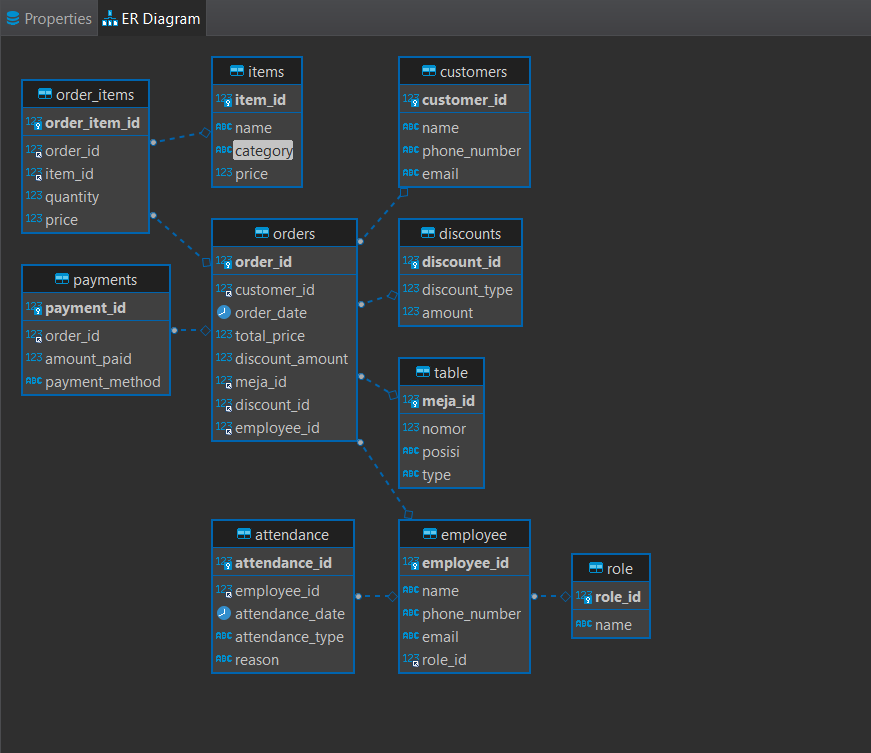
Berikut adalah rancangan diagram database dan teknologi yang akan digunakan untuk aplikasi rumah makan sesuai dengan kebutuhan yang telah disebutkan.

**1. Diagram Database**

**Entity Relationship Diagram (ERD)**

****

**Penjelasan Tabel:**

1. **Customers**: Menyimpan data pelanggan.
   * customer\_id (Primary Key)
   * name
   * contact\_info
2. **Employees**: Menyimpan data karyawan.
   * employee\_id (Primary Key)
   * name
   * email
   * role\_id (Foreign Key ke Role)
3. **Orders**: Menyimpan data pesanan.
   * order\_id (Primary Key)
   * customer\_id (Foreign Key ke Customers)
   * employee\_id (Foreign Key ke Employee)
   * meja\_id (Foreign Key ke table)
   * discount\_id (Foreign Key ke discounts)
   * order\_date
   * total\_price
4. **OrderItems**: Menyimpan rincian item yang dipesan.
   * order\_item\_id (Primary Key)
   * order\_id (Foreign Key ke Orders)
   * items\_id (Foreign Key ke Items)
   * quantity
   * price
5. **Payments**: Menyimpan informasi pembayaran.
   * payment\_id (Primary Key)
   * order\_id (Foreign Key ke Orders)
   * amount\_paid
   * payment\_method
6. **Items**: Menyimpan data items atau menu.
   * product\_id (Primary Key)
   * name
   * price
   * category
7. **Attendance**: Menyimpan informasi absen karyawan.
   * attendance\_id (Primary Key)
   * employee\_id (Foreign Key ke Employees)
   * attendance\_date
8. Role: Menyimpan data jabatan karyawan
   * Role\_id (Primary key)
   * name
9. Table : menyimpan data meja
   * Meja\_id primary key
   * Nomor
   * Posisi
   * type
10. Discounts :menyimpan data discount
    * Discount\_id primary key
    * Discount\_type
    * amount

**2. Teknologi yang Akan Digunakan**

**a. Backend**

* **Go (Golang)**: Digunakan untuk membangun API karena performanya yang cepat, efisiensi memori, dan kemampuannya dalam menangani banyak koneksi secara bersamaan. Golang juga mendukung concurrency, yang sangat penting untuk aplikasi yang memerlukan respons cepat.

**b. Database**

* **PostgreSQL**: Digunakan sebagai database utama karena dukungan kuat terhadap integritas data melalui constraints, transaction, dan fitur-fitur seperti JSONB untuk penyimpanan data semi-struktur. PostgreSQL juga mendukung berbagai tipe data dan kueri yang kompleks yang dibutuhkan dalam aplikasi restoran.

**c. Frontend**

* **React.js**: Untuk membangun user interface yang responsif dan dinamis. React memungkinkan pembuatan komponen-komponen UI yang dapat digunakan kembali, sehingga pengembangan UI lebih cepat dan efisien.
* **TypeScript**: Digunakan bersama React untuk meningkatkan kualitas kode melalui type checking, yang membantu dalam mencegah bug sejak awal pengembangan.

**d. API**

* **REST**: Sebagai interface API.

**e. Authentication & Authorization**

* **OAuth 2.0**: Digunakan untuk mengatur autentikasi dan otorisasi user dengan aman. Implementasi seperti JWT (JSON Web Token) akan digunakan untuk mengelola session user.

**f. Caching**

* **Redis**: Digunakan untuk caching data yang sering diakses seperti daftar menu dan stok barang, yang mempercepat respons API dan mengurangi beban pada database utama.

**g. Load Balancing & Scalability**

* **NGINX**: Digunakan sebagai reverse proxy dan load balancer untuk mendistribusikan trafik ke beberapa instance backend, memastikan aplikasi dapat menangani peningkatan beban pengguna.
* **Docker & Kubernetes**: Untuk containerization dan orchestration aplikasi. Dengan Docker, setiap bagian aplikasi bisa dikemas dalam container yang terisolasi, sementara Kubernetes membantu mengatur deployment, scaling, dan pengelolaan container secara otomatis.

**3. Fitur Tambahan**

* **Loyalty Program**: Menyediakan fitur loyalitas untuk pelanggan tetap, yang bisa memberikan poin setiap kali melakukan pembelian, dan poin ini bisa ditukarkan dengan diskon atau hadiah tertentu.
* **Analitik Penjualan**: Menggunakan alat analitik untuk menganalisis pola penjualan berdasarkan waktu, kategori produk, dan kinerja karyawan, memberikan wawasan yang lebih dalam bagi manajemen untuk pengambilan keputusan.
* **Integrasi dengan Layanan Pengantaran**: Menyediakan integrasi API dengan layanan pengantaran lokal untuk mempermudah pemesanan online dan pengiriman pesanan.
* **Data masuk karyawan untuk menganalisa kinerja karyawan dan penentu dalam payroll**

**Alasan Pemilihan Teknologi**

* **Performance & Scalability**: Pemilihan Golang dan PostgreSQL memberikan kinerja tinggi dengan skala yang mudah diperluas.
* **Security**: Menggunakan OAuth 2.0 dengan JWT, dan PostgreSQL yang sudah terbukti aman, menjamin keamanan data.
* **Maintainability & Flexibility**: React dan TypeScript memudahkan pengembangan frontend yang fleksibel dan mudah dipelihara.

Dengan teknologi dan rancangan ini, aplikasi rumah makan yang dibangun akan mampu menangani operasi harian dengan efisien, aman, dan siap untuk skala besar di masa depan.