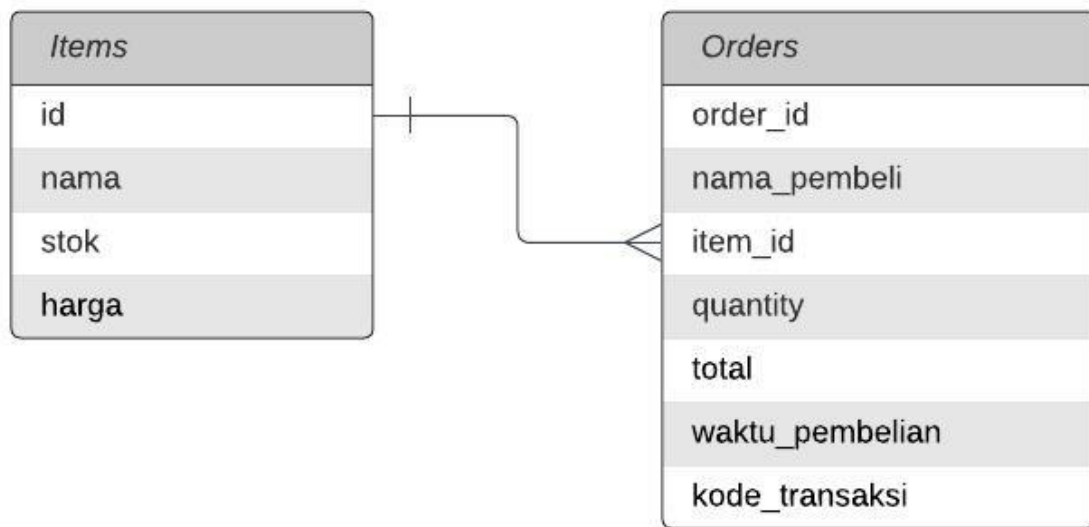


# Test 1

Sistem dibangun dengan menggunakan Golang dan Postgres.



## Menerima Pesanan

Untuk menyimpan pesanan, sistem akan menerima input yang akan di masukkan ke dalam struct, berupa nama pembeli, menu\_id serta jumlah pesanan, contohnya seperti berikut.

```
type Order struct {
    NamaPembeli string `json:"nama_pembeli"`
    Items Items `json:"items"`
}

type Items struct {
    ItemID int `json:"item_id"`
    Quantity int `json:"quantity"`
}
```

Contoh user memesan Nasi Goreng 3, Mie Goreng 1

Sistem melakukan pengecekan stok tiap item nya apakah mencukupi sesuai pesanan atau tidak.

```
SELECT stok FROM items WHERE item_id = Order.Items.ItemID;
```

Apabila stok tidak mencukupi, sistem mengembalikan pesan "Stok tidak mencukupi".  
Jika mencukupi, akan dilanjutkan ke tahap berikutnya.

Sistem akan loop untuk menghitung total harga dari tiap item yang dibeli dengan tahapan seperti berikut.

- Ambil harga tiap item nya dengan query berikut.

```
SELECT harga FROM item WHERE item_id = Order.Items.ItemID;
```

- Totalkan keseluruhan biaya pesanan.

```
total = total + (harga * quantity)
```

Contoh TotalNasiGoreng = 15000 \* 3 dan TotalMieGoreng = 20000 \* 1

Setelah total harga di dapat, sistem menyimpan data pesanan ke database dengan contoh query berikut.

```
INSERT INTO orders(nama_pembeli, item_id, quantity, total, waktu_pembelian, kode_transaksi)  
VALUES(Order>NamaPembeli, Order.Items.ItemID, Order.Items.Quantity, total, waktuPesan, hashTransaksi);
```

Dimana variable hashTransaksi merupakan kode unik hash dari gabungan string `NamaPembeli` + `waktuPesan` .

Kemudian sistem akan melakukan pengurangan stok pada database sesuai dengan pesanan yang dilakukan.

```
curStok = stok - quantity
```

```
UPDATE items SET stok = curStok WHERE item_id = Order.Items.ItemID;
```

## Cetak Struk Pembelian

Untuk mengeluarkan struk pembelian, dapat digunakan dengan contoh query berikut.

```
SELECT nama_pembeli, total, waktu_pembelian FROM order WHERE kode_transaksi = hashTransaksi;
```

Sistem akan memasukkan hasil perbaris item hasil query tersebut ke dalam struct seperti berikut.

```
type Bill struct {
    NamaPembeli string json:"nama_pembeli"
    NamaItem string json:"nama"
    Harga double json:"harga"
    Quantity int json:"quantity"
    SubTotal double json:"sub_total"
    WaktuPembelian time.time json:"waktu_pembelian"
    Total double json:"total"
}
```

Kemudian sistem akan melakukan perhitungan persubtotalnya dan akan mencetak struk pembelian dalam format berikut.

```
{
  "nama_pembeli": "John Doe",
  "items": [
    {
      "nama": "Nasi Goreng",
      "harga": 15000,
      "quantity": 3,
      "sub_total": 45000
    },
    {
      "nama": "Mie Goreng",
      "harga": 20000,
      "quantity": 1,
      "sub_total": 20000
    }
  ],
  "waktu_pembelian": "2023-03-19 10:48:24",
  "total": 65000
}
```

## Cetak Laporan Stok

Untuk laporan stok, akan menampilkan 3 buah laporan, yaitu:

1. Laporan stok saat ini.

```
SELECT id, nama, stok FROM items;
```

id	nama	stok
1	Nasi Goreng	10
2	Mie	15

## 2. Laporan stok seminggu terakhir.

```
SELECT id, nama, (
  SELECT SUM(quantity) FROM orders
  WHERE item_id = i.id
  AND waktu_pembelian >= current_date at time zone 'UTC' - interval '7 days') terjual,
stok as current_stok
FROM items i;
```

id	nama	terjual	current_stok
1	Nasi Goreng	7	10
2	Mie	6	15

## 3. Laporan stok sebulan terakhir.

```
SELECT id, nama, (
  SELECT SUM(quantity) FROM orders
  WHERE item_id = i.id
  AND waktu_pembelian >= current_date at time zone 'UTC' - interval '30 days') terjual,
stok as current_stok
FROM items i;
```

id	nama	terjual	current_stok
1	Nasi Goreng	9	10
2	Mie	9	15

# Laporan Penjualan

Untuk laporan penjualan, akan menampilkan 2 buah laporan, yaitu:

## 1. Laporan penjualan seminggu terakhir.

```

SELECT id, nama, (
  SELECT SUM(quantity) FROM orders
  WHERE item_id = i.id
  AND waktu_pembelian >= current_date at time zone 'UTC' - interval '7 days') stok_terjual,
  (SELECT SUM(total) FROM orders
  WHERE item_id = i.id
  AND waktu_pembelian >= current_date at time zone 'UTC' - interval '7 days') nominal
FROM items i;

```

id	nama	stok_terjual	nominal
1	Nasi Goreng	7	105000
2	Mie	6	120000

## 2. Laporan penjualan sebulan terakhir.

```

SELECT id, nama, (
  SELECT SUM(quantity) FROM orders
  WHERE item_id = i.id
  AND waktu_pembelian >= current_date at time zone 'UTC' - interval '30 days') stok_terjual,
  (SELECT SUM(total) FROM orders
  WHERE item_id = i.id
  AND waktu_pembelian >= current_date at time zone 'UTC' - interval '30 days') nominal
FROM items i;

```

id	nama	stok_terjual	nominal
1	Nasi Goreng	9	135000
2	Mie	9	180000