Untuk menampilkan message di Pesanan:

cart.html

{% extends 'base.html' %}

{% block content %}

**{% if messages %}**

**<div class="container">**

**<div class="row">**

**<div class="col-md-12">**

**{% for message in messages %}**

**<div class="alert alert-{{ message.tags }}">**

**{{ message }}**

**</div>**

**{% endfor %}**

**</div>**

**</div>**

**</div>**

**{% endif %}**

<div class="table-responsive">

    <table class="table table-bordered">

        <thead>

            <tr>

                <th>Item</th>

                <th>Harga</th>

                <th>Jumlah</th>

                <th>Sub total</th>

                <th>Aksi</th>

            </tr>

        </thead>

        <tbody>

            {% for item in cart\_items %}

            <tr>

                <td>{{ item.makanan.nama }}</td>

                <td>Rp{{ item.makanan.harga }}</td>

                <td>{{ item.quantity }}</td>

                <td>Rp{{ item.item\_total }}</td>

                <td>

                    {% if item.makanan.id %}

                        <a **onclick='return confirm("hapus pesanan?")'** href="{% url 'remove\_from\_cart' item.makanan.id %}">

                            <button class="btn btn-danger">Hapus</button>

                        </a>

                    {% endif %}

                </td>

            </tr>

**{% empty %}**

**<tr>**

**<td colspan="5">Tidak ada item dalam keranjang.</td>**

            </tr>

            {% endfor %}

        </tbody>

    </table>

</div>

<p>Total : Rp{{ total\_price }}</p>

{% endblock content %}

Perubahan di views.py

from django.shortcuts import render

from django.http import HttpResponse

from .models import Makanan

from django.shortcuts import get\_object\_or\_404

from django.shortcuts import redirect

from .models import CartItem

from django.contrib import messages

# Create your views here.

def home(request):

  searchMakanan = request.GET.get('nama')

  if searchMakanan:

    makanans = Makanan.objects.filter(nama\_\_icontains=searchMakanan)

  else:

    makanans = Makanan.objects.all()

  return render(request, 'home.html', {'searchMakanan':searchMakanan,'makanans' : makanans})

def detail(request,makanan\_id):

  makanan = get\_object\_or\_404(Makanan,pk=makanan\_id)

  return render(request,'detail.html',{'makanan' : makanan})

def add\_to\_cart(request, makanan\_id):

    makanan\_id = int(makanan\_id)

    makanan = get\_object\_or\_404(Makanan, pk=makanan\_id)

    if request.method == 'POST':

        quantity = int(request.POST.get('quantity', 1))  # jumlah pesanan dari form

        if quantity < 1:

            quantity = 1  # minimal 1

**if makanan.stok >= quantity:  # cek apakah stok ada**

            # cart\_item jika sudah ada pesanan makanan tersebut

            # created jika belum ada pesanan makanan tersebut maka di simpan dan created = true

            cart\_item, created = CartItem.objects.get\_or\_create(makanan=makanan)

            # If the item exists, update the quantity and item\_total

            if not created:

                cart\_item.quantity += quantity

                cart\_item.item\_total = cart\_item.quantity \* makanan.harga

                cart\_item.save()

            else:

                cart\_item.quantity = quantity

                cart\_item.item\_total = quantity \* makanan.harga

                cart\_item.save()

            # kurangi stok

**makanan.stok -= quantity**

**makanan.save()**

        else:

            # message jika stok kurang

**messages.error(request, 'Stok habis untuk makanan ' + makanan.nama)**

    return redirect('cart')

def view\_cart(request):

    cart\_items = CartItem.objects.all()

    total\_price = sum(item.item\_total for item in cart\_items)

    return render(request, 'cart.html', {'cart\_items': cart\_items, 'total\_price': total\_price})

def remove\_from\_cart(request, makanan\_id):

    makanan\_id = int(makanan\_id)

    makanan = Makanan.objects.get(pk=makanan\_id)

    try:

        cart\_item = CartItem.objects.get(makanan=makanan)

        cart\_item.delete()

**messages.success(request, 'Pesanan berhasil** dihapus')

    except CartItem.DoesNotExist:

**messages.warning(request, 'Pesanan tidak ditemukan')**

    return redirect('cart')