

## LISTING PROGRAM

views.py: Program back end dari *system order*

```
import json

from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.conf import settings

import requests

from rest_framework import generics
from . import models, serializers, permissions

# Create your views here.
class PlaceList(generics.ListCreateAPIView):
    serializer_class = serializers.PlaceSerializer

    def get_queryset(self):
        return models.Place.objects.filter(owner_id=self.request.user.id)

    def perform_create(self, serializer):
        serializer.save(owner=self.request.user)

class PlaceDetail(generics.RetrieveUpdateDestroyAPIView):
    permission_classes = [permissions.IsOwnerOrReadOnly]
    serializer_class = serializers.PlaceDetailSerializer
    queryset = models.Place.objects.all()

class CategoryList(generics.CreateAPIView):
    permission_classes = [permissions.PlaceOwnerOrReadOnly]
    serializer_class = serializers.CategorySerializer

class CategoryDetail(generics.UpdateAPIView, generics.DestroyAPIView):
    permission_classes = [permissions.PlaceOwnerOrReadOnly]
    serializer_class = serializers.CategorySerializer
    queryset = models.Category.objects.all()

class MenuItemList(generics.CreateAPIView):
    permission_classes = [permissions.PlaceOwnerOrReadOnly]
    serializer_class = serializers.MenuItemSerializer

class MenuItemDetail(generics.UpdateAPIView, generics.DestroyAPIView):
    permission_classes = [permissions.PlaceOwnerOrReadOnly]
    serializer_class = serializers.MenuItemSerializer
    queryset = models.MenuItem.objects.all()

@csrf_exempt
def create_payment(request):
    try:
        data = json.loads(request.body)

        order = models.Order.objects.create(
            place_id = data['place'],
            table = data['table'],
            detail = json.dumps(data['detail']),
```

```
        amount = data['amount'],
    )

    return JsonResponse({
        "success": True,
        "order": order.id,
    })
except Exception as e:
    return JsonResponse({
        "success": False,
        "error": str(e),
    })

class OrderList(generics.ListAPIView):
    serializer_class = serializers.OrderSerializer

    def get_queryset(self):
        return models.Order.objects.filter(place__owner_id=self.request.user.id,
        place_id=self.request.GET.get('place'))

class OrderDetail(generics.UpdateAPIView):
    permission_classes = [permissions.PlaceOwnerOrReadOnly]
    serializer_class = serializers.OrderSerializer
    queryset = models.Order.objects.all()
```