LAMPIRAN

Lampiran 1 Kode Python (Model)

```python
from django.db import models
from django.utils import timezone

class Video(models.Model):
    video_file = models.FileField(upload_to='videos/')
    upload_date = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.video_file.name

class DeteksiKendaraan(models.Model):
    timestamp = models.DateTimeField(default=timezone.now)
    motor_masuk = models.IntegerField(default=0)
    motor_keluar = models.IntegerField(default=0)
    mobil_masuk = models.IntegerField(default=0)
    mobil_keluar = models.IntegerField(default=0)
    deteksi_durasi = models.CharField(max_length=10)

    def _str_(self):
        return f"Deteksi pada {self.timestamp}: Motor Masuk={self.motor_masuk}, Motor Keluar={self.motor_keluar}, Mobil Masuk={self.mobil_masuk},

    def total_motor_masuk(self):
        return self.motor_masuk

    def total_motor_keluar(self):
        return self.motor_keluar
    def total_mobil_masuk(self):
        return self.mobil_masuk

    def total_mobil_keluar(self):
        return self.mobil_keluar
```

Lampiran 2 Kode Python (Fungsi Upload Video)

```python
@login_required
def upload_video(request):
    if request.method == 'POST':
        form = VideoForm(request.POST, request.FILES)
        if form.is_valid():
            video_instance = form.save()
            return redirect('process_video', video_id=video_instance.id)
    else:
        form = VideoForm()
    return render(request, 'dashboard/upload.html', {'form': form})
```

Lampiran 3 Kode Python (Fungsi proses penyimpanan file video ke dalam local)

```python
def process_video(request, video_id):
    video = Video.objects.get(id=video_id)
    video_path = video.video_file.path
    process_video_with_yolo(video_path)
    return redirect('deteksi_kendaraan_list')
```

Lampiran 4 Kode Python (Integrasi model yolo untuk memproses deteksi)

```python
def process_video_with_yolo(video_path):
    model_path = 'models/best.pt'  # Sesuaikan dengan path model YOLO Anda
    model = YOLO(model_path)

    cap = cv2.VideoCapture(video_path)
    fps = cap.get(cv2.CAP_PROP_FPS)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('cuda-output6.avi', fourcc, fps, (int(cap.get(3)), int(cap.get(4))))

    line1_start, line1_end = (335, 425), (584, 425)
    line2_start, line2_end = (849, 315), (1067, 319)

    counts = {
        'motor_in': 0,
        'motor_out': 0,
        'mobil_in': 0,
        'mobil_out': 0
    }

    def crossing_line(point, line_start, line_end):
        x, y = point
        x1, y1 = line_start
        x2, y2 = line_end
        dx = x2 - x1
        dy = y2 - y1
        if dx == 0:
            return x == x1 and min(y1, y2) <= y <= max(y1, y2)
        else:
            slope = dy / dx
            intercept = y1 - slope * x1
            y_line = slope * x + intercept
            return y == int(y_line)

    frame_count = 0
    video_start_time = timezone.now()  # Waktu mulai pemrosesan video dengan timezone Asia/Makassar
```

```python
    channel_layer = get_channel_layer()

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        frame_count += 1
        current_time = frame_count / fps
        minutes = int(current_time // 60)
        seconds = int(current_time % 60)
        waktu_deteksi = f'{minutes:02}:{seconds:02}'
        timestamp = video_start_time + timedelta(seconds=current_time)  # Menghitung timestamp berdasarkan waktu mulai video

        results = model(frame)
        detected_points = []

        for result in results:
            boxes = result.boxes
            for box in boxes:
                x1, y1, x2, y2 = map(int, box.xyxy[0])
                center_x, center_y = (x1 + x2) // 2, (y1 + y2) // 2
                detected_points.append((center_x, center_y, int(box.cls[0])))

                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
                class_id = int(box.cls[0])
                confidence = box.conf[0]
                label = f'{model.names[class_id]} {confidence:.2f}'
                cv2.putText(frame, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

        for point in detected_points:
            center_x, center_y, class_id = point
            class_name = model.names[class_id]
```

```
183
184             if crossing_line((center_x, center_y), line1_start, line1_end):
185                 if class_name == 'motor':
186                     counts['motor_in'] += 1
187                     time_text = f'Motor In: {waktu_deteksi}'
188                     cv2.putText(frame, time_text, (center_x, center_y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
189                     print(f'Motor In at {waktu_deteksi}')
190                 elif class_name == 'mobil':
191                     counts['mobil_in'] += 1
192                     time_text = f'Mobil In: {waktu_deteksi}'
193                     cv2.putText(frame, time_text, (center_x, center_y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
194                     print(f'Mobil In at {waktu_deteksi}')
195
196             elif crossing_line((center_x, center_y), line2_start, line2_end):
197                 if class_name == 'motor':
198                     counts['motor_out'] += 1
199                     time_text = f'Motor Out: {waktu_deteksi}'
200                     cv2.putText(frame, time_text, (center_x, center_y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
201                     print(f'Motor Out at {waktu_deteksi}')
202                 elif class_name == 'mobil':
203                     counts['mobil_out'] += 1
204                     time_text = f'Mobil Out: {waktu_deteksi}'
205                     cv2.putText(frame, time_text, (center_x, center_y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
206                     print(f'Mobil Out at {waktu_deteksi}')
207
208         # Simpan data ke database setiap frame
209         DeteksiKendaraan.objects.create(
210             timestamp = timezone.now(),  # Mengatur timezone ke Asia/Makassar sebelum menyimpan
211             motor_masuk=counts['motor_in'],
212             motor_keluar=counts['motor_out'],
213             mobil_masuk=counts['mobil_in'],
214             mobil_keluar=counts['mobil_out'],
215             deteksi_durasi=waktu_deteksi
216         )
```

```
218         async_to_sync(channel_layer.group_send)(
219             "dashboard_group", {
220                 "type": "send_data",
221                 "motor_in": counts['motor_in'],
222                 "motor_out": counts['motor_out'],
223                 "mobil_in": counts['mobil_in'],
224                 "mobil_out": counts['mobil_out'],
225             }
226         )
227
228         print(f"Total Motor Masuk: {counts['motor_in']}")
229         print(f"Total Motor Keluar: {counts['motor_out']}")
230         print(f"Total Mobil Masuk: {counts['mobil_in']}")
231         print(f"Total Mobil Keluar: {counts['mobil_out']}")
232
233         cv2.line(frame, line1_start, line1_end, (0, 0, 255), 2)
234         cv2.line(frame, line2_start, line2_end, (255, 0, 0), 2)
235         cv2.putText(frame, f'Motor In: {counts["motor_in"]}', (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
236         cv2.putText(frame, f'Motor Out: {counts["motor_out"]}', (10, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
237         cv2.putText(frame, f'Mobil In: {counts["mobil_in"]}', (10, 150), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
238         cv2.putText(frame, f'Mobil Out: {counts["mobil_out"]}', (10, 200), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
239
240         out.write(frame)
241         if cv2.waitKey(1) & 0xFF == ord('q'):
242             break
243
244     cap.release()
245     out.release()
246     cv2.destroyAllWindows()
```

Lampiran 5 Kode Python (Fungsi Web Socket)

Settings.py

```python
INSTALLED_APPS = [
    'channels',

    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'django_cas_ng',

    'parkir',
]

ASGI_APPLICATION = 'mysite.asgi.application'


CHANNEL_LAYERS = {
    'default': {
        'BACKEND': 'channels.layers.InMemoryChannelLayer',  # Gunakan InMemoryChannelLayer untuk mode pengembangan
        'CONFIG': {
            'capacity': 20000,  # Kapasitas maksimum antrian pesan
        },
```

asgi.py

```python
import os
from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack
import parkir.routing

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'mysite.settings')

application = ProtocolTypeRouter({
    'http': get_asgi_application(),
    'websocket': AuthMiddlewareStack(
        URLRouter(
            parkir.routing.websocket_urlpatterns
        )
    ),
})
```

Consumers.py

```python
import json
from channels.generic.websocket import AsyncWebsocketConsumer
from asgiref.sync import async_to_sync

class MonitoringConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        await self.channel_layer.group_add(
            "dashboard_group",
            self.channel_name
        )
        await self.accept()

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(
            "dashboard_group",
            self.channel_name
        )

    async def send_data(self, event):
        await self.send(text_data=json.dumps({
            'motor_in': event['motor_in'],
            'motor_out': event['motor_out'],
            'mobil_in': event['mobil_in'],
            'mobil_out': event['mobil_out'],
        }))
```

Routing.py

```python
from django.urls import re_path
from . import consumers

websocket_urlpatterns = [
    re_path(r'ws/video/$', consumers.MonitoringConsumer.as_asgi()),
]
```

Javascript koneksi websocket

```
<script>
    document.addEventListener('DOMContentLoaded', function () {
        var socket = new WebSocket('ws://' + window.location.host + '/ws/video/');

        socket.onmessage = function (e) {
            var data = JSON.parse(e.data);
            console.log("Data received:", data);

            document.getElementById('motor_in').innerText = data.motor_in;
            document.getElementById('motor_out').innerText = data.motor_out;
            document.getElementById('mobil_in').innerText = data.mobil_in;
            document.getElementById('mobil_out').innerText = data.mobil_out;
        };

        socket.onopen = function (e) {
            console.log('WebSocket connection opened');
        };

        socket.onclose = function (e) {
            console.log('WebSocket connection closed');
        };

        socket.onerror = function (error) {
            console.error('WebSocket Error:', error);
        };
    });
</script>
```

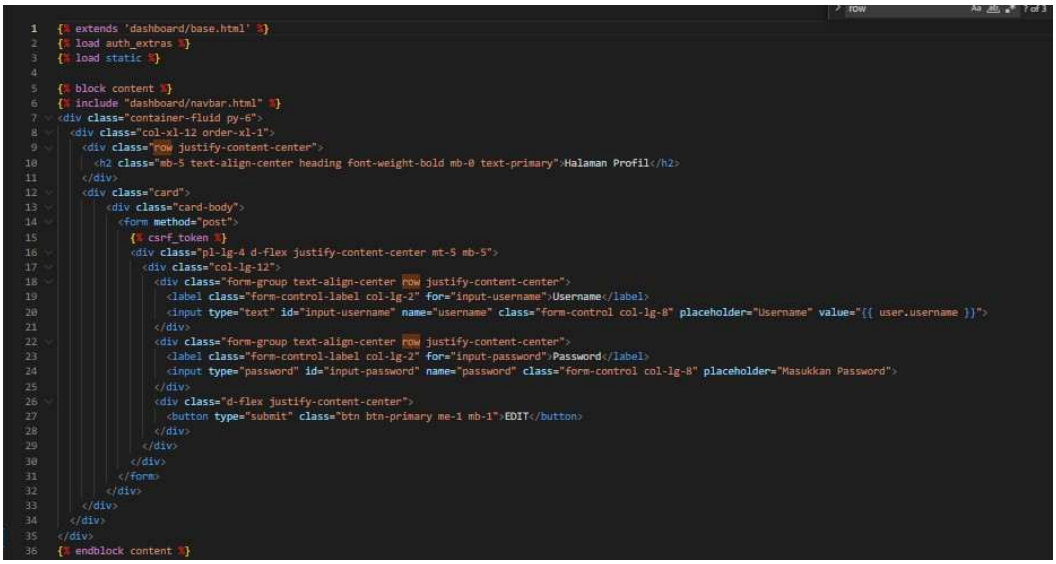
Lampiran 6 Kode Python (Profile)

```
@login_required
def profile(request):
    user = request.user

    if request.method == 'POST':
        form = UserProfileForm(request.POST, instance=user)
        if form.is_valid():
            user = form.save(commit=False)
            if form.cleaned_data['password']:
                user.set_password(form.cleaned_data['password'])
                user.save()
                update_session_auth_hash(request, user)
            else:
                user.save()
            messages.success(request, 'Profile updated successfully')
            return redirect('profile')
        else:
            messages.error(request, 'Please correct the error below.')
    else:
        form = UserProfileForm(instance=user)

    context = {
        'form': form,
        'user': user
    }

    return render(request, 'dashboard/profile.html', context)
```



```
{% extends 'dashboard/base.html' %}
{% load auth_extras %}
{% load static %}

{% block content %}
{% include "dashboard/navbar.html" %}
<div class="container-fluid py-6">
  <div class="col-xl-12 order-xl-1">
    <div class="row justify-content-center">
      <h2 class="mb-5 text-align-center heading font-weight-bold mb-0 text-primary">Halaman Profil</h2>
    </div>
    <div class="card">
        <div class="card-body">
          <form method="post">
            {% csrf_token %}
            <div class="pl-lg-4 d-flex justify-content-center mt-5 mb-5">
              <div class="col-lg-12">
                <div class="form-group text-align-center row justify-content-center">
                  <label class="form-control-label col-lg-2" for="input-username">Username</label>
                  <input type="text" id="input-username" name="username" class="form-control col-lg-8" placeholder="Username" value="{{ user.username }}">
                </div>
                <div class="form-group text-align-center row justify-content-center">
                  <label class="form-control-label col-lg-2" for="input-password">Password</label>
                  <input type="password" id="input-password" name="password" class="form-control col-lg-8" placeholder="Masukkan Password">
                </div>
                <div class="d-flex justify-content-center">
                  <button type="submit" class="btn btn-primary me-1 mb-1">EDIT</button>
                </div>
              </div>
            </div>
          </form>
        </div>
    </div>
  </div>
</div>
{% endblock content %}
```

Lampiran 7 Kartu Kendali Bimbingan

KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

Nama Mahasiswa : Bulan Suci Cahayawati
NIM : 2011102441094
Nama Dosen Pembimbing : Sayekti Harits Suryawan., S.Kom., M.Kom
Judul Penelitian : Implementasi Model YOLOV8 Dalam Sistem Informasi Monitoring Kendaraan: Studi Kasus UMKT

No	Tanggal	Uraian Pembimbing	Paraf Dosen
1	19 Mar, 2024	Bimbingan terkait BAB I (menambahkan grafik data jumlah mahasiswa UMKT 5 tahun terakhir)	
2	5 Apr, 2024	Bimbingan lanjutan mengenai BAB I (konsultasi mengenai tahap pengujian aplikasi)	
3	29 Mei, 2024	Bimbingan terkait BAB II (memperbaiki sub bab dan membuat use case)	
4	13 Jun, 2024	Bimbingan lanjutan mengenai BAB II (membahas fitur fitur pada sistem informasi)	
5	21 Jun, 2024	Arahan dosen pembimbing mengenau batasan masalah, pengujian sistem informasi	
6	24 Mar, 2024	Bimbingan terkait pengumpulan proposal skripsi di website simpel	
7	26 Jun, 2024	Konsultasi terkait metode yang digunakan dalam pengujian data.	
8	27 Jun, 2024	Bimbingan terkait BAB III	
9	27 Jun, 2024	Bimbingan lanjutan mengenai BAB III dengan alur penelitian	
10	27 Jun, 2024	BAB IV mengenai Publish Jurnal	

Dosen Pembimbing

Sayekti Harits Suryawan, S.Kom., M.Kom
NIDN. 1119048901

Mengetahui,
Ketua Program Studi

Anggiyah, S.Kom., M.TI
NIDN. 1118019203

Lampiran 8 Hasil Uji Plagiasi Turnitin

Bulan Suci Cahayawati

by Teknik Informatika UMKT

Submission date: 24-Jul-2024 08:18AM (UTC+0800)
Submission ID: 2421526559
File name: skripsi_untuk_turnitin_-_BULAN_SUCI_CAHAYAWATI.docx (1.2M)
Word count: 3463
Character count: 23414

Bulan Suci Cahayawati

ORIGINALITY REPORT

16%	16%	6%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.umkt.ac.id Internet Source	1%
2	docplayer.info Internet Source	1%
3	123dok.com Internet Source	1%
4	dspace.uii.ac.id Internet Source	1%
5	ejurnal.unima.ac.id Internet Source	1%
6	ejournal.bsi.ac.id Internet Source	1%
7	repository.unwira.ac.id Internet Source	1%
8	repository.dinamika.ac.id Internet Source	1%
9	www.researchgate.net Internet Source	1%

Lampiran 9 Riwayat Hidup

Penulis dilahirkan di Samarinda pada tanggal 12 November 2003 sebagai anak ke sebelas dari dua belas bersaudara dari pasangan Sugeng dan Tamlika. Saat ini penulis tinggal di Jl. Kebon Agung Lempake Samarinda. Pendidikan sekolah di TK Darul Huda tahun 2008, SDN 009 Samarinda Kota tamat tahun 2014, SMPN 21 Samarinda Kota tamat tahun 2017, SMKN 7 Samarinda tamat tahun 2020. Saat ini Penulis sedang menempuh pendidikan kuliah di Universitas Muhammadiyah Kalimantan Timur Kota Samarinda, jurusan yang diambil adalahTeknik Informatika dan belajar di gedung Fakultas Sains dan Teknologi. Penulis pernah mengikuti program magang pada semester 7 di Universitas Muhammadiyah Kalimantan Timur di Unit Teknologi Informasi sebagai junior programmer. Saat ini penulis sedang menyelesaikan tugas skripsi.