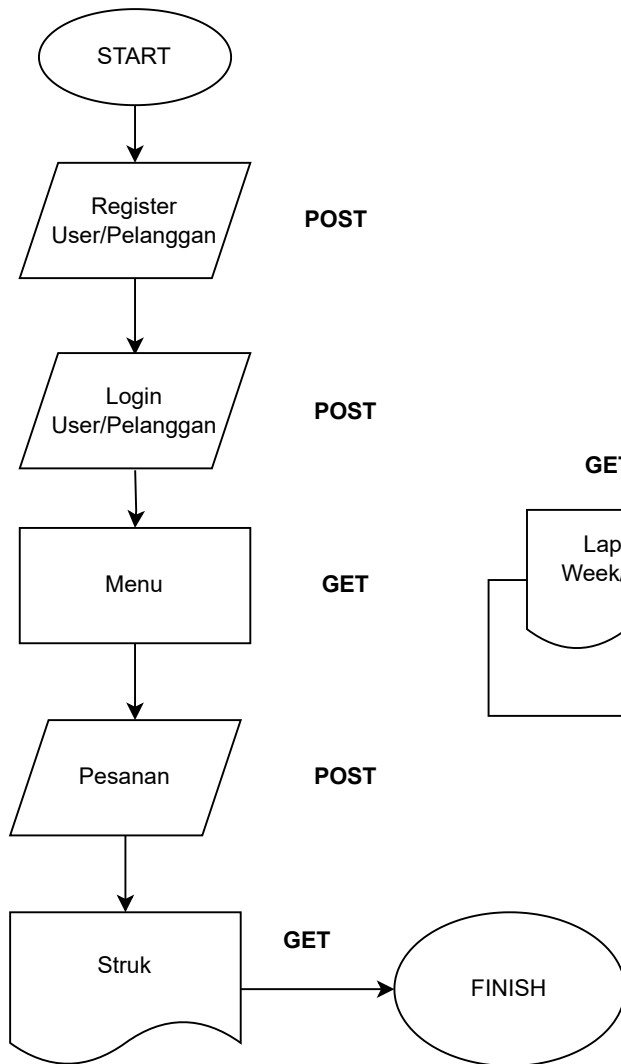


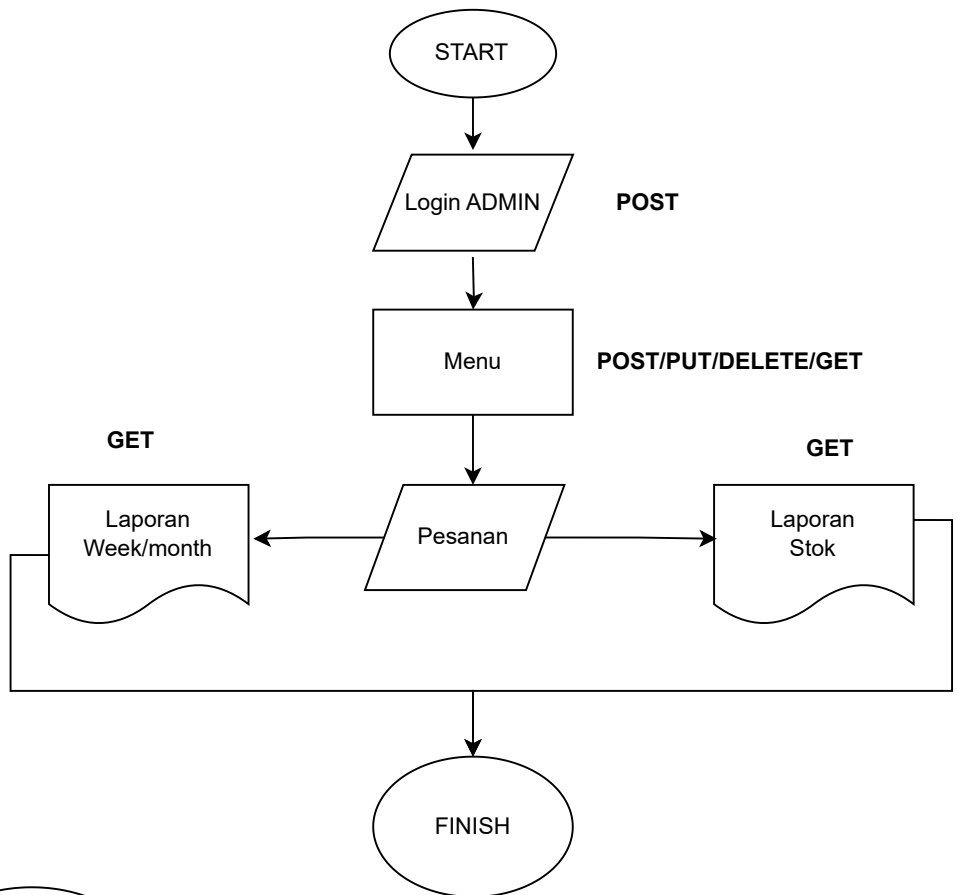
RUMAH MAKAN APP

Untuk merancang database aplikasi rumah makan APP ini, perlu mempertimbangkan entitas-entitas utama dan hubungan antara entitas tersebut. Berikut adalah skema database yang mungkin untuk aplikasi rumah makan

FLOW User / Pelanggan



FLOW ADMIN



Gambaran Entitas Umum :

Pelanggan

ID_Pelanggan (Primary Key)
 Nama_Pelanggan
 Nomor_Telepon
 email & Password
 Alamat

Menu

ID_Menu (Primary Key)
 Nama_Menu
 Harga
 Stok
 Deskripsi

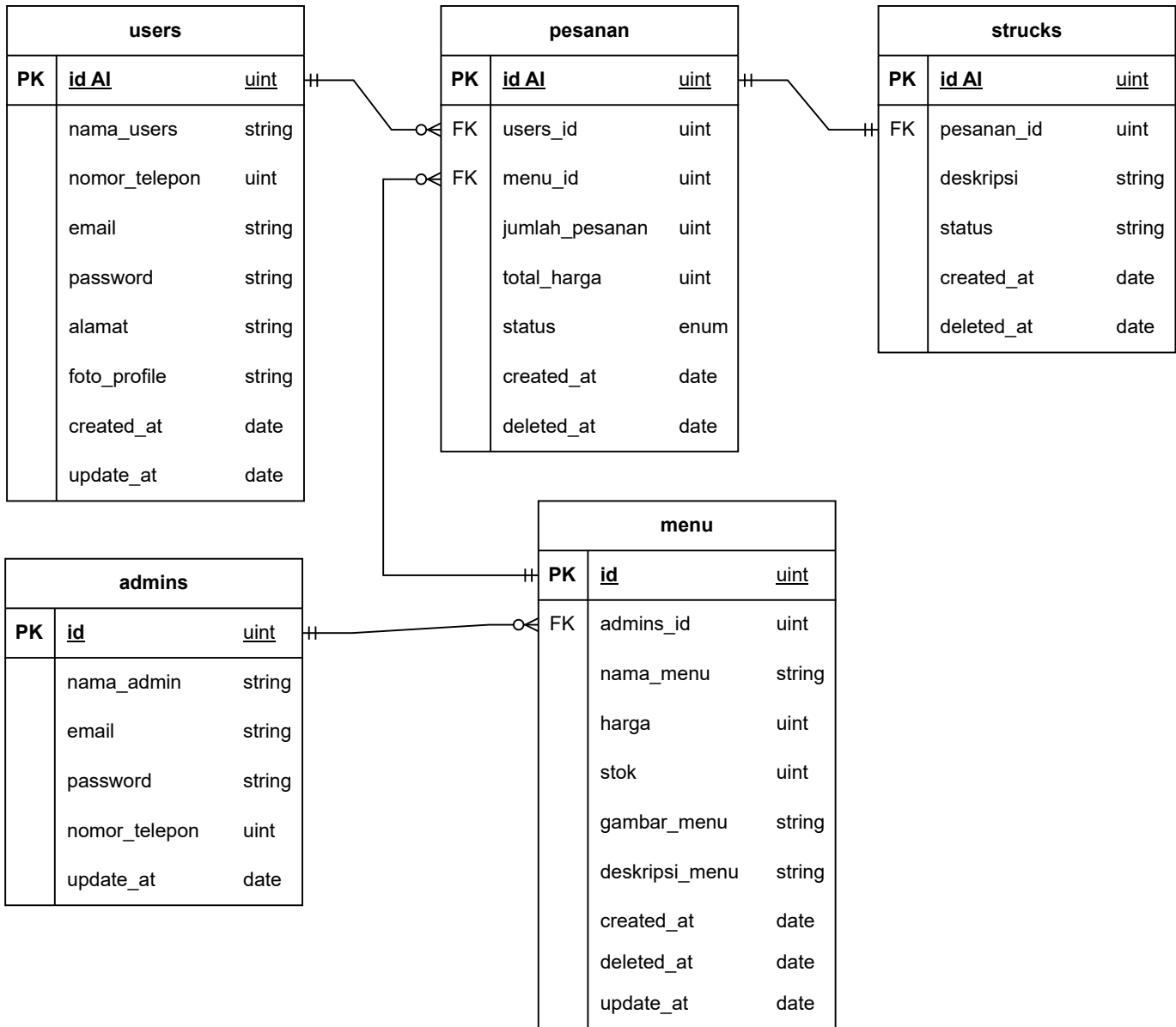
Pesanan

ID_Pesanan (Primary Key)
 Tanggal_Pesanan
 Jumlah
 Status_Pesanan (misalnya: Diproses, Selesai)
 Total_Harga

Struk

ID_Struk (Primary Key)
 Deskripsi
 Status

ERD (RUMAH MAKAN APP)



Hubungan:

USER dan Pesanan:

One-to-Many (Satu pelanggan bisa memiliki banyak pesanan).

Menu dan Pesanan:

One-to-Many (Satu menu bisa ada di banyak pesanan).

Pesanan dan Detail_Pesanan:

One-to-Many (Satu pesanan bisa memiliki banyak pesanan)

Pesanan dan Struk:

One-to-One(Satu pesanan menghasilkan satu struk).

Admin dan Menu:

One-to-Many (Satu Admin bisa input/ubah banyak menu)

Laporan:

1. Laporan Penghasilan Mingguan dan Bulanan dapat dihasilkan dengan menghitung total harga dari semua pesanan pada rentang waktu yang diinginkan. (menggunakan Endpoint GET dalam pesanan).
2. Laporan Stok dapat dihasilkan dengan melihat jumlah masing-masing menu yang tersisa dalam stok. (menggunakan Endpoint GET dalam Menu)

Teknologi Yang Digunakan:

1. **Bahasa Pemrograman:** Golang (Go)
2. **Database:** MySQL
3. **Framework (Opsional):** Echo Framework
4. **ORM :** GORM
5. **Deployment (CI/CD):** Google Cloud Platform
6. **Struckture Code :** Clean Architecture
7. **No Database :** Redis
8. **Platform :** Github, Docker
9. **Lain-lain :** Sweageer Open API, Postman

Alasan Pemilihan:

1. **Golang:** Karena Ringan, cepat, dan mudah untuk dikembangkan. Cocok untuk aplikasi berkinerja tinggi seperti sistem pembayaran dan manajemen pesanan.
2. **MySQL :** Database ini telah terbukti andal dalam manajemen data terstruktur, dan memiliki dukungan transaksi yang baik.
3. **Echo Framework:** Memudahkan pengembangan aplikasi web dengan Golang. Ringan dan efisien.
4. **GORM :** Karena Abstraksi,Basis Data, Portabilitas Kode, Pengembangan Cepat, dan Optimalisasi Kinerja akan lebih cepat.
5. **GCP :** Karena dengan teknologi Cloud, kita tidak perlu mengeluarkan biaya besar di awal untuk membangun infrastruktur aplikasi.
6. **Clean Architecture :** Meminimalisir terjadinya konflik dalam suatu kolaborasi, code mudah dipahami.
7. **Redis :** Karena untuk mempercepat dan memperingan kinerja pengambilan data dari server/database.
8. **Github :** Sebagai sarana untuk menyimpan dan berkolaborasi serta proses deployment
9. **Docker :** Sebagai sarana kontainer deployment
10. **Sweeager Open API :** karena lebih mempermudah untuk sarana berkomunikasi antara backend dan frondend
11. **Postman :** Untuk sarana pengecekan endpoint dari code yang kita buat.

Ide Tambahan:

1. **Integrasi Pembayaran:** integrasi dengan gateway pembayaran untuk memungkinkan pelanggan membayar secara online dengan mudah, misalkan dengan menggunakan pembayaran qoin, transfer bank, indomaret, alfa dll.
2. **Sistem Reward/Poin Pelanggan:** Penggunaan poin atau diskon kepada pelanggan setia berdasarkan jumlah pesanan atau total pengeluaran.
3. **Pantauan Stok Otomatis:** Implementasi sistem yang memberi tahu staf ketika stok menu mencapai batas tertentu. seperti notifikasi.