELESAI (y/n)? : n
```

```
+-----+
SILAHKAN MASUKKAN KODE PRODUK: cdib
Masukkan jumlah produk yang dibeli: 1
Apakah ingin Masukkan catatan untuk transaksi barang ini? (y/n): y
Masukkan catatan: pedas ya , banyakin cabainya
Apakah TRANSAKSI SUDAH SELESAI (y/n)? : y
```

Jika sudah cukup untuk transaksi atau menambahkan barang untuk pembelian kali ini, bisa inputkan “y” saat muncul pertanyaan nya Kembali.

```
Apakah TRANSAKSI SUDAH SELESAI (y/n)? : y
```

-Pilih Metode pembayaran(1-6)

-Masukkan pembeli membayar dengan uang berapa pecahan

```
Apakah TRANSAKSI SUDAH SELESAI (y/n)? : y
1. TUNAI
2. DANA
3. OVO
4. GOPAY
5. LINKAJA
6. SHOPEEPAY
Pilih metode pembayaran (1-6): 1
Masukkan jumlah uang yang dibayarkan: 200000
```

-Jika uang yang dibayarkan mencukupi , maka akan muncul pemberitahuan:

```
Masukkan jumlah uang yang dibayarkan: 200000
Pembayaran berhasil.
Press any key to continue . . .
```

-Press Any Key

-Maka secara otomatis akan didapatkan sebuah struk berbentuk file pdf, seperti di bawah ini:

Warung Bebas Galau

Jl. Kebahagiaan No. 42, Yogyakarta

Telp: 0812-3456-7890

ID Transaksi: jB7O

Tanggal : 2024-12-12 20:34:12

Kasir : Himmel san

Item	Qty	Harga Satuan	Subtotal
Yakiniku	1	Rp 20,000	Rp 20,000
Es Kopi Aren	2	Rp 8,000	Rp 16,000
Gudeg Krecek	2	Rp 13,000	Rp 26,000

Total:

Rp 62,000

Pembayaran:

Rp 200,000

Kembalian:

Rp 138,000

Terima kasih telah berbelanja di Warung Bebas Galau

4.8 Menampilkan seluruh Riwayat transaksi

-Masuk ke menu utama

-Masuk ke menu Tampilkan riwayat seluruh transaksi

-Maka akan muncul seluruh Riwayat transaksi

ID Struk	ID Transaksi	Nama Produk	Nama Pegawai	Jumlah Produk	Harga Produk	Total Harga	Waktu Transaksi
15pt	xjXy	Ikan Bakar + Nasi	Cihuy	2	20000	40000	2024-12-12 11:08:18
1j0Y	vpSV	Es Teh Pengajian	Cihuy	1	5000	5000	2024-12-12 12:39:50
6Giu	0qcX	Es Teh Pengajian	Cihuy	1	5000	5000	2024-12-12 12:38:03
6Giu	DchV	Ikan Bakar + Nasi	Cihuy	4	20000	80000	2024-12-12 12:37:46
gv12	SA0E	Ikan Bakar + Nasi	Cihuy	3	20000	60000	2024-12-12 11:39:32
gwnD	1NjE	Ikan Bakar + Nasi	Cihuy	4	20000	80000	2024-12-12 11:44:12
gwnD	PcP9	Es Teh Pengajian	Cihuy	1	5000	5000	2024-12-12 11:44:32
ojaH	3RpW	Ikan Bakar + Nasi	Cihuy	2	20000	40000	2024-12-12 11:27:55
owBQ	1w7g	Ikan Bakar + Nasi	Cihuy	2	20000	40000	2024-12-12 11:35:41
uokK	s9zN	Ikan Bakar + Nasi	Cihuy	4	20000	80000	2024-12-12 11:30:47
wDny	jB7O	Yakiniku	Himmel san	1	20000	20000	2024-12-12 20:34:12
wDny	lieh	Es Kopi Aren	Himmel san	2	8000	16000	2024-12-12 20:33:48
wDny	m00T	Gudeg Krecek	Himmel san	2	13000	26000	2024-12-12 20:30:59
ZngV	3Vt5	Ikan Bakar + Nasi	Cihuy	6	20000	120000	2024-12-12 11:41:59
ZngV	B0yk	Ikan Bakar + Nasi	Cihuy	3	20000	60000	2024-12-12 11:41:30

Press any key to continue . . .

4.9 Menampilkan seluruh Riwayat transaksi di Tanggal Tertentu.

-Masuk ke menu utama

-Masuk ke menu Tampilkan Riwayat Struk Transaksi pada waktu tertentu

-Masukkan tanggal dengan formal yyyy-mm-dd

```
=====
                        MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 9
Masukkan tanggal (YYYY-MM-DD): 2024-12-12
```

-Maka akan muncul daftar seluruh transaksi barang di tanggal tersebut

Masukkan tanggal (YYYY-MM-DD): 2024-12-12							
ID Struk	ID Transaksi	Nama Produk	Nama Pegawai	Jumlah Produk	Harga Produk	Total Harga	Waktu Transaksi
15pt	xjXy	Ikan Bakar + Nasi	Cihuy	2	20000	40000	2024-12-12 11:08:18
1jQY	vpSV	Es Teh Pengajian	Cihuy	1	5000	5000	2024-12-12 12:39:50
66iu	0qcX	Es Teh Pengajian	Cihuy	1	5000	5000	2024-12-12 12:38:03
66iu	DchV	Ikan Bakar + Nasi	Cihuy	4	20000	80000	2024-12-12 12:37:46
gv12	SA0E	Ikan Bakar + Nasi	Cihuy	3	20000	60000	2024-12-12 11:39:32
gwnD	1NjE	Ikan Bakar + Nasi	Cihuy	4	20000	80000	2024-12-12 11:44:12
gwnD	PcP9	Es Teh Pengajian	Cihuy	1	5000	5000	2024-12-12 11:44:32
ojaH	3RpW	Ikan Bakar + Nasi	Cihuy	2	20000	40000	2024-12-12 11:27:55
owBQ	iW7g	Ikan Bakar + Nasi	Cihuy	2	20000	40000	2024-12-12 11:35:41
uokK	s9zN	Ikan Bakar + Nasi	Cihuy	4	20000	80000	2024-12-12 11:30:47
wDny	jB7O	Yakiniku	Himmel san	1	20000	20000	2024-12-12 20:34:12
wDny	li6h	Es Kopi Aren	Himmel san	2	8000	16000	2024-12-12 20:33:48
wDny	m00T	Gudeg Krecek	Himmel san	2	13000	26000	2024-12-12 20:30:59
ZngV	3Vt5	Ikan Bakar + Nasi	Cihuy	6	20000	120000	2024-12-12 11:41:59
ZngV	B0yk	Ikan Bakar + Nasi	Cihuy	3	20000	60000	2024-12-12 11:41:30

Press any key to continue . . .

4.10 Print 1 Struk tertentu dengan id Struk.

-Masuk ke menu utama

-Masuk ke menu Print Struk Transaksi dengan ID

```
=====
                        MENU UTAMA
=====
1. Tambah Pegawai
2. Tambah Produk
3. Tampilkan Seluruh Product
4. Hapus Produk
5. Tampilkan Semua Pegawai
6. Hapus Pegawai
7. Buat Transaksi Baru
8. Tampilkan Riwayat Seluruh Struk Transaksi
9. Tampilkan Riwayat Struk Transaksi pada waktu tertentu
10. Print Struk Transaksi dengan ID
0. Exit
=====
PILIH MENU: 10
+-----+-----+-----+-----+
```

-Akan ditampilkan terlebih dahulu daftar seluruh Riwayat transaksi barang.

-Masukkan ID STRUK dari struk yang ingin diprint atau dibuatkan file pdf nya

```
Press any key to continue .
Masukkan ID Struk: zngv
```

-Maka akan otomatis dibuatkan file pdf nya

Warung Bebas Galau

Jl. Kebahagiaan No. 42, Yogyakarta

Telp: 0812-3456-7890

ID Transaksi: ZngV
Tanggal : 2024-12-12 11:41:59
Kasir : Cihuy

Item	Qty	Harga Satuan	Subtotal
Ikan Bakar + Nasi	6	Rp 20,000	Rp 120,000
Ikan Bakar + Nasi	3	Rp 20,000	Rp 60,000
Total:			Rp 180,000
Pembayaran:			Rp 10,000,000
Kembalian:			Rp 9,820,000

Terima kasih telah berbelanja di Warung Bebas Galau