



Prepared by group 11

Proyek Kelas

Perancangan sistem dan prototipe
aplikasi sederhana pengelola restoran
“Restorify”



Team Members



**Radithya
Guntoro Adhi**



**Ahmad
Hasyir Bastari**



**Chindy
Herpati**



Deskripsi Sistem



Restorify adalah aplikasi perangkat lunak yang dirancang untuk **mengelola operasional restoran** secara efisien. Sistem ini menyediakan alat bantu bagi **manajer** restoran untuk mengelola berbagai **aspek operasional** seperti pengelolaan **bahan baku**, manajemen **karyawan**, transaksi penjualan, hingga **laporan keuangan** dan hubungan dengan **supplier**.



Tujuan

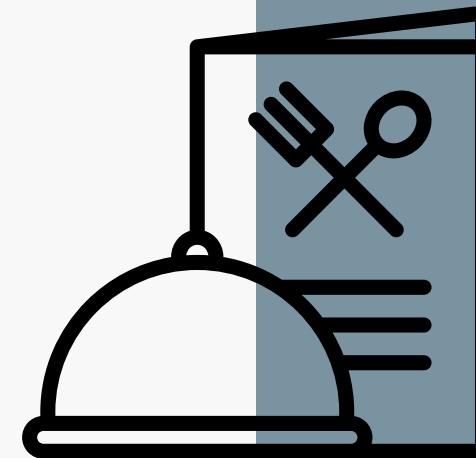
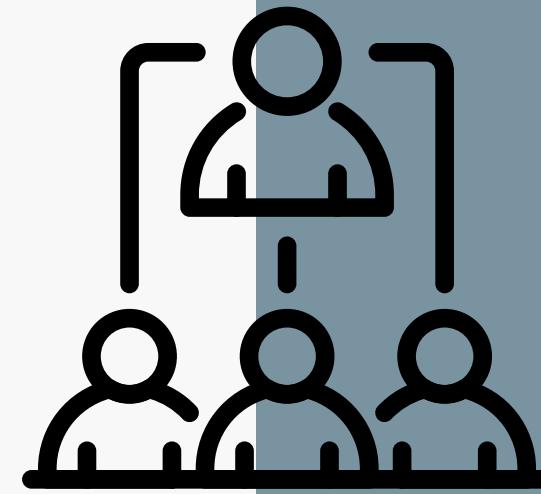
- **Efisiensi Operasional:** Pengelolaan bahan baku, supplier, karyawan, dan transaksi menjadi **lebih efisien**.
- **Pengelolaan Terpadu:** Semua data yang diperlukan restoran (bahan baku, menu, karyawan, transaksi) terintegrasi dalam satu sistem yang **mudah diakses**.
- **Laporan Real-Time:** Manajer restoran dapat dengan mudah **memantau kinerja** restoran dan **membuat keputusan** berdasarkan laporan yang dihasilkan sistem.



Fitur Utama Aplikasi

- Pengelolaan Bahan Baku
- Pengelolaan Supplier
- Pengelolaan Karyawan
- Pengelolaan Menu
- Pengelolaan Pelanggan
- Pengelolaan Transaksi dan Keuangan

.....

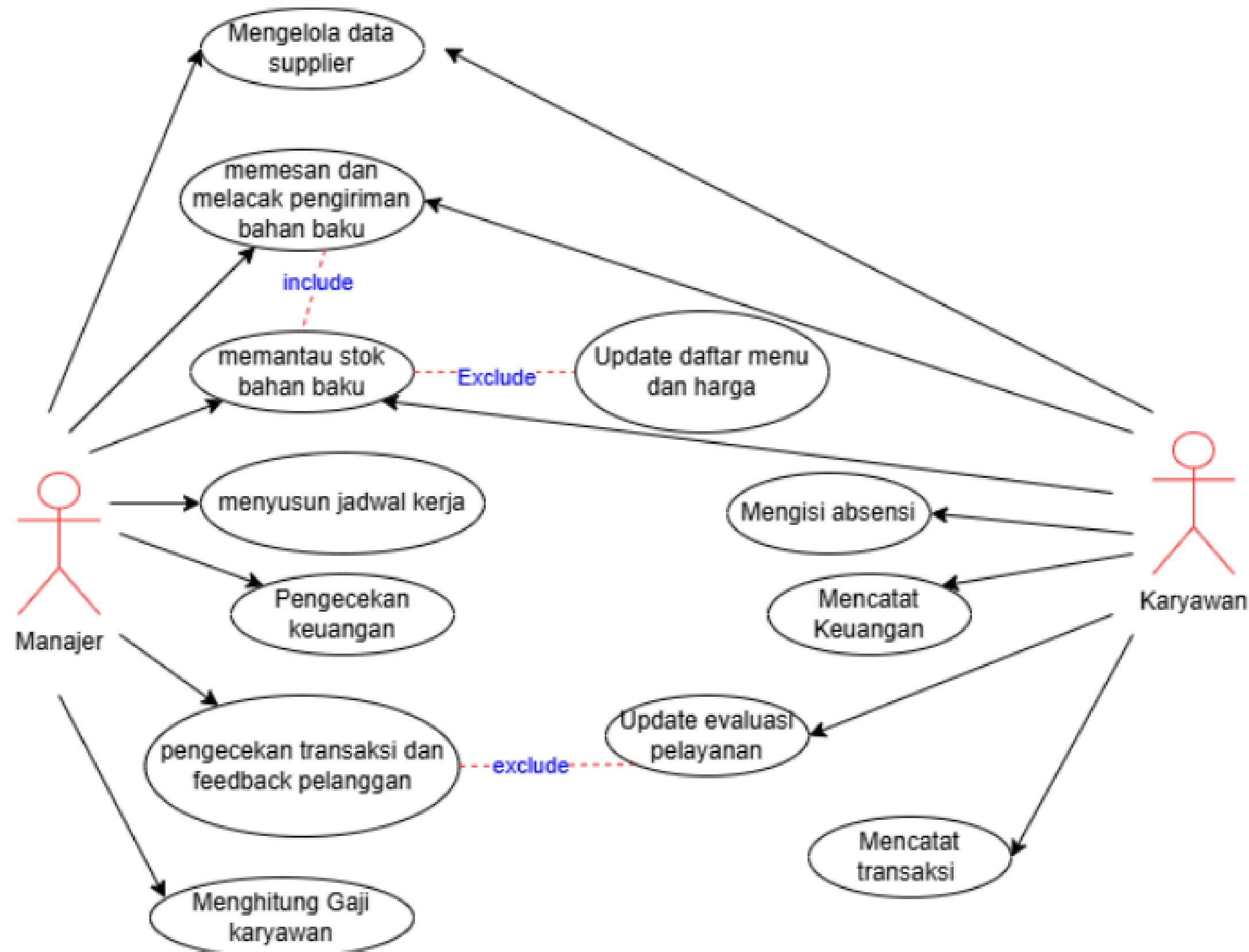


Fungsi fitur aplikasi

- **Manajemen Supplier:** Mengelola data supplier, memesan bahan baku, melacak pengiriman dan pembayaran.
- **Manajemen Bahan Baku:** Memantau stok bahan baku dan memesan ulang saat stok menipis.
- **Manajemen Karyawan:** Menyusun jadwal kerja karyawan, mencatat absensi, dan menghitung gaji.
- **Pengelolaan Menu:** Mengupdate daftar menu dan harga sesuai dengan bahan baku yang tersedia.
- **Pengelolaan Keuangan:** Mencatat transaksi penjualan dan menghitung pendapatan dan pengeluaran restoran.
- **Manajemen Pelanggan:** Menyimpan riwayat transaksi pelanggan untuk meningkatkan pelayanan.



Use case



Block Diagram



Entity relationship diagram

• • • •

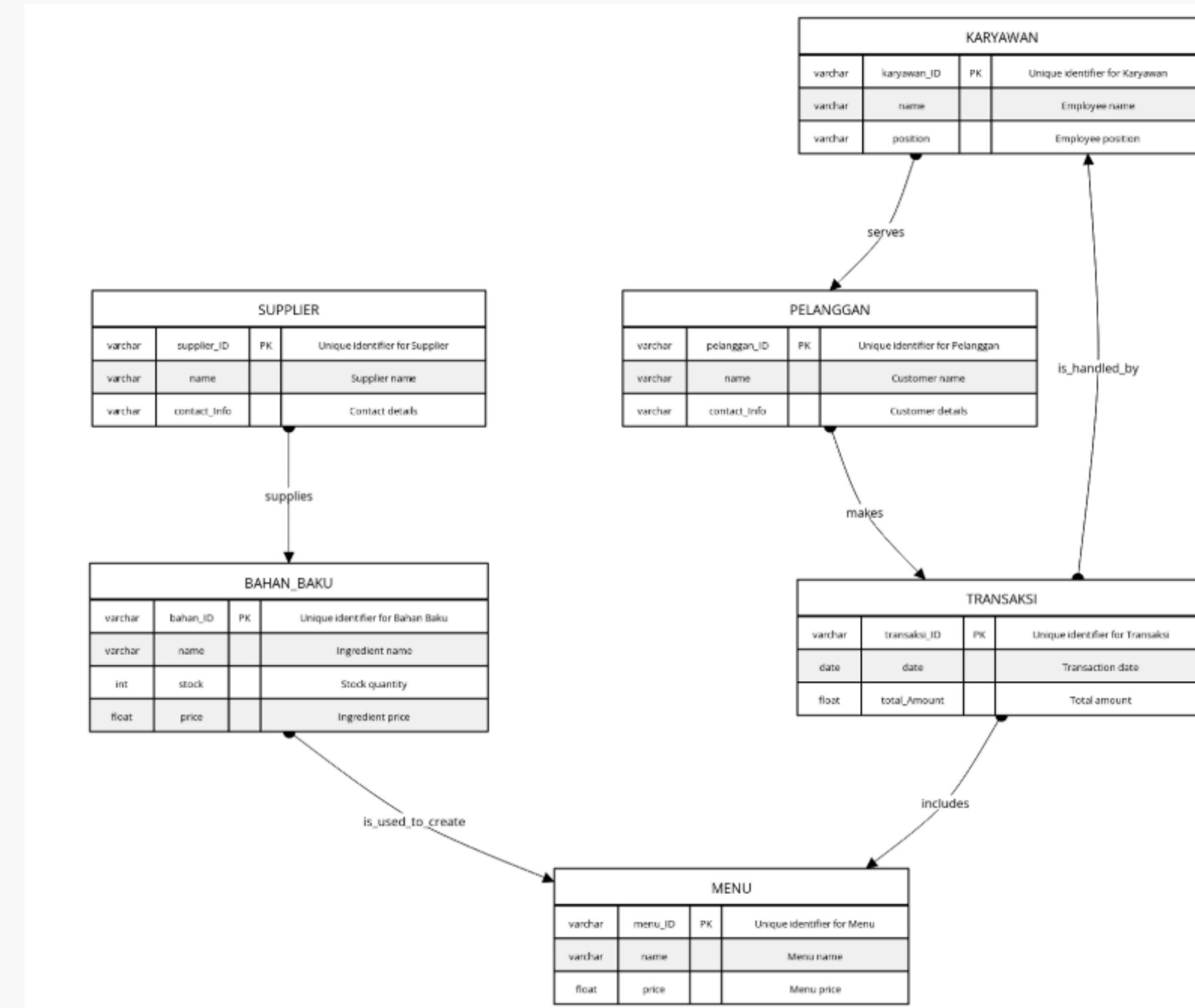
Entitas dalam ERD:

- **Supplier:** Informasi tentang pemasok bahan baku (nama, kontak, bahan baku yang disuplai, harga, dll).
- **Bahan Baku:** Mengelola bahan baku yang ada di restoran, termasuk stok, penggunaan, dan pemasok.
- **Karyawan:** Data karyawan, jadwal kerja, absensi, dan gaji.
- **Menu:** Daftar menu, harga, dan bahan baku yang dibutuhkan.
- **Transaksi:** Mencatat setiap transaksi penjualan yang terjadi di restoran.
- **Pelanggan:** Menyimpan data pelanggan dan riwayat transaksi mereka.

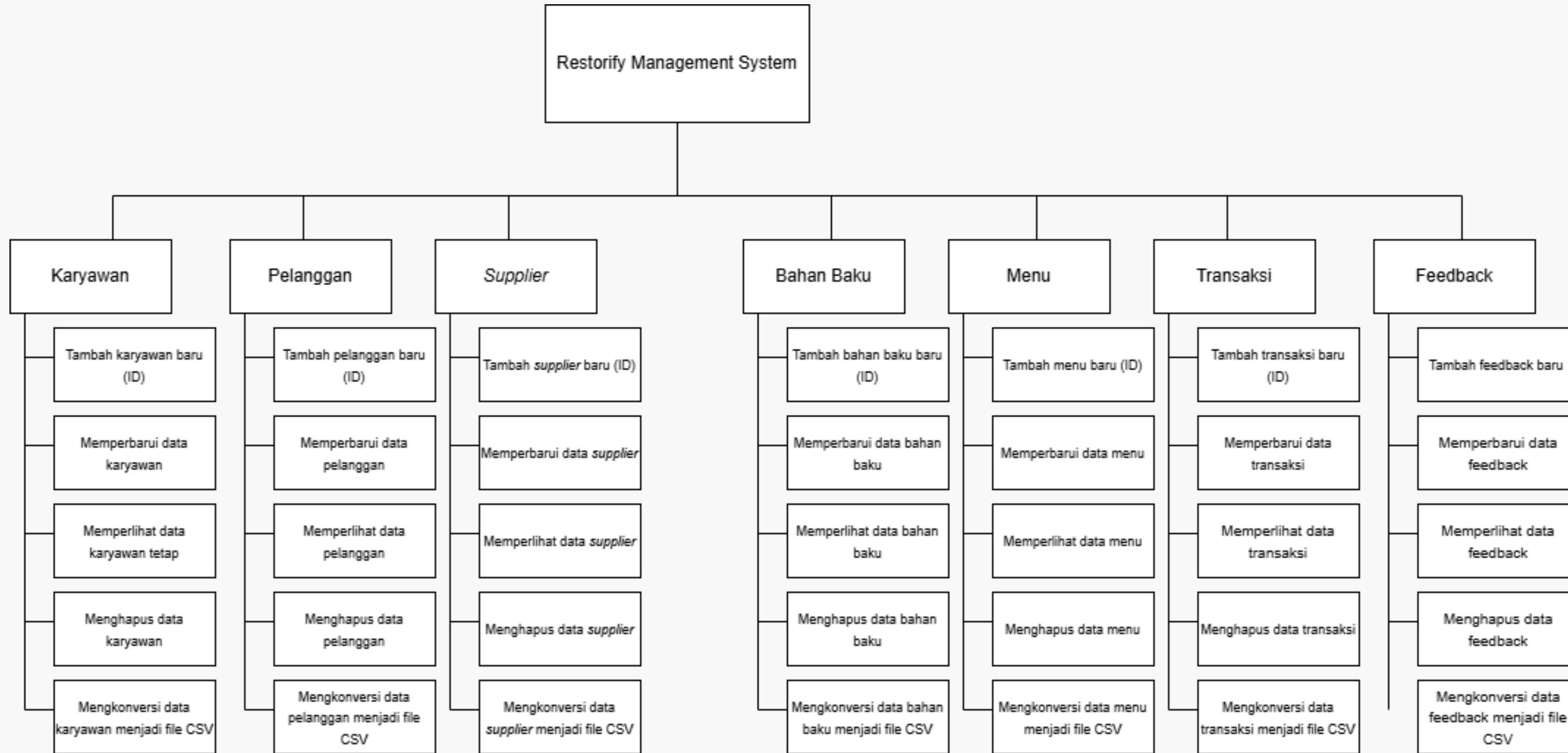
• • • •

Entity relationship diagram

• • • •



Decomposition Diagram



SQL Database

```
1 -- Tabel karyawan
2 CREATE TABLE IF NOT EXISTS karyawan (
3     karyawan_id VARCHAR(5) PRIMARY KEY,
4     employee_name VARCHAR(50) NOT NULL,
5     position VARCHAR(25) NOT NULL
6 );
7
8 INSERT INTO karyawan (karyawan_id, employee_name, position) VALUES
9     ('K001', 'Adi', 'Waiter'),
10    ('K002', 'Agus', 'Waiter'),
11    ('K003', 'Sinta', 'Cashier'),
12    ('K004', 'Budi', 'Manajer'),
13    ('K005', 'Bryan', 'Chef'),
14    ('K006', 'Dian', 'Operational');
15
16 -- Tabel pelanggan
17 CREATE TABLE IF NOT EXISTS pelanggan (
18     pelanggan_id VARCHAR(5) PRIMARY KEY,
19     cus_name VARCHAR(50) NOT NULL,
20     contact_info VARCHAR(15) NOT NULL
21 );
22
23 INSERT INTO pelanggan (pelanggan_id, cus_name, contact_info) VALUES
24     ('P001', 'Dini', '0823416777'),
25     ('P002', 'Anita', '0823452267'),
26     ('P003', 'Daffa', '0890867736'),
27     ('P004', 'Tiara', '083342156'),
28     ('P005', 'Torik', '0876544325'),
29     ('P006', 'Aziz', '087665489'),
30     ('P007', 'Ayu', '087655394'),
31     ('P008', 'Dedi', '0823416657');
```

```
-- Tabel supplier
CREATE TABLE IF NOT EXISTS supplier (
    supplier_id VARCHAR(5) PRIMARY KEY,
    supplier_name VARCHAR(50) NOT NULL,
    address VARCHAR(100) NOT NULL
);

INSERT INTO supplier (supplier_id, supplier_name, address) VALUES
    ('S001', 'Indofood', 'Jakarta'),
    ('S002', 'Ayla Farm', 'Bandung'),
    ('S003', 'FoodFresh', 'Tangerang');

-- Tabel bahan_baku
CREATE TABLE IF NOT EXISTS bahan_baku (
    bahan_id VARCHAR(5) PRIMARY KEY,
    nama_bahan VARCHAR(50) NOT NULL,
    stock INT NOT NULL CHECK (stock >= 0),
    satuan VARCHAR(20) NOT NULL,
    harga_bahan DECIMAL(15,2) NOT NULL,
    supplier_id VARCHAR(5),
    FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
);

INSERT INTO bahan_baku (bahan_id, nama_bahan, stock, satuan, harga_bahan, supplier_id) VALUES
    ('BB1', 'Ayam Potong', 50, 'Kg', 500000.00, 'S001'),
    ('BB2', 'Sayuran', 30, 'Kg', 200000.00, 'S002'),
    ('BB3', 'Ikan Lele', 20, 'Kg', 350000.00, 'S003'),
    ('BB4', 'Bawang', 50, 'Kg', 250000.00, 'S001'),
    ('BB5', 'Bumbu', 100, 'Bungkus', 600000.00, 'S001');
```

SQL Database

```
63 -- Tabel menu
64 CREATE TABLE IF NOT EXISTS menu (
65     menu_id VARCHAR(5) PRIMARY KEY,
66     nama_menu VARCHAR(50) NOT NULL,
67     harga DECIMAL(15,2) NOT NULL CHECK (harga >= 0)
68 );
69
70 INSERT INTO menu (menu_id, nama_menu, harga) VALUES
71     ('MN1', 'Ayam Bakar', 20000.00),
72     ('MN2', 'Nasi Goreng', 15000.00),
73     ('MN3', 'Oseng Kangkung', 10000.00),
74     ('MN4', 'Tumis Labu', 12000.00),
75     ('MN5', 'Es Krim', 10000.00),
76     ('MN6', 'Es Jeruk', 6000.00),
77     ('MN7', 'Es Teh Manis', 5000.00);
78
79 -- Tabel transaksi
80 CREATE TABLE IF NOT EXISTS transaksi (
81     transaksi_id VARCHAR(5) PRIMARY KEY,
82     tanggal_pembelian DATE NOT NULL,
83     pelanggan_id VARCHAR(5) NOT NULL,
84     karyawan_id VARCHAR(5) NOT NULL,
85     total_transaksi DECIMAL(15,2) NOT NULL CHECK (total_transaksi >= 0),
86     FOREIGN KEY (pelanggan_id) REFERENCES pelanggan(pelanggan_id),
87     FOREIGN KEY (karyawan_id) REFERENCES karyawan(karyawan_id)
88 );
89
90 INSERT INTO transaksi (transaksi_id, tanggal_pembelian, pelanggan_id, karyawan_id, total_transaksi) VALUES
91     ('T001', '2024-12-01', 'P001', 'K003', 170000.00),
92     ('T002', '2024-12-03', 'P002', 'K003', 250000.00),
93     ('T003', '2024-12-05', 'P003', 'K003', 150000.00),
94     ('T004', '2024-12-04', 'P004', 'K003', 200000.00),
95     ('T005', '2024-12-06', 'P005', 'K003', 280000.00),
96     ('T006', '2024-12-07', 'P006', 'K003', 500000.00);
--
```

```
98 -- Tabel detail_transaksi
99 CREATE TABLE IF NOT EXISTS detail_transaksi (
100    detail_id INTEGER PRIMARY KEY AUTOINCREMENT,
101    transaksi_id VARCHAR(5) NOT NULL,
102    menu_id VARCHAR(5) NOT NULL,
103    jumlah INT NOT NULL CHECK (jumlah > 0),
104    harga DECIMAL(15,2) NOT NULL CHECK (harga >= 0),
105    FOREIGN KEY (transaksi_id) REFERENCES transaksi(transaksi_id),
106    FOREIGN KEY (menu_id) REFERENCES menu(menu_id)
107 );
108
109 INSERT INTO detail_transaksi (transaksi_id, menu_id, jumlah, harga) VALUES
110     ('T001', 'MN1', 2, 20000.00),
111     ('T001', 'MN6', 2, 6000.00),
112     ('T002', 'MN2', 2, 15000.00),
113     ('T002', 'MN6', 3, 6000.00),
114     ('T003', 'MN3', 1, 10000.00),
115     ('T003', 'MN7', 2, 5000.00),
116     ('T004', 'MN4', 2, 12000.00),
117     ('T004', 'MN7', 1, 5000.00),
118     ('T005', 'MN1', 3, 20000.00),
119     ('T005', 'MN5', 1, 10000.00),
120     ('T006', 'MN2', 4, 15000.00),
121     ('T006', 'MN6', 2, 6000.00);
122
123 -- Tabel komposisi_menu
124 CREATE TABLE IF NOT EXISTS komposisi_menu (
125     menu_id VARCHAR(5) NOT NULL,
126     bahan_id VARCHAR(5) NOT NULL,
127     jumlah_bahan INT NOT NULL CHECK (jumlah_bahan > 0),
128     PRIMARY KEY (menu_id, bahan_id),
129     FOREIGN KEY (menu_id) REFERENCES menu(menu_id),
130     FOREIGN KEY (bahan_id) REFERENCES bahan_baku(bahan_id)
131 );
```

SQL Database

```
133 INSERT INTO komposisi_menu (menu_id, bahan_id, jumlah_bahan) VALUES
134     ('MN1', 'BB1', 1),
135     ('MN1', 'BB5', 1),
136     ('MN2', 'BB2', 1),
137     ('MN2', 'BB5', 1),
138     ('MN3', 'BB2', 1),
139     ('MN3', 'BB5', 1),
140     ('MN4', 'BB2', 1),
141     ('MN4', 'BB5', 1),
142     ('MN5', 'BB5', 1),
143     ('MN6', 'BB5', 1),
144     ('MN7', 'BB4', 1);
145
146 -- Tabel jadwal_kerja
147 CREATE TABLE IF NOT EXISTS jadwal_kerja (
148     jadwal_id INTEGER PRIMARY KEY AUTOINCREMENT,
149     karyawan_id VARCHAR(5) NOT NULL,
150     tanggal DATE NOT NULL,
151     shift VARCHAR(10) NOT NULL,
152     FOREIGN KEY (karyawan_id) REFERENCES karyawan(karyawan_id)
153 );
154
155 INSERT INTO jadwal_kerja (karyawan_id, tanggal, shift) VALUES
156     ('K001', '2024-12-01', 'Pagi'),
157     ('K002', '2024-12-01', 'Siang'),
158     ('K003', '2024-12-01', 'Malam'),
159     ('K001', '2024-12-02', 'Siang'),
160     ('K002', '2024-12-02', 'Pagi'),
161     ('K003', '2024-12-02', 'Malam');
```

```
163 -- Tabel absensi
164 CREATE TABLE IF NOT EXISTS absensi (
165     absensi_id INTEGER PRIMARY KEY AUTOINCREMENT,
166     karyawan_id VARCHAR(5) NOT NULL,
167     tanggal DATE NOT NULL,
168     status VARCHAR(10) NOT NULL,
169     FOREIGN KEY (karyawan_id) REFERENCES karyawan(karyawan_id)
170 );
171
172 INSERT INTO absensi (karyawan_id, tanggal, status) VALUES
173     ('K001', '2024-12-01', 'Hadir'),
174     ('K002', '2024-12-01', 'Hadir'),
175     ('K003', '2024-12-01', 'Hadir'),
176     ('K001', '2024-12-02', 'Sakit'),
177     ('K002', '2024-12-02', 'Hadir'),
178     ('K003', '2024-12-02', 'Hadir');
179
180 -- Tabel penggajian
181 CREATE TABLE IF NOT EXISTS penggajian (
182     penggajian_id INTEGER PRIMARY KEY AUTOINCREMENT,
183     karyawan_id VARCHAR(5) NOT NULL,
184     bulan INT NOT NULL CHECK (bulan BETWEEN 1 AND 12),
185     tahun INT NOT NULL,
186     jumlah_gaji DECIMAL(15,2) NOT NULL CHECK (jumlah_gaji >= 0),
187     FOREIGN KEY (karyawan_id) REFERENCES karyawan(karyawan_id)
188 );
189
190 INSERT INTO penggajian (karyawan_id, bulan, tahun, jumlah_gaji) VALUES
191     ('K001', 12, 2024, 3000000.00),
192     ('K002', 12, 2024, 2800000.00),
193     ('K003', 12, 2024, 3200000.00);
```

SQL Database

```
195 -- Tabel pemesanan_bahan
196 CREATE TABLE IF NOT EXISTS pemesanan_bahan (
197     pemesanan_id VARCHAR(5) PRIMARY KEY,
198     supplier_id VARCHAR(5) NOT NULL,
199     tanggal_pemesanan DATE NOT NULL,
200     status VARCHAR(20) NOT NULL,
201     FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
202 );
203
204 INSERT INTO pemesanan_bahan (pemesanan_id, supplier_id, tanggal_pemesanan, status) VALUES
205     ('PB001', 'S001', '2024-12-05', 'Dipesan'),
206     ('PB002', 'S002', '2024-12-05', 'Dikirim');
207
208 -- Tabel detail_pemesanan_bahan
209 CREATE TABLE IF NOT EXISTS detail_pemesanan_bahan (
210     detail_pemesanan_id INTEGER PRIMARY KEY AUTOINCREMENT,
211     pemesanan_id VARCHAR(5) NOT NULL,
212     bahan_id VARCHAR(5) NOT NULL,
213     jumlah INT NOT NULL CHECK (jumlah > 0),
214     harga_satuan DECIMAL(15,2) NOT NULL CHECK (harga_satuan >= 0),
215     FOREIGN KEY (pemesanan_id) REFERENCES pemesanan_bahan(pemesanan_id),
216     FOREIGN KEY (bahan_id) REFERENCES bahan_baku(bahan_id)
217 );
218
219 INSERT INTO detail_pemesanan_bahan (pemesanan_id, bahan_id, jumlah, harga_satuan) VALUES
220     ('PB001', 'BB1', 20, 500000.00),
221     ('PB001', 'BB4', 30, 250000.00),
222     ('PB002', 'BB2', 50, 200000.00);
```

```
-- Tabel feedback pelanggan
CREATE TABLE IF NOT EXISTS feedback (
    feedback_id INTEGER PRIMARY KEY AUTOINCREMENT,
    pelanggan_id VARCHAR(5) NOT NULL,
    karyawan_id VARCHAR(5) NOT NULL,
    tanggal DATE NOT NULL,
    rating INT NOT NULL CHECK (rating BETWEEN 1 AND 5),
    komentar TEXT,
    FOREIGN KEY (pelanggan_id) REFERENCES pelanggan(pelanggan_id),
    FOREIGN KEY (karyawan_id) REFERENCES karyawan(karyawan_id)
);

INSERT INTO feedback (pelanggan_id, karyawan_id, tanggal, rating, komentar) VALUES
    ('P001', 'K003', '2024-12-02', 5, 'Pelayanan sangat memuaskan.'),
    ('P002', 'K001', '2024-12-03', 4, 'Makanan enak dan tempat nyaman.'),
    ('P003', 'K002', '2024-12-04', 3, NULL);
```

SQL Queries



1. Menampilkan daftar karyawan berdasarkan posisi

```
SELECT * FROM karyawan WHERE position = 'waiter';
```

karyawan_id	employee_name	position
K001	Adi	waiter
K002	Agus	waiter

2. Menampilkan daftar menu dengan harga di bawah Rp15.000

```
SELECT * FROM menu WHERE harga < 15000;
```

menu_id	nama_menu	harga
MN3	oseng kangkung	10000
MN4	tumis labu	12000
MN5	eskrim	10000
MN6	es jeruk	6000
MN7	esteh manis	5000

3. Menampilkan transaksi dengan total di atas Rp200.000

```
SELECT * FROM transaksi WHERE total_transaksi > 200000;
```

transaksi_id	tanggal_pembelian	total_transaksi
T002	2024-12-3	250000
T005	2024-12-6	280000
T006	2024-12-7	500000

SQL Queries

4. Menampilkan total pengeluaran berdasarkan pembelian bahan baku

SELECT

nama_bahan AS Nama_Bahan,
harga_bahan AS Harga_Per_Unit,
stock AS Jumlah_Stok,
(CAST(REPLACE(stock, ' bungkus', '') AS INT) * harga_bahan) AS Total_Pengeluaran

FROM

bahan_baku;

Nama_Bahan	Harga_Per_Unit	Jumlah_Stok	Total_Pengeluaran
Ayam potong	500000	50	25000000
Sayuran	200000	30	6000000
Ikan Lele	350000	20	7000000
Bawang	250000	50	12500000
Bumbu	600000	100 bungkus	60000000

SQL Queries

5. Menentukan total pendapatan per hari

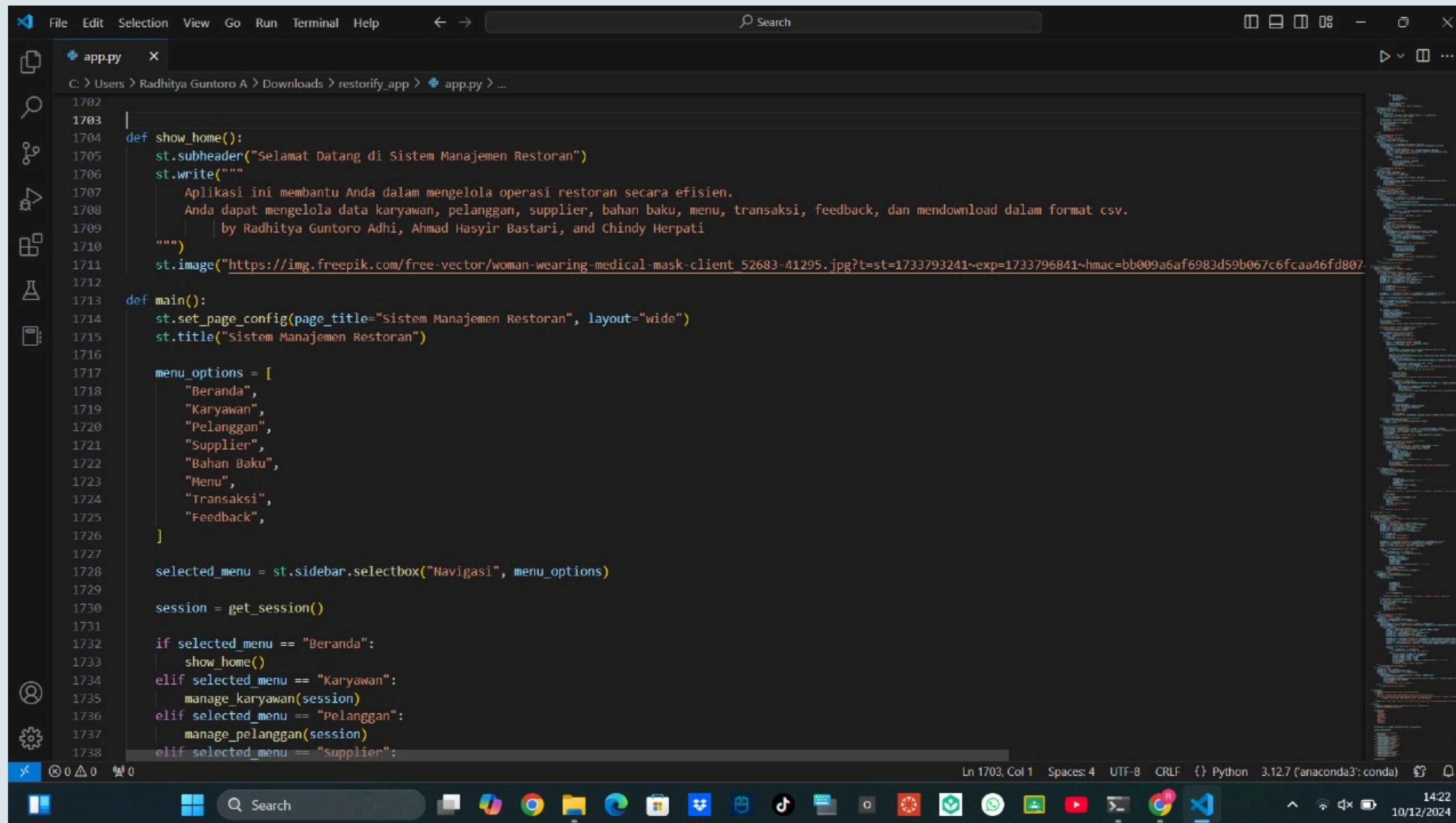
```
SELECT  
    tanggal_pembelian AS Tanggal,  
    SUM(total_transaksi) AS Total_Pendapatan  
FROM transaksi  
GROUP BY tanggal_pembelian  
ORDER BY Tanggal ASC;
```

```
SELECT  
    tanggal_pembelian AS Tanggal,  
    SUM(total_transaksi) AS Total_Pendapatan  
FROM  
    transaksi  
GROUP BY  
    tanggal_pembelian  
ORDER BY  
    Tanggal ASC;
```

Tanggal	Total_Pendapatan
2024-12-1	170000
2024-12-3	250000
2024-12-4	200000
2024-12-5	150000
2024-12-6	280000
2024-12-7	500000

Kode Program Python

Kutipan dari programming python: bagian app.py

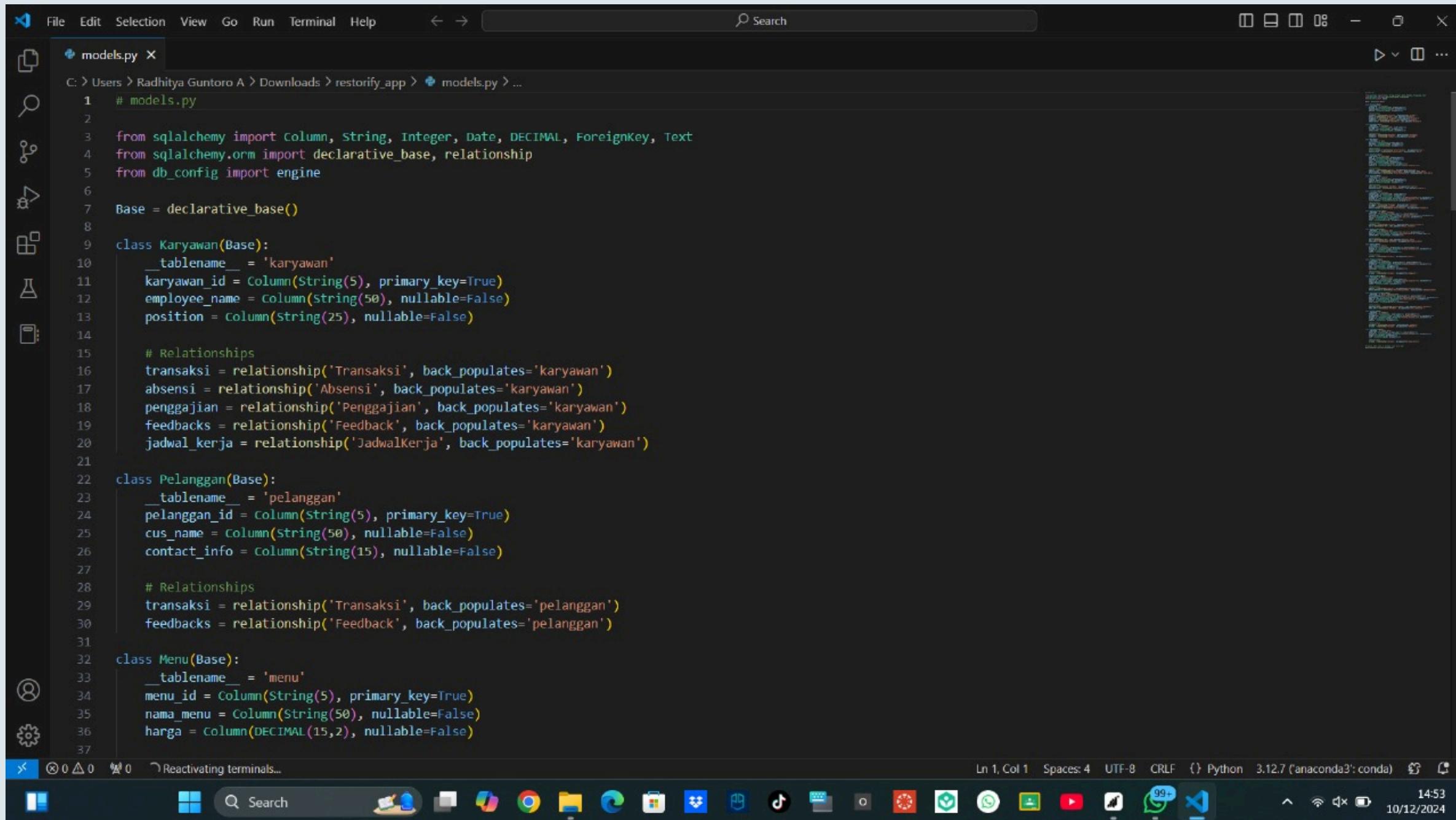


The screenshot shows a Jupyter Notebook interface with a single code cell containing Python code. The code defines a function `show_home()` which displays a welcome message, application details, and an image of a woman wearing a medical mask. It then defines a `main()` function which sets the page title, creates a sidebar menu with various options, and handles user selection. The code uses the Streamlit library (`st`) for the UI.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ Search
app.py
C: > Users > Radhiya Guntoro A > Downloads > restorify_app > app.py > ...
1702
1703     def show_home():
1704         st.subheader("Selamat Datang di Sistem Manajemen Restoran")
1705         st.write("""
1706             Aplikasi ini membantu Anda dalam mengelola operasi restoran secara efisien.
1707             Anda dapat mengelola data karyawan, pelanggan, supplier, bahan baku, menu, transaksi, feedback, dan mendownload dalam format csv.
1708             | by Radhiya Guntoro Adhi, Ahmad Hasyir Bastari, and Chindy Herpati
1709             """)
1710         st.image("https://img.freepik.com/free-vector/woman-wearing-medical-mask-client_52683-41295.jpg?t=st=1733793241~exp=1733796841~hmac=bb009a6af6983d59b067c6fcaa46fd807")
1711
1712     def main():
1713         st.set_page_config(page_title="Sistem Manajemen Restoran", layout="wide")
1714         st.title("Sistem Manajemen Restoran")
1715
1716         menu_options = [
1717             "Beranda",
1718             "Karyawan",
1719             "Pelanggan",
1720             "Supplier",
1721             "Bahan Baku",
1722             "Menu",
1723             "Transaksi",
1724             "Feedback",
1725         ]
1726
1727         selected_menu = st.sidebar.selectbox("Navigasi", menu_options)
1728
1729         session = get_session()
1730
1731         if selected_menu == "Beranda":
1732             show_home()
1733         elif selected_menu == "Karyawan":
1734             manage_karyawan(session)
1735         elif selected_menu == "Pelanggan":
1736             manage_pelanggan(session)
1737         elif selected_menu == "Supplier":
1738
Ln 1703, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  3.12.7 ('anaconda3': conda)  ⌂  ⌂
14:22  10/12/2024
```

Kode Program Python

Kutipan dari programming python: bagian models.py

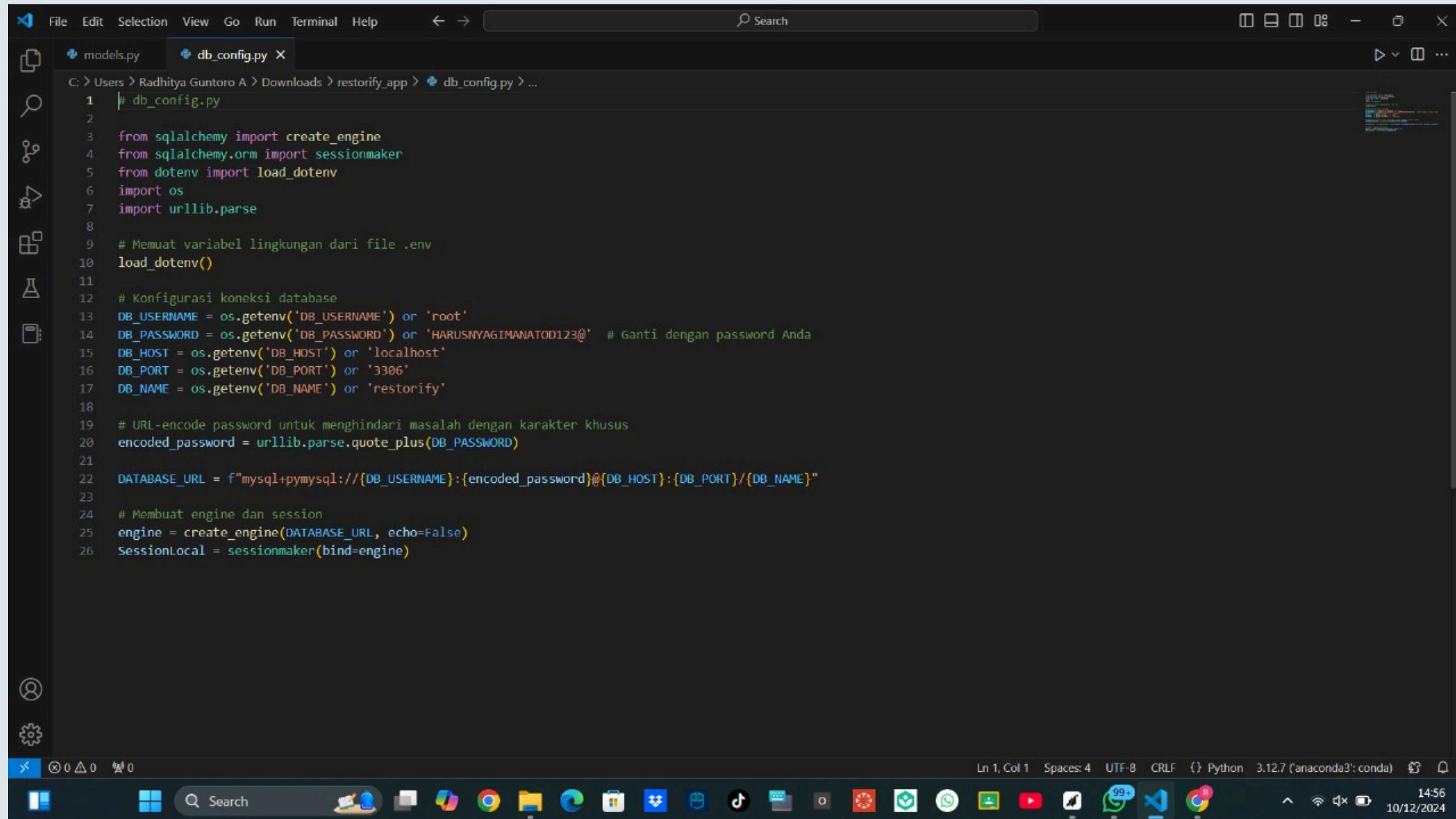


The screenshot shows a code editor window with the file 'models.py' open. The code defines three SQLAlchemy models: Karyawan, Pelanggan, and Menu. Each model has its own table name and primary key. The Karyawan model includes columns for employee_id, employee_name, and position. It also defines relationships for Transaksi, Absensi, Penggajian, Feedback, and Jadwal_Kerja. The Pelanggan model includes columns for pelanggan_id, cus_name, and contact_info. It defines relationships for Transaksi and Feedback. The Menu model includes columns for menu_id, nama_menu, and harga.

```
File Edit Selection View Go Run Terminal Help Search
models.py
C: > Users > Radhitya Guntoro A > Downloads > restorify_app > models.py > ...
1 # models.py
2
3 from sqlalchemy import Column, String, Integer, Date, DECIMAL, ForeignKey, Text
4 from sqlalchemy.orm import declarative_base, relationship
5 from db_config import engine
6
7 Base = declarative_base()
8
9 class Karyawan(Base):
10     __tablename__ = 'karyawan'
11     karyawan_id = Column(String(5), primary_key=True)
12     employee_name = Column(String(50), nullable=False)
13     position = Column(String(25), nullable=False)
14
15     # Relationships
16     transaksi = relationship('Transaksi', back_populates='karyawan')
17     absensi = relationship('Absensi', back_populates='karyawan')
18     penggajian = relationship('Penggajian', back_populates='karyawan')
19     feedbacks = relationship('Feedback', back_populates='karyawan')
20     jadwal_kerja = relationship('JadwalKerja', back_populates='karyawan')
21
22 class Pelanggan(Base):
23     __tablename__ = 'pelanggan'
24     pelanggan_id = Column(String(5), primary_key=True)
25     cus_name = Column(String(50), nullable=False)
26     contact_info = Column(String(15), nullable=False)
27
28     # Relationships
29     transaksi = relationship('Transaksi', back_populates='pelanggan')
30     feedbacks = relationship('Feedback', back_populates='pelanggan')
31
32 class Menu(Base):
33     __tablename__ = 'menu'
34     menu_id = Column(String(5), primary_key=True)
35     nama_menu = Column(String(50), nullable=False)
36     harga = Column(DECIMAL(15,2), nullable=False)
37
Ln 1, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  3.12.7 ('anaconda3':conda)  14:53
Reactivating terminals...
Search 10/12/2024
```

Kode Program Python

Kutipan dari programming python: bagian db_config.py



The screenshot shows a code editor window with the file `db_config.py` open. The code is written in Python and configures a database connection using SQLAlchemy. It loads environment variables from `.env`, handles URL encoding for the password, creates a database engine, and defines a sessionmaker.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ Search
C: > Users > Radhiya Guntoro A > Downloads > restorify_app > db_config.py > ...
models.py db_config.py X
1 # db_config.py
2
3 from sqlalchemy import create_engine
4 from sqlalchemy.orm import sessionmaker
5 from dotenv import load_dotenv
6 import os
7 import urllib.parse
8
9 # Memuat variabel lingkungan dari file .env
10 load_dotenv()
11
12 # Konfigurasi koneksi database
13 DB_USERNAME = os.getenv('DB_USERNAME') or 'root'
14 DB_PASSWORD = os.getenv('DB_PASSWORD') or 'HARUSNYAGIMANATOD123@' # Ganti dengan password Anda
15 DB_HOST = os.getenv('DB_HOST') or 'localhost'
16 DB_PORT = os.getenv('DB_PORT') or '3306'
17 DB_NAME = os.getenv('DB_NAME') or 'restorify'
18
19 # URL-encode password untuk menghindari masalah dengan karakter khusus
20 encoded_password = urllib.parse.quote_plus(DB_PASSWORD)
21
22 DATABASE_URL = f"mysql+pymysql://{DB_USERNAME}:{encoded_password}@{DB_HOST}:{DB_PORT}/{DB_NAME}"
23
24 # Membuat engine dan session
25 engine = create_engine(DATABASE_URL, echo=False)
26 SessionLocal = sessionmaker(bind=engine)
```

INTERFACE APLIKASI



The screenshot shows a web browser window titled "Sistem Manajemen Restoran" with the URL "localhost:8501". The interface has a dark theme. On the left, a sidebar labeled "Navigasi" contains a dropdown menu with "Beranda" selected, and other options like "Karyawan", "Pelanggan", "Supplier", "Bahan Baku", "Menu", "Transaksi", and "Feedback". The main content area features a large title "Sistem Manajemen Restoran" and a sub-title "Selamat Datang di Sistem Manajemen Restoran". Below this, a descriptive text states: "Aplikasi ini membantu Anda dalam mengelola operasi restoran secara efisien. Anda dapat mengelola data karyawan, pelanggan, supplier, bahan baku, menu, transaksi, feedback, dan mendownload dalam format csv. by Radhitya Guntoro Adhi, Ahmad Hasyir Bastari, and Chindy Herpati". At the bottom of the page is a cartoon illustration of a woman wearing a mask and a red and white striped awning over a shop. The browser's taskbar at the bottom shows various open tabs and icons, including Microsoft Edge, Google Chrome, and file explorers. The date and time in the bottom right corner are "10/12/2024" and "14:23".



Kesimpulan



Restorify merupakan sistem yang dapat mengelola seluruh **aspek operasional** restoran dengan **lebih efisien**, mulai dari **pengelolaan bahan baku** yang terintegrasi dengan **supplier**, **manajemen karyawan**, hingga **pencatatan transaksi** dan **laporan keuangan**. Dengan sistem yang terintegrasi ini, restoran akan memiliki kontrol yang lebih baik dalam pengelolaan bahan baku, karyawan, dan transaksi, serta mendapatkan laporan yang lebih akurat untuk **mendukung pengambilan keputusan** yang lebih baik.



Saran



Aplikasi ini masih dapat dikembangkan seperti **menambahkan fitur** untuk menyambungkan ke aplikasi pesanan online seperti, Grab, Gojek, Shopee dsb. Agar restoran dapat **mengakspan pasar** nya melalui penjualan online. Selain itu aplikasi ini masih memiliki kekurangan dalam sistem nya yaitu tidak adanya fitur untuk **mengatur fasilitas restoran** seperti meja, kursi, alat masak dan lainnya.





Thank you

