**Nama : Syafira Dewi Fatimah**

**Nim : 12030123140249**

**Kasus: Sistem Manajemen Restoran**

1. Latar Belakang

Restoran "Delish Eats" memiliki masalah dalam mengelola pesanan dan persediaan makanan. Saat ini, restoran tersebut masih menggunakan sistem manual di mana pelanggan mencatat pesanan di kertas, kasir menghitung total secara manual, dan admin harus memeriksa stok bahan setiap hari secara fisik. Hal ini menyebabkan beberapa masalah, seperti:

* Waktu pemesanan yang lambat: Pelanggan sering kali menunggu lama untuk memesan karena proses pencatatan manual.
* Kesalahan pesanan: Kadang-kadang pesanan pelanggan salah dicatat, sehingga menyebabkan ketidakpuasan pelanggan.
* Pengelolaan stok tidak efisien: Admin harus memeriksa stok bahan makanan secara manual, yang sering kali memakan waktu lama dan tidak akurat.
* Kesalahan dalam penghitungan pembayaran: Beberapa kali, kasir salah menghitung total pesanan atau memberikan kembalian yang salah.

2. Tujuan Sistem

Restoran ingin mengembangkan sistem manajemen restoran berbasis digital untuk:

* Mengotomatisasi proses pemesanan: Pelanggan dapat memilih makanan langsung dari perangkat restoran atau melalui aplikasi mobile.
* Menyederhanakan pengelolaan pembayaran: Kasir hanya perlu mengonfirmasi pesanan dan sistem akan secara otomatis menghitung total pembayaran.
* Memantau stok bahan secara real-time: Setiap kali pesanan diterima, stok bahan makanan akan berkurang secara otomatis di sistem sehingga admin bisa memantau persediaan dengan mudah.
* Menyimpan data pelanggan: Sistem menyimpan data pelanggan agar bisa memberikan promosi atau diskon khusus bagi pelanggan setia.

3. Fitur Sistem

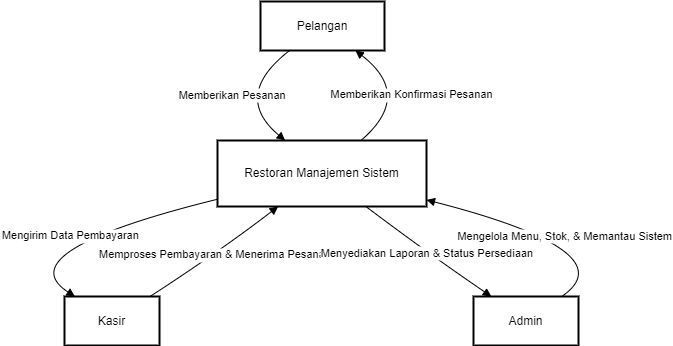
1. Manajemen Pesanan
   * Pelanggan dapat melihat menu secara digital dan membuat pesanan.
   * Sistem mencatat pesanan dan mengirimkannya ke dapur.
   * Kasir dapat melihat pesanan dan memproses pembayaran.
2. Manajemen Pembayaran
   * Sistem mendukung berbagai metode pembayaran, seperti tunai, kartu kredit, dan dompet digital.
   * Kasir dapat melihat total pembayaran secara otomatis, termasuk pajak dan diskon jika ada.
3. Manajemen Persediaan
   * Setiap kali ada pesanan, sistem secara otomatis memperbarui jumlah stok bahan makanan yang tersedia.
   * Admin dapat melihat laporan persediaan kapan saja dan mendapatkan notifikasi jika stok bahan hampir habis.
4. Manajemen Pelanggan
   * Sistem menyimpan data pelanggan, seperti nama, nomor telepon, dan riwayat pesanan.
   * Pelanggan dapat diberikan program loyalitas, seperti poin atau diskon untuk pesanan berikutnya.

**Answer:**

**Diagram Penyelesaian**

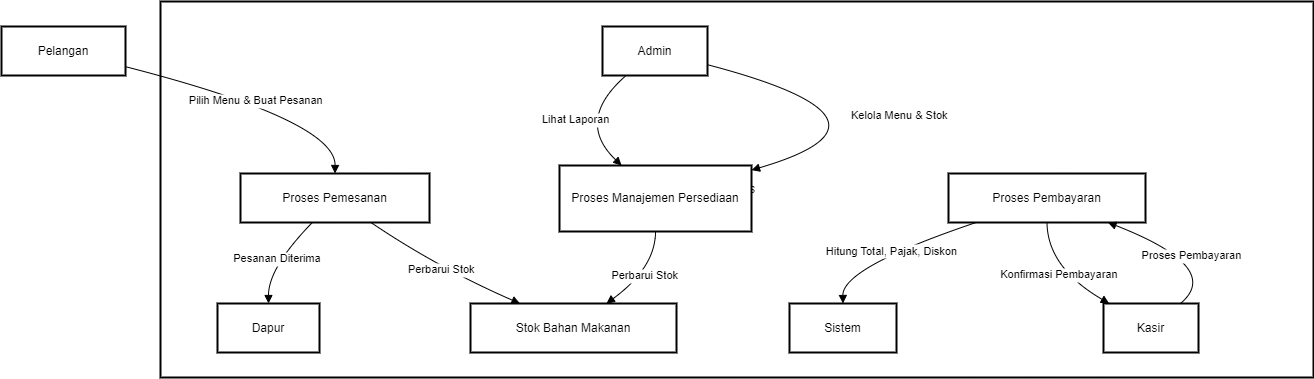
**1. Data Flow Diagram (DFD)**

**Level 0 - Context Diagram** Sistem manajemen restoran terhubung dengan beberapa entitas eksternal:



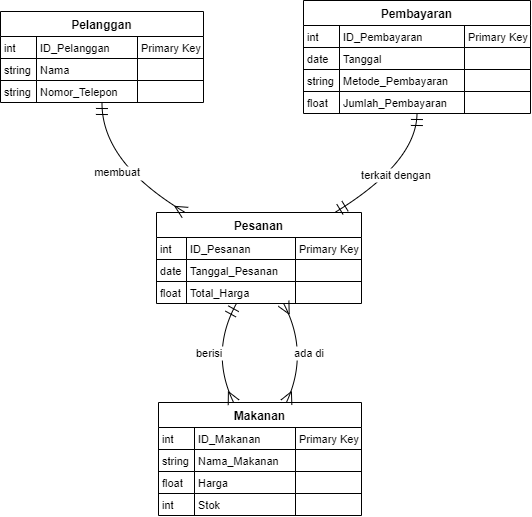
* **Pelanggan**: Pelanggan memberikan pesanan.
* **Kasir**: Kasir memproses pembayaran dan menerima pesanan.
* **Admin**: Admin mengelola menu, stok, dan memantau sistem.

**Level 1 - DFD Detil** Sistem dibagi menjadi beberapa proses utama:



1. **Proses Pemesanan**: Pelanggan memilih menu dan membuat pesanan. Pesanan diterima oleh dapur dan sistem memperbarui stok bahan.
2. **Proses Pembayaran**: Kasir memproses pembayaran berdasarkan pesanan pelanggan. Sistem menghitung total pembayaran, termasuk pajak dan diskon, serta mendukung berbagai metode pembayaran.
3. **Proses Manajemen Persediaan**: Setiap kali ada pesanan, stok bahan makanan diperbarui. Admin dapat melihat laporan persediaan dan mengelola stok.

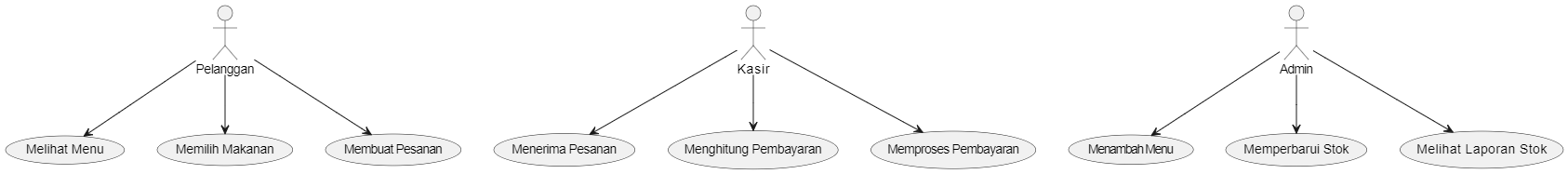
**2. Entity Relationship Diagram (ERD)**



* **Pelanggan**:
  + ID Pelanggan, Nama, Nomor Telepon.
  + Relasi: Setiap pelanggan dapat membuat beberapa pesanan.
* **Pesanan**:
  + ID Pesanan, Tanggal Pesanan, Total Harga.
  + Relasi: Setiap pesanan berisi beberapa makanan.
* **Makanan**:
  + ID Makanan, Nama Makanan, Harga, Stok.
  + Relasi: Setiap makanan dapat ada di beberapa pesanan.
* **Pembayaran**:
  + ID Pembayaran, Tanggal, Metode Pembayaran, Jumlah Pembayaran.
  + Relasi: Setiap pembayaran terkait dengan satu pesanan.

**3. UML Diagram**

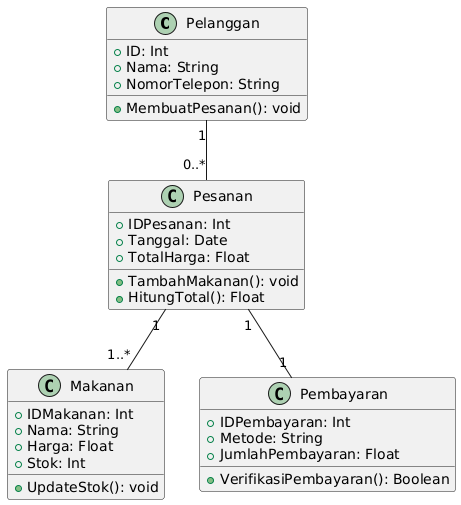
**a. Use Case Diagram**



Menggambarkan fungsionalitas utama sistem yang berhubungan dengan aktor:

* **Aktor**:
  + Pelanggan: Melihat menu, Membuat pesanan.
  + Kasir: Memproses pembayaran.
  + Admin: Mengelola stok, Menambah makanan.
* **Use Case**:
  + **Pelanggan**: Melihat menu, memilih makanan, membuat pesanan.
  + **Kasir**: Menerima pesanan, menghitung pembayaran, memproses pembayaran.
  + **Admin**: Menambah menu, memperbarui stok, melihat laporan stok.

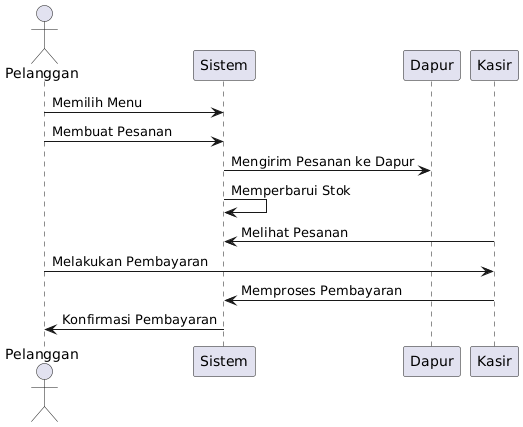
**b. Class Diagram**



Menggambarkan struktur data sistem, termasuk atribut dan method setiap kelas.

* **Kelas Pelanggan**:
  + Atribut: ID, Nama, Nomor Telepon
  + Method: Membuat Pesanan()
* **Kelas Pesanan**:
  + Atribut: ID Pesanan, Tanggal, Total Harga
  + Method: Tambah Makanan(), Hitung Total()
* **Kelas Makanan**:
  + Atribut: ID Makanan, Nama, Harga, Stok
  + Method: Update Stok()
* **Kelas Pembayaran**:
  + Atribut: ID Pembayaran, Metode, Jumlah Pembayaran
  + Method: Verifikasi Pembayaran()

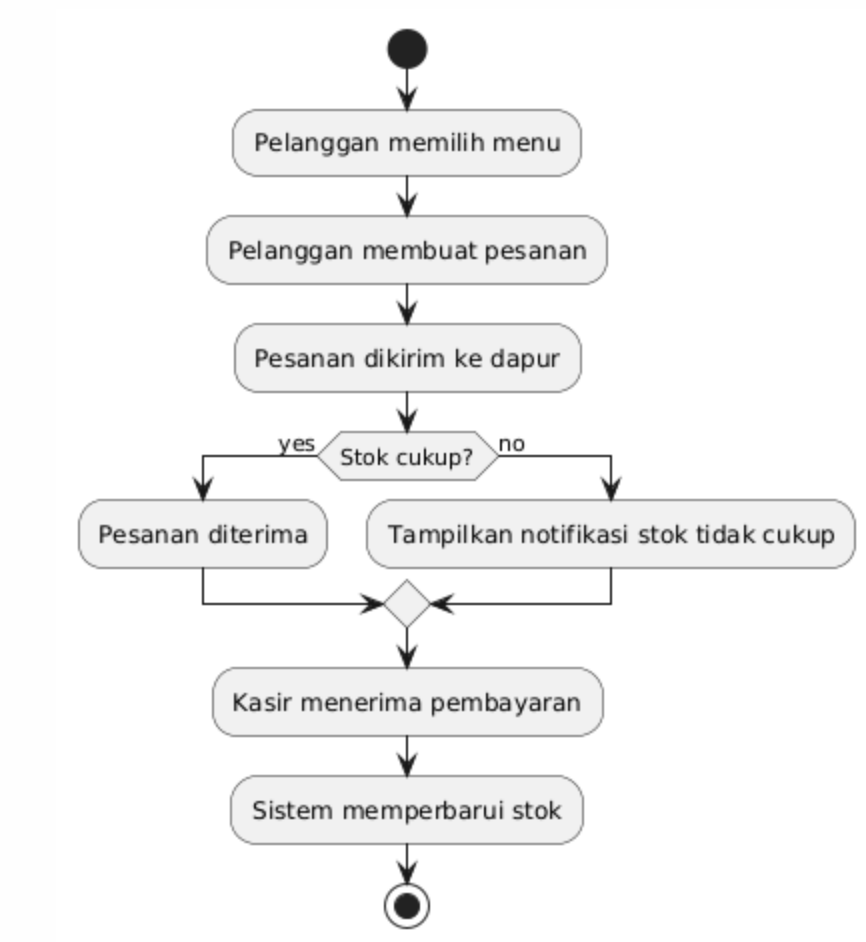
**c. Sequence Diagram**



Menggambarkan alur waktu interaksi antar entitas. Contoh:

1. Pelanggan memilih makanan dari menu.
2. Pelanggan melakukan pesanan.
3. Sistem mengirimkan pesanan ke dapur dan memperbarui stok.
4. Kasir memproses pesanan dan menerima pembayaran.

**d. Activity Diagram**

****

Menggambarkan langkah-langkah dalam sebuah proses, contohnya **Proses Pemesanan**:

1. Pelanggan memilih menu.
2. Pelanggan membuat pesanan.
3. Pesanan dikirim ke dapur.
4. Kasir menerima pembayaran.
5. Sistem memperbarui stok makanan.