



UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN
Gedung Rektorat Lantai 3,
Jl.Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

SURAT KETERANGAN HASIL PENGECEKAN TURNITIN

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : Membangun Sistem Parkir IOT Berbasis Web dan Mobile di
Puskesmas Kecamatan Cakung

Penulis : MUHAMMAD SUHAEMI

NIM : 2018230023

Tgl pemeriksaan : 21 Februari 2024

Dengan hasil Tingkat Kesamaan (*similarity index*) 25%

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 21 Februari 2024

Ka.UPT Perpustakaan Unsada



Yus Rusmiyati, SS., MM

Batas maksimal *similarity* 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal *similarity* 25% untuk Fakultas Teknik, Kelautan dan Pasca Sarjana



SURAT KETERANGAN

Nomor : 901 /KS.05

Yang bertanda tangan di bawah ini :

Nama : drg. Junaidah
NIP / NRK : 196507171992032009 / 162136
Pangkat / Gol : Pembina Tingkat I / IV/B
Jabatan : Kepala Puskesmas Cakung

Dengan ini menerangkan bahwa yang bersangkutan :

Nama : Muhammad Suhaemi
NIM : 2018230023
Perguruan Tinggi : Universitas Darma Persada
Judul Penelitian : Membangun Sistem Parkir IOT Berbasis Web dan Mobile di Puskesmas Cakung

Telah melaksanakan penelitian dan selesai melaksanakan tugas – tugasnya di Puskesmas Cakung pada 25 Januari 2024 s.d 5 Februari 2024 dengan kondite baik.

Demikian surat keterangan ini kami buat untuk dapat dipergunakan sebagaimana mestinya.

Dikeluarkan di Jakarta
pada tanggal 5 Februari 2024
Kepala Pusat Kesehatan Masyarakat Cakung,



drg. Junaidah

NIP. 196507171992032009



PEMERINTAH PROVINSI DAERAH KHUSUS IBUKOTA JAKARTA
DINAS KESEHATAN
SUKU DINAS KESEHATAN
KOTA ADMINISTRASI JAKARTA TIMUR
Jl. Matraman Raya No. 218. Telp. 021-8192202 Fax. 021-8506319
JAKARTA

Kode Pos : 13310

Nomor : 382/KS.02.03
Sifat : Biasa
Lampiran. :
Perihal : Izin Pengambilan Data

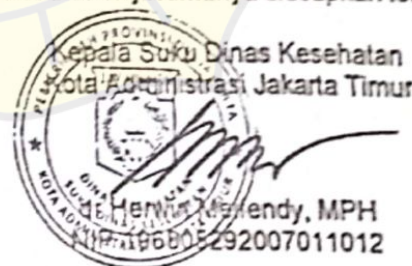
1 Februari 2024

Kepada
Yth. Wakil Dekan I
Fakultas Teknik
Universitas Darma Persada
di
Tempat

Sehubungan dengan pengajuan surat Nomor : 045/P/FT-Wakil Dekan I /KP-TA/2024 Tanggal 25 Januari 2024 Perihal : Permohonan Izin Melaksanakan Tugas Akhir bagi mahasiswa Program Studi Teknologi Informasi Fakultas Teknik Universitas Darma Persada untuk Penyusunan Tugas Akhir yang dilaksanakan di Wilayah Jakarta Timur. Maka dengan ini kami sampaikan hal-hal sebagai berikut:

1. Pada prinsipnya kami mengizinkan atas permohonan pengambilan data yang akan dilaksanakan di wilayah Jakarta Timur tanggal 5 Februari s.d 5 April 2024 dengan mengikuti semua aturan yang berlaku pada Puskesmas tersebut.
2. Apabila dalam pelaksanaan kegiatan terjadi tindakan yang tidak sesuai dengan SOP (Standart Operasional Prosedur) oleh mahasiswa / institusi, maka hal itu merupakan tanggung jawab mahasiswa dan institusi.
3. Lahan yang kami berikan untuk melaksanakan kegiatan tersebut adalah Puskesmas Cakung
4. Melaporkan kembali hasil pelaksanaan kegiatan tersebut kepada Suku Dinas Kesehatan Kota Administrasi Jakarta Timur melalui <https://bit.ly/LaporanPengambilandataJakartaTimur>
5. Mahasiswa yang melakukan praktek lapangan, pengambilan data dan penelitian di Puskesmas, membayar Retribusi sesuai dengan Peraturan Gubernur Provinsi Daerah Khusus Ibukota Jakarta Nomor 143 tahun 2018 tentang Tarif Pelayanan Pusat Kesehatan Masyarakat.
6. Nama Mahasiswa : Muhammad Suhaerni
NIM : 2018230023

Demikian kami sampaikan, atas perhatian dan kerjasamanya diucapkan terima kasih.



Tembusan Yth :
Kepala Puskesmas Cakung

Lampiran

```
import cv2
from pyzbar import pyzbar
import datetime
import serial
import time
import re # Import the regular expressions module
import pymysql

# Set up serial connection (update the port name to match your system)
ser = serial.Serial('COM4', 9600, timeout=1)
time.sleep(2)

last_scanned = {}

# Regular expression pattern for QR code data
pattern = r"username:;\nmidnumber:;"

parking_status = {"A1": "Available", "A2": "Available", "A3": "Available"}
signal_received = {"5": False, "6": False, "7": False}
last_check_time = time.time()

def get_next_id_transaksi(connection):
    cursor = connection.cursor()
    cursor.execute("SELECT MAX(id_transaksi) FROM transaksi parkir")
    result = cursor.fetchone()
    cursor.close()
    return (result[0] or 0) + 1 # Increment the maximum ID by 1, or start at 1

def insert_into_database(id_card, jam_masuk):
    # Database connection parameters - replace with your database info
    db_host = "sistemparkirjatinegara.my.id"
    db_user = "sisa7496"
    db_password = "6PCJw7K7JizB83"
    db_name = "sisa7496_sistemparkir"

    try:
        connection = pymysql.connect(host=db_host, user=db_user,
        password=db_password, database=db_name)

        cursor = connection.cursor()

        current_date = datetime.datetime.now().strftime("%Y-%m-%d")

        check_query = "SELECT COUNT(*) FROM transaksi parkir WHERE
        id_card = %s AND tanggal = %s AND jam_masuk = '00:00:00'"

        cursor.execute(check_query, (id_card, current_date))
```

```

result = cursor.fetchone()

if result[0] == 1:

    if result:
        id_user = result[0]
        next_id_transaksi = get_next_id_transaksi(connection)

        update_query = """
UPDATE transaksiparkir
SET tanggal = %s, jam_masuk = %s
WHERE id_card = %s
"""

        cursor.execute(update_query, (current_date, jam_masuk, id_card))
        connection.commit()

    else:
        print(f"No user found with id_card: {id_card}")

else:
    print(f"Check-in record already exists for id_card: {id_card} on date:
{current_date}")

cursor.close()
connection.close()
except Exception as e:
    print(f"Error: {e}")

def check_user_checked_in(id_card):
    db_host = "sistemparkirjatinegara.my.id"
    db_user = "sis7496"
    db_password = "6PCJw7K7JizB83"
    db_name = "sis7496_sistemparkir"

    try:
        connection = pymysql.connect(host=db_host, user=db_user,
password=db_password, database=db_name)
        cursor = connection.cursor()

        current_date = datetime.datetime.now().strftime("%Y-%m-%d")

        # SQL query to check for existing check-in
        check_query = """
SELECT COUNT(*) FROM transaksiparkir
WHERE id_card = %s AND tanggal = %s AND jam_masuk != '00:00:00'
AND totalharga = '0'
"""

```

```

cursor.execute(check_query, (id_card, current_date))
result = cursor.fetchone()

cursor.close()
connection.close()

return result[0] > 0 # Return True if a check-in record exists
except Exception as e:
    print(f"Error: {e}")
    return False

def update_parkir_status(nama_parkir, status_parkir):
    db_host = "sistemparkirjatinegara.my.id"
    db_user = "sis7496"
    db_password = "6PCJw7K7JizB83"
    db_name = "sis7496_sistemparkir"

    try:
        connection = pymysql.connect(host=db_host, user=db_user,
password=db_password, database=db_name)
        cursor = connection.cursor()

        update_query = """
UPDATE parkir
SET status_parkir = %s
WHERE nama_parkir = %s
"""
        cursor.execute(update_query, (status_parkir, nama_parkir))

        connection.commit()
        cursor.close()
        connection.close()
        print(f"Updated parkir status for {nama_parkir} to {status_parkir}")
    except Exception as e:
        print(f"Error: {e}")

def update_database_on_exit(id_card):
    db_host = "sistemparkirjatinegara.my.id"
    db_user = "sis7496"
    db_password = "6PCJw7K7JizB83"
    db_name = "sis7496_sistemparkir"

    try:
        connection = pymysql.connect(host=db_host, user=db_user,
password=db_password, database=db_name)
        cursor = connection.cursor()

# Get current time

```

```

current_time = datetime.datetime.now()
formatted_time = current_time.strftime('%H:%M:%S')
current_date = current_time.strftime('%Y-%m-%d')

# Get the latest harga_parkir
cursor.execute("SELECT harga_parkir FROM hargaparkir ORDER BY
id_harga DESC LIMIT 1")
harga_parkir = cursor.fetchone()[0]

# SQL query to update jam_keluar and calculate lama_parkir
update_query = """
UPDATE transaksiparkir
SET jam_keluar = %s, lama_parkir = CONCAT(HOUR(TIMEDIFF(%s,
jam_masuk)), ' jam'), totalharga = HOUR(TIMEDIFF(%s, jam_masuk)) * %s
WHERE id_card = %s AND tanggal = %s
"""

cursor.execute(update_query, (formatted_time, formatted_time,
formatted_time, harga_parkir, id_card, current_date))

# Retrieve the total_harga for the user
cursor.execute("SELECT totalharga FROM transaksiparkir WHERE id_card
= %s AND tanggal = %s", (id_card, current_date))
total_harga = cursor.fetchone()[0]

# Update the user's saldo
update_saldo_query = """
UPDATE user
SET saldo = saldo - %s
WHERE id_card = %s
"""

cursor.execute(update_saldo_query, (total_harga, id_card))

connection.commit()
cursor.close()
connection.close()
except Exception as e:
    print(f"Error: {e}")

def scan_qr_codes(frame):
    global last_scanned
    qr_codes = pyzbar.decode(frame)
    for qr_code in qr_codes:
        qr_data = qr_code.data.decode('utf-8')
        current_time = datetime.datetime.now()

```

```

if qr_data in last_scanned and (current_time - last_scanned[qr_data]).seconds
3600:
    continue

last_scanned[qr_data] = current_time
print(f"Decoded QR Code: {qr_data}")

# Check if the QR code matches the pattern
if re.match(pattern, qr_data):

    # Extract idnumber from QR code data
    idnumber = re.search(r"idnumber:(+)", qr_data).group(1)

    # Format current time for database insertion (only time, no date)
    formatted_time = current_time.strftime("%H:%M:%S")

    # Insert data into the database
    if check_user_checked_in(idnumber):
        print("User checked in. Updating checkout record...") # Debugging
statement
        update_database_on_exit(idnumber)
        ser.write(b'4') # Send command to move servo for checkout
        last_scanned.pop(qr_data, None)
    else:
        print("User has not checked in today. Inserting into database...") #
Debugging statement
        insert_into_database(idnumber, formatted_time)
        ser.write(b'2') # Send command to move servo for check-in
        last_scanned.pop(qr_data, None)

def main():
    global signal_received, last_check_time
    camera = cv2.VideoCapture(0)
    try:
        while True:
            current_time = time.time()
            # Check for serial input from Arduino
            if ser.in_waiting > 0:
                line = ser.readline().decode('utf-8').strip()
                print(f"Received Signal: {line}") # Print the received signal for
debugging

                if line in ["5", "6", "7"]:
                    signal_received[line] = True
                    spot = {"5": "A1", "6": "A2", "7": "A3"}[line]
                    parking_status[spot] = "Occupied"

```



```
update_parkir_status(spot, "Occupied")

# Update the status to 'Available' if no signal is received for more than 10
seconds
if current_time - last_check_time > 10:
    for signal, received in signal_received.items():
        if not received:
            spot = {"5": "A1", "6": "A2", "7": "A3"}[signal]
            if parking_status[spot] != "Available":
                parking_status[spot] = "Available"
                update_parkir_status(spot, "Available")
        # Reset the signals and update the last check time
        signal_received = {"5": False, "6": False, "7": False}
        last_check_time = current_time

# Check for QR code scanning
ret, frame = camera.read()
if ret:
    scan_qr_codes(frame)
    cv2.imshow("QR Code Scanner", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
finally:
    camera.release()
    cv2.destroyAllWindows()
    ser.close()

if __name__ == "__main__":
    main()
```



LEMBAR REVISI - SIDANG SKRIPSI

NIM>Nama : 2018230023 Muhammad Suhaemi
Fakultas/Prodi : Teknik / Teknologi Informasi

No.	Keterangan Revisi	Dosen
1.	Penulisan, format penulisan & perbaikan sesuai dg panduan penulisan	Mr. Adnan Ibu. Linda
2.	perbaiki format penulisan	
3.	laporan tidak sesuai format penulisan	P. An 14/2-2. F 19/2/24.

Mengetahui,

Ka Prodi Teknologi Informasi

Herianto, S.Pd., MT.