DAFTAR PUSTAKA

Azizi, F. A. A., 2021. *Perancangan Web-Gis Ruang Terbuka Hijau Kota Binjai.* Skripsi ed. Medan: Departemen Manajemen Hutan Fakultas Kehutanan Universitas Sumatera Utara.

Clinton, R. M. R. & Sengkey , R., 2019. Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer Raspberry Pi. *Teknik Elektro dan Komputer,* 8(3), pp. 181-192.

Dinda, N. F. & Setiyawati, N., 2021. Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request. *Jurnal Janitra Informatika dan Sistem Informasi,* 1(1), pp. 19-34.

Endra, R. Y. & Aprilita, D. S., 2018. E-Report Berbasis Web Menggunakan Metode Model View Controller Untuk Mengetahui Peningkatan Perkembangan Prestasi Anak Didik. *Jurnal Sistem Informasi dan Telematika,* 9(1), pp. 15-22.

Harismawan, A. F., Kharisma, A. P. & Afirianto, T., 2018. Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada Client Berbasis Android. *Pengembangan Teknologi Informasi dan Ilmu Komputer,* 2(1), pp. 237-245.

Hasanuddin, A. A., 2020. *Rancang Bangun Web-Gis Berbasis Geodjango-Python.* Skripsi ed. Makassar: Departemen Geofisika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Hutagalung, C., 2008. *Analisis Hubungan Antara Tingkat Experience Konsumen Dengan Tingkat Loyalitas Konsumen.* Skripsi ed. Yogyakarta: Program Studi Manajemen Jurusan Manajemen Universitas Sanata Dharma.

Irsyad, R., 2018. *Penggunaan Python Web Framework Flask Untuk Pemula.* Bandung: Laboratorium Telematika, Sekolah Teknik Elektro & Informatika, Institut Teknologi Bandung.

Maulana, H., 2016. Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source. *InfoTekJar,* 1(1), pp. 32-37.

Mertha, M. P., Simadiputra, V., Setyawan, E. & S., 2019. Implementasi WebGIS untuk Pemetaan Objek Wisata Kota Jakarta Barat dengan Metode Location Based Service menggunakan Google Maps API. *InfoTekJar,* Volume 4, pp. 22-28.

Monoarfa, S., 2020. *Metadata Indikator Tujuan Pembangunan Berkelanjutan (Tpb)/Sustainable Development Goals (Sdgs) Indonesia.* Jakarta: Kementerian Perencanaan Pembangunan Nasional/Badan Perencanaan Pembangunan Nasiona.

Mulyahati, I. L., 2020. *Implementasi Machine Learning Prediksi Harga Sewa Apartemen Menggunakan Algoritma Random Forest Melalui Framework Website Flask Python.* Skripsi ed. Yogyakarta: Program Studi Statistika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Islam Indonesia.

Nuristiqomah, R. P. & Anistyasari, Y., 2021. Pengembangan Kamus Istilah Basis Data Berbasis Website Menggunakan Algoritma Cosine Similarity Untuk Meningkatkan Hasil Belajar Siswa. *It-Edu.,* 5(2), pp. 621-630.

Pakereng, M. A. I. & Ngantung, R. K., 2021. Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python. *Jurnal Media Informatika Budidarma,* 5(3), pp. 1051-1062.

Prehandayana, G., Yahya, W. & Nurwarsito, H., 2018. Implementasi Struktur Data Dictionary Untuk Sistem Monitoring Perangkat Internet Of Things. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer ,* 2(10), pp. 3466-3473.

Ridwan, M., Arifin, Z. & Y., 2016. Rancang Bangun E-Voting Dengan Menggunakan Keamanan Algoritma Rivest Shamir Adleman (Rsa) Berbasis Web (Studi Kasus : Pemilihan Ketua Bem Fmipa). *Jurnal Informatika Mulawarman,* 11(2), pp. 22-28.

Rudiyanto, A., 2020. *Pedoman Teknis Penyusunan Rencana Aksi Tujuan Pembangunan Berkelanjutan (Tpb)/Sustainable Development Goals (Sdgs).* Jakarta: Kementerian Perencanaan Pembangunan Nasional / Badan Perencanaan Pembangunan Nasional.

Siregar, J., 2018. *Pengembangan Aplikasi Pendaftaran Online Layanan Pencatatan Sipil Berbasis Web Menggunakan Php Dan Basis Data Mysql.* Skripsi ed. Malang: Program Studi Sistem Informasi Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.

Sovia, R. & Febio, J., 2011. Membangun Aplikasi E-Library Menggunakan Html, Php Script, Dan Mysql Database. *Processor,* 6(2), pp. 38-54.

Suryamen, H. & Hsb, H., 2017. *Pembangunan Sistem Informasi Komoditi Berbasis Webgis Untuk Pertanian Perkebunan Dan Kehutanan Daerah Tanjung Raya Maninjau.* Jakarta: Sistem Infomasi Fakultas Teknik Universitas Muhammadiyah.

Wirawati, I., 2017. *Pemodelan Pengeluaran Per Kapita Rumah Tangga Di Maluku Utara Menggunakan Struktur Hirarki Dua Tingkat Dengan Pendekatan Bayesian.* Tesis ed. Surabaya: Program Magister Jurusan Statistika Fakultas Matematika Dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember.

ZAINI, I. F., 2021. *Kebijakan Sustainable Development Goals (Sdgs) Dalam Penanggulangan Kemiskinan Masyarakat Desa.* Skripsi ed. Makassar: Departemen Sosiologi Fakultas Ilmu Sosial Dan Ilmu Politik Universitas Hasanuddin.

LAMPIRAN

LAMPIRAN 1. Script Python app.py

```
from app import app

# python -m venv env 1

# env\scripts\activate 2
# set FLASK_APP=app.py
# set FLASK_DEBUG=1
# flask run

# pip freeze > requirements.txt
# pip install -r requirements.txt 3

# >>> open file "config.inc.php"

# >>> py
# >>> from app import db, create_app, models
# >>> db.create_all(app=create_app())
```

LAMPIRAN 2. Script Python main.py

```
import os
from msilib import add_data
import numpy as np
import folium
import patoolib
import folium
import json
import geopandas as gpd
from folium.plugins import MousePosition
from flask import Blueprint, render_template, request, redirect,
url_for, flash
from flask_login import login_required, current_user
from .models import Kuisioner_Individu, User, data_peta
from . import db

main = Blueprint('homepage', __name__)

@main.route("/")
def home():
    if current_user.is_authenticated:
        return render_template("homepage.html",
name=current_user.nama)
```

```
    return render_template("homepage.html")



@main.route("/dashboard")
@login_required
def index():
    Kuisindiv = Kuisioner_Individu.query.all()

    data = '600000'
    minus =
int(Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Peng
eluaran_Perkapita <= data).count())
    plus =
int(Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Peng
eluaran_Perkapita > data).count())
    totaldatkus = Kuisioner_Individu.query.count()

    # PerKem = Persentase Kemiskinan
    # analisis persentase pengeluaran perkapita diatas
kemiskinan
    perkemplus = plus / Kuisioner_Individu.query.count()*100

    # analisis persentase pengeluaran perkapita dibawah kemiskinan
    perkemminus = minus / Kuisioner_Individu.query.count()*100

    user = User.query.all()


    return render_template('index.html',
Kuisioner_Individu=Kuisindiv, totaldatkus=totaldatkus,
perkemplus=perkemplus, perkemminus=perkemminus, Minus=minus,
Plus=plus, user=user)


@main.route("/base")
@login_required
def base():
    user = User.query.all()
    return render_template('base.html', user=user)

@main.route("/forms", methods=['GET', 'POST'])
@main.route("/forms/<id>", methods=['GET','POST'])
@login_required
def forms(id=None):
```

```
    if id:
        update = Kuisioner_Individu.query.filter_by(id=id).first()
        Provinsi = data_peta.query.all()

        return render_template('forms.html', row = update,
id=id,Provinsi=Provinsi)
    else:
        if request.method == 'POST' :
            edit = request.form.get('edit')
            print (edit)
            if edit:
                update =
Kuisioner_Individu.query.get(request.form.get('edit'))

                # Analisis Pengeluaran Perkapita
                ldata = int(update.Pengeluaran_Sebulan_Terakhir) /
int(update.Jumlah_Anggota_Keluarga)
                update.Analisis_Pengeluaran_Perkapita=ldata

                update.Nomor_KK=request.form['NoKK']
                update.Jumlah_Anggota_Keluarga=request.form['JumlahA
K']
                update.Nama_Kepala_Keluarga=request.form['NamaKK']
                update.Jenis_Kelamin_Keluarga=request.form['JKKK']
                update.NIK=request.form['NIK']
                update.Nama_Lengkap=request.form['Namalkp']
                update.Jenis_Kelamin=request.form['JK']
                update.Tempat_Lahir=request.form['tempatlhr']
                update.Tanggal_Lahir=request.form['tgllhr']
                # tambahan alamat
                update.Alamat_Tempat_Tinggal=request.form['alamatttg
']
                update.Provinsi=request.form['provinsi']
                update.Kota_Kabupaten=request.form['kotkab']
                update.Kecamatan=request.form['kec']
                update.Kelurahan_Atau_Desa=request.form['keldes']



                update.Status_Perkawinan=request.form['Skwn']
                update.Agama=request.form['Agama']
                update.Suku=request.form['Suku']
                update.Warga_Negara=request.form['warganeg']
                update.Nomor_Telp=request.form['notelp']
                update.Nomor_Wa=request.form['nowa']
```

```
                update.Email=request.form['email']
                update.Media_Sosial=request.form['akunmedsos']

                # elif request.form ['submit'] =='deskjob':
                update.Kondisi_Pekerjaan=request.form['konjob']
                update.Pekerjaan_Utama=request.form['jobut']
                update.Jaminan_Sosial_Ketenagakerjaan=request.form['
bpjs']
                update.Penghasilan_Sebulan_Terakhir=request.form['pe
ng1bln']
                update.Pengeluaran_Sebulan_Terakhir=request.form['pe
nlu1bln']
                update.Kepemilikan_Rumah=request.form['keprmh']
                update.Kepemilikan_Lahan_Luas=request.form['keplhn']

                # elif request.form ['submit'] =='deskkes':
                update.Penyakit_Setahun_Terakhir=request.form['penya
kit']
                update.Fasilitas_Kesehatan_Dikunjungi=request.form['
faskes']
                update.Jumlah_Berapa_Kali_Fasilitas_Kesehatan_Dikunj
ungi=request.form['jmlhfaskes']
                update.Jaminan_Kesehatan_Asuransi_KIS=request.form['
kis']
                update.Disabilitas=request.form['disabil']

                # elif request.form ['submit'] =='deskpend':
                update.Pendidikan_Terakhir=request.form['pendidk']
                update.Bahasa_Digunakan_Dirumah=request.form['bhs']
                update.Bahasa_Digunakan_Disekolah_Kantor_Tempat_Kerj
a=request.form['bhsskl']
                update.Kerja_Bakti_Setahun_Terakhir=request.form['kr
jbkti']
                update.Siskamling_Setahun_Terakhir=request.form['sis
kaeee']
                update.Pesta_Rakyat_atau_Adat_Terakhir_Dilaksanakan=
request.form['pestaryt']
                update.Menolong_warga_Mengalami_Kematian_Setahun_Ter
akhir=request.form['tlngdet']
                update.Menolong_warga_Mengalami_Sakit_Setahun_Terakh
ir=request.form['tlngskt']
                update.Menolong_warga_Mengalami_Kecelakaan_Setahun_T
erakhir=request.form['tlnglaka']

                db.session.commit()
```

```
                flash("Data berhasil diubah")

                return redirect(url_for('homepage.tables'))

            # if request.form ['submit'] =='deskindiv':
            Nomor_KK=request.form['NoKK']
            Jumlah_Anggota_Keluarga=request.form['JumlahAK']
            Nama_Kepala_Keluarga=request.form['NamaKK']
            Jenis_Kelamin_Keluarga=request.form['JKKK']
            NIK=request.form['NIK']
            Nama_Lengkap=request.form['Namalkp']
            Jenis_Kelamin=request.form['JK']
            Tempat_Lahir=request.form['tempatlhr']
            Tanggal_Lahir=request.form['tgllhr']
            # tambahan alamat
            Alamat_Tempat_Tinggal=request.form['alamatttg']
            Provinsi=request.form['provinsi']
            Kota_Kabupaten=request.form['kotkab']
            Kecamatan=request.form['kec']
            Kelurahan_Atau_Desa=request.form['keldes']

            Status_Perkawinan=request.form['Skwn']
            Agama=request.form['Agama']
            Suku=request.form['Suku']
            Warga_Negara=request.form['warganeg']
            Nomor_Telp=request.form['notelp']
            Nomor_Wa=request.form['nowa']
            Email=request.form['email']
            Media_Sosial=request.form['akunmedsos']

            # elif request.form ['submit'] =='deskjob':
            Kondisi_Pekerjaan=request.form['konjob']
            Pekerjaan_Utama=request.form['jobut']
            Jaminan_Sosial_Ketenagakerjaan=request.form['bpjs']
            Penghasilan_Sebulan_Terakhir=request.form['peng1bln']
            Pengeluaran_Sebulan_Terakhir=request.form['penlu1bln']
            Kepemilikan_Rumah=request.form['keprmh']
            Kepemilikan_Lahan_Luas=request.form['keplhn']

            # elif request.form ['submit'] =='deskkes':
            Penyakit_Setahun_Terakhir=request.form['penyakit']
            Fasilitas_Kesehatan_Dikunjungi=request.form['faskes']
            Jumlah_Berapa_Kali_Fasilitas_Kesehatan_Dikunjungi=reques
t.form['jmlhfaskes']
            Jaminan_Kesehatan_Asuransi_KIS=request.form['kis']
```

```
            Disabilitas=request.form['disabil']

            # elif request.form ['submit'] =='deskpend':
            Pendidikan_Terakhir=request.form['pendidk']
            Bahasa_Digunakan_Dirumah=request.form['bhs']
            Bahasa_Digunakan_Disekolah_Kantor_Tempat_Kerja=request.f
orm['bhsskl']
            Kerja_Bakti_Setahun_Terakhir=request.form['krjbkti']
            Siskamling_Setahun_Terakhir=request.form['siskaeee']
            Pesta_Rakyat_atau_Adat_Terakhir_Dilaksanakan=request.for
m['pestaryt']
            Menolong_warga_Mengalami_Kematian_Setahun_Terakhir=reque
st.form['tlngdet']
            Menolong_warga_Mengalami_Sakit_Setahun_Terakhir=request.
form['tlngskt']
            Menolong_warga_Mengalami_Kecelakaan_Setahun_Terakhir=req
uest.form['tlnglaka']

            # Analisis Pengeluaran Perkapita
            ldata = int(Pengeluaran_Sebulan_Terakhir) /
int(Jumlah_Anggota_Keluarga)


            add_data_Kuisioner_Individu =
Kuisioner_Individu(Analisis_Pengeluaran_Perkapita=ldata,
                                                Nomor_KK=Nomor_KK,
                                                Jumlah_Anggota_Keluarga=
Jumlah_Anggota_Keluarga,
                                                Nama_Kepala_Keluarga=Nam
a_Kepala_Keluarga,
                                                Jenis_Kelamin_Keluarga=J
enis_Kelamin_Keluarga,
                                                NIK=NIK,
                                                Nama_Lengkap=Nama_Lengka
p,
                                                Jenis_Kelamin=Jenis_Kela
min,
                                                Tempat_Lahir=Tempat_Lahi
r,
                                                Tanggal_Lahir=Tanggal_La
hir,
                                                # tambahan alamat
                                                Alamat_Tempat_Tinggal=Al
amat_Tempat_Tinggal,
                                                Provinsi=Provinsi,
```

```
                                                Kota_Kabupaten=Kota_Kabu
paten,
                                                Kecamatan=Kecamatan,
                                                Kelurahan_Atau_Desa=Kelu
rahan_Atau_Desa,




                                                Status_Perkawinan=Status
_Perkawinan,
                                                Agama=Agama,
                                                Suku=Suku,
                                                Warga_Negara=Warga_Negar
a,
                                                Nomor_Telp=Nomor_Telp,
                                                Nomor_Wa=Nomor_Wa,
                                                Email=Email,
                                                Media_Sosial=Media_Sosia
l,
                                                # Deskjob
                                                Kondisi_Pekerjaan=Kondis
i_Pekerjaan,
                                                Pekerjaan_Utama=Pekerjaa
n_Utama,
                                                Jaminan_Sosial_Ketenagak
erjaan=Jaminan_Sosial_Ketenagakerjaan,
                                                Penghasilan_Sebulan_Tera
khir=Penghasilan_Sebulan_Terakhir,
                                                Pengeluaran_Sebulan_Tera
khir=Pengeluaran_Sebulan_Terakhir,
                                                Kepemilikan_Rumah=Kepemi
likan_Rumah,
                                                Kepemilikan_Lahan_Luas=K
epemilikan_Lahan_Luas,
                                                # Deskkes
                                                Penyakit_Setahun_Terakhi
r=Penyakit_Setahun_Terakhir,
                                                Fasilitas_Kesehatan_Diku
njungi=Fasilitas_Kesehatan_Dikunjungi,
                                                Jumlah_Berapa_Kali_Fasil
itas_Kesehatan_Dikunjungi=Jumlah_Berapa_Kali_Fasilitas_Kesehatan_Dik
unjungi,
                                                Jaminan_Kesehatan_Asuran
si_KIS=Jaminan_Kesehatan_Asuransi_KIS,
```

```
                                                Disabilitas=Disabilitas,
                                                # Deskpend
                                                Pendidikan_Terakhir=Pend
idikan_Terakhir,
                                                Bahasa_Digunakan_Dirumah
=Bahasa_Digunakan_Dirumah,
                                                Bahasa_Digunakan_Disekol
ah_Kantor_Tempat_Kerja=Bahasa_Digunakan_Disekolah_Kantor_Tempat_Kerj
a,
                                                Kerja_Bakti_Setahun_Tera
khir=Kerja_Bakti_Setahun_Terakhir,
                                                Siskamling_Setahun_Terak
hir=Siskamling_Setahun_Terakhir,
                                                Pesta_Rakyat_atau_Adat_T
erakhir_Dilaksanakan=Pesta_Rakyat_atau_Adat_Terakhir_Dilaksanakan,
                                                Menolong_warga_Mengalami
_Kematian_Setahun_Terakhir=Menolong_warga_Mengalami_Kematian_Setahun
_Terakhir,
                                                Menolong_warga_Mengalami
_Sakit_Setahun_Terakhir=Menolong_warga_Mengalami_Sakit_Setahun_Terak
hir,
                                                Menolong_warga_Mengalami
_Kecelakaan_Setahun_Terakhir=Menolong_warga_Mengalami_Kecelakaan_Set
ahun_Terakhir,

                                                )


            db.session.add(add_data_Kuisioner_Individu)

            db.session.commit()
            flash("Data Telah Dimasukkan :)")

        user = User.query.all()

        # Data Peta
        Provinsi = data_peta.query.all()

        return render_template('forms.html', user=user,
Provinsi=Provinsi)

@main.route("/tables", methods=['GET','POST'])
@login_required
def tables():
    Kuisindiv = Kuisioner_Individu.query.all()
```

```
    if request.method == 'POST' :
        edit = request.form.get('ubah')
        hapus = request.form.get('hapus')
        if edit:
            id = request.form.get('ubah')
            return redirect(url_for('homepage.forms', id=id))
        elif hapus:
            delete = Kuisioner_Individu.query.get(hapus)

            db.session.delete(delete)
            db.session.commit()
            return redirect(url_for('homepage.tables',
Kuisioner_Individu=Kuisindiv))

    user = User.query.all()

    return render_template('tables.html',
Kuisioner_Individu=Kuisindiv, user=user)


@main.route("/hasil_analisis")
@login_required
def hasil_analisis():
    Kuisindiv = Kuisioner_Individu.query.all()

    data = '600000'
    minus =
Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Pengelua
ran_Perkapita <= data)
    plus =
Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Pengelua
ran_Perkapita > data)

    user = User.query.all()
    return render_template('hasil_analisis.html',
Kuisioner_Individu=Kuisindiv, Minus=minus, Plus=plus, user=user)


@main.route("/hasil_analisis_persentase")
@login_required
def hasil_analisis_persentase():
    Kuisindiv = Kuisioner_Individu.query.all()

    data = '600000'
```

```
    minus =
int(Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Peng
eluaran_Perkapita <= data).count())
    plus =
int(Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Peng
eluaran_Perkapita > data).count())

    # totaldatkus = Total  Data Kuisioner Individu
    totaldatkus = Kuisioner_Individu.query.count()
    # PerKem = Persentase Kemiskinan
    # analisis persentase pengeluaran perkapita diatas
kemiskinan
    perkemplus = plus / Kuisioner_Individu.query.count()*100

    # analisis persentase pengeluaran perkapita dibawah kemiskinan
    perkemminus = minus / Kuisioner_Individu.query.count()*100

    user = User.query.all()

    return render_template('hasil_analisis_persentase.html',
Kuisioner_Individu=Kuisindiv, totaldatkus=totaldatkus,
perkemplus=perkemplus, perkemminus=perkemminus, Minus=minus,
Plus=plus, user=user)




@main.route("/input_data_shp",methods=['GET','POST'])
@main.route("/input_data_shp/<id>",methods=['GET','POST'])
@login_required
def input_data_shp(id=None):
    datapet=data_peta.query.all()
    if id :
            update=data_peta.query.filter_by(id_datapeta=id).first()
            return render_template('input_data_shp.html', row =
update, id=id)
    if request.method == 'POST':
        if id :
            update=data_peta.query.filter_by(id_datapeta=id).first()
            return render_template('input_data_shp.html', row =
update, id=id)
        else :
            edit = request.form.get('ubah')
            hapus = request.form.get('hapus')
```

```
            if edit:
                id = request.form.get('ubah')
                return redirect(url_for('homepage.input_data_shp',
id=id,))
            elif hapus:
                delete = data_peta.query.get(hapus)

                db.session.delete(delete)
                db.session.commit()
                return redirect(url_for('homepage.input_data_shp',
datapet=data_peta))

            Provinsi = request.form['prov']
            KotaKabupaten = request.form['kotkab']
            Kecamatan = request.form['kec']
            Desa = request.form['des']
            file = request.files['json']

            dir = 'app/static/json/temporary'
            if file:
                namefile = file.filename
                json = "{}.json".format(namefile)

                file.save(os.path.join("app/static/json/temporary",
"{}".format(namefile)))
                patoolib.extract_archive("app/static/json/temporary/
{}".format(namefile), outdir="app/static/json/temporary/")

                search_shp = [f for f in os.listdir(dir) if
f.endswith(".shp")]

                gjson =
gpd.read_file("app/static/json/temporary/{}".format(search_shp[0]))
                gjson.to_file("app/static/json/temporary/{}".format(
json), driver='GeoJSON')
                gjson = gjson.to_json()
            else:
                gjson = None

            add_Data = data_peta( Nama_File_SHP = namefile,
                                  Provinsi = Provinsi,
                                  KotaKabupaten = KotaKabupaten,
                                  Kecamatan = Kecamatan,
                                  Desa = Desa,
                                  Type_SHP = gjson)
```

```
            db.session.add(add_Data)
            db.session.commit()

            for f in os.listdir(dir):
                os.remove(os.path.join(dir, f))

            flash("Data berhasil ditambahkan")

            return redirect(url_for('homepage.input_data_shp'))
# def input_data_shp():
    user = User.query.all()
    return render_template('input_data_shp.html', data_peta=datapet,
user=user)



@main.route("/login")
@login_required
def login():
    return render_template('login.html')

@main.route("/register")
@login_required
def register():
    return render_template('register.html')

@main.route("/usermanager")
@login_required
def usermanager():
    user = User.query.all()
    return render_template('usermanager.html', user=user)

@ main.route("/gis", methods=['GET', 'POST'])
@ main.route("/gis/<id>", methods=['GET', 'POST'])
def gis(id=None):
    if id:
        data = data_peta.query.filter_by(id_datapeta=id).first()

        layer = gpd.read_file(data.Type_SHP)
        layer = layer.to_crs("EPSG:4326")

        m = folium.Map(location=[-1.1265694, 118.6380067],
                   zoom_start=5, min_zoom=5)
```

```
        for x in layer.index:
            color = np.random.randint(16, 256, size=3)
            # color= "green"
            color = [str(hex(i))[2:] for i in color]
            color = '#'+''.join(color).upper()
            layer.at[x, 'color'] = color

        def style(feature):
            return {
                'fillColor': feature['properties']['color'],
                'color': feature['properties']['color'],
                'weight': 1,
                'fillOpacity': 0.7
            }

        gjson = folium.GeoJson(layer, name=data.Nama_File_SHP,
style_function=style).add_to(m)

        m.fit_bounds(gjson.get_bounds())

        # folium.Popup(data.desa).add_to(gjson)

        formatter = "function(num) {return L.Util.formatNum(num, 3)
+ ' º ';};"
        MousePosition(
            position="bottomleft",
            separator=" | ",
            empty_string="NaN",
            lng_first=True,
            num_digits=20,
            prefix="Kordinat:",
            lat_formatter=formatter,
            lng_formatter=formatter,
        ).add_to(m)

        tile_layer = folium.TileLayer(
            tiles="http://www.google.cn/maps/vt?lyrs=s@189&gl=cn&x={
x}&y={y}&z={z}",
            attr='google.com',
            max_zoom=19,
            name='darkmatter',
            control=False,
            opacity=1
        )
        tile_layer.add_to(m)
```

```
        m.save("app/templates/gis-maps.html")


        # Persentase Kemiskinan
        Kuisindiv = Kuisioner_Individu.query.all()

        data = '600000'
        minus =
int(Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Peng
eluaran_Perkapita <= data).count())
        plus =
int(Kuisioner_Individu.query.filter(Kuisioner_Individu.Analisis_Peng
eluaran_Perkapita > data).count())
        totaldatkus = Kuisioner_Individu.query.count()

        # PerKem = Persentase Kemiskinan
        # analisis persentase pengeluaran perkapita diatas
kemiskinan
        perkemplus = plus / Kuisioner_Individu.query.count()*100

        # analisis persentase pengeluaran perkapita dibawah
kemiskinan
        perkemminus = minus / Kuisioner_Individu.query.count()*100



        # Data SHP

        datapet = data_peta.query.filter_by(id_datapeta=id).first()

        return render_template("maps.html",
Kuisioner_Individu=Kuisindiv, totaldatkus=totaldatkus,
perkemplus=perkemplus, perkemminus=perkemminus, Minus=minus,
Plus=plus, datapet=datapet, data=data, id=id)

    m = folium.Map(location=[-1.1265694, 118.6380067],
                   zoom_start=5, min_zoom=5)

    formatter = "function(num) {return L.Util.formatNum(num, 3) + '
º ';};"
    MousePosition(
        position="bottomleft",
        separator=" | ",
        empty_string="NaN",
```

```
        lng_first=True,
        num_digits=20,
        prefix="Kordinat:",
        lat_formatter=formatter,
        lng_formatter=formatter,
    ).add_to(m)

    tile_layer = folium.TileLayer(
        tiles="http://www.google.cn/maps/vt?lyrs=s@189&gl=cn&x={x}&y
={y}&z={z}",
        attr='google.com',
        max_zoom=19,
        name='darkmatter',
        control=False,
        opacity=1
    )
    tile_layer.add_to(m)

    m.save("app/templates/gis-maps.html")

    return render_template("maps.html")
```

LAMPIRAN 3. Script Python models.py

```
from flask_login import UserMixin
from datetime import datetime
from . import db


class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.VARCHAR(200), unique=True)
    password = db.Column(db.VARCHAR(200))
    nama = db.Column(db.VARCHAR(200))
    nomorhp = db.Column(db.VARCHAR(200))
    lvl = db.Column(db.VARCHAR(200))


class Kuisioner_Individu(db.Model):
# class Deskripsi_Individu(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    Analisis_Pengeluaran_Perkapita = db.Column(db.Integer)
    Nomor_KK = db.Column(db.Integer)
    Jumlah_Anggota_Keluarga = db.Column(db.Integer)
```

```
    Nama_Kepala_Keluarga = db.Column(db.VARCHAR(200))
    Jenis_Kelamin_Keluarga = db.Column(db.VARCHAR(200))
    NIK = db.Column(db.Integer)
    Nama_Lengkap = db.Column(db.VARCHAR(200))
    Jenis_Kelamin = db.Column(db.VARCHAR(200))
    Tempat_Lahir = db.Column(db.VARCHAR(200))
    Tanggal_Lahir = db.Column(db.VARCHAR(200))
    # tambahan alamat
    Alamat_Tempat_Tinggal = db.Column(db.VARCHAR(200))
    Provinsi = db.Column(db.VARCHAR(200))
    Kota_Kabupaten = db.Column(db.VARCHAR(200))
    Kecamatan = db.Column(db.VARCHAR(200))
    Kelurahan_Atau_Desa= db.Column(db.VARCHAR(200))
    Status_Perkawinan = db.Column(db.VARCHAR(200))
    Agama = db.Column(db.VARCHAR(200))
    Suku = db.Column(db.VARCHAR(200))
    Warga_Negara = db.Column(db.VARCHAR(200))
    Nomor_Telp = db.Column(db.VARCHAR(200))
    Nomor_Wa = db.Column(db.VARCHAR(200))
    Email = db.Column(db.VARCHAR(200))
    Media_Sosial = db.Column(db.VARCHAR(200))

# class Deskripsi_Pekerjaan(db.Model):
    Kondisi_Pekerjaan = db.Column(db.VARCHAR(200))
    Pekerjaan_Utama = db.Column(db.VARCHAR(200))
    Jaminan_Sosial_Ketenagakerjaan = db.Column(db.VARCHAR(200))
    Penghasilan_Sebulan_Terakhir = db.Column(db.VARCHAR(200))
    Pengeluaran_Sebulan_Terakhir = db.Column(db.VARCHAR(200))
    Kepemilikan_Rumah = db.Column(db.VARCHAR(200))
    Kepemilikan_Lahan_Luas = db.Column(db.Integer)

# class Deskripsi_Kesehatan(db.Model):
    Penyakit_Setahun_Terakhir = db.Column(db.VARCHAR(200))
    Fasilitas_Kesehatan_Dikunjungi = db.Column(db.VARCHAR(200))
    Jumlah_Berapa_Kali_Fasilitas_Kesehatan_Dikunjungi =
db.Column(db.Integer)
    Jaminan_Kesehatan_Asuransi_KIS = db.Column(db.VARCHAR(200))
    Disabilitas = db.Column(db.VARCHAR(200))

# class Deskripsi_Pendidikan(db.Model):
    Pendidikan_Terakhir = db.Column(db.VARCHAR(200))
    Bahasa_Digunakan_Dirumah = db.Column(db.VARCHAR(200))
    Bahasa_Digunakan_Disekolah_Kantor_Tempat_Kerja =
db.Column(db.VARCHAR(200))
    Kerja_Bakti_Setahun_Terakhir = db.Column(db.Integer)
```

```
    Siskamling_Setahun_Terakhir = db.Column(db.Integer)
    Pesta_Rakyat_atau_Adat_Terakhir_Dilaksanakan =
db.Column(db.Integer)
    Menolong_warga_Mengalami_Kematian_Setahun_Terakhir =
db.Column(db.Integer)
    Menolong_warga_Mengalami_Sakit_Setahun_Terakhir =
db.Column(db.Integer)
    Menolong_warga_Mengalami_Kecelakaan_Setahun_Terakhir =
db.Column(db.Integer)


class data_peta(UserMixin, db.Model):
    id_datapeta = db.Column(db.Integer, primary_key=True)
    Nama_File_SHP = db.Column(db.VARCHAR(200))
    Provinsi = db.Column(db.VARCHAR(200))
    KotaKabupaten = db.Column(db.VARCHAR(200))
    Kecamatan = db.Column(db.VARCHAR(200))
    Desa = db.Column(db.VARCHAR(200))
    Type_SHP = db.Column(db.JSON)
```

LAMPIRAN 4. Script Python config.py

```
SECRET_KEY = "tacokkk"
SQLALCHEMY_DATABASE_URI = "mysql://root:@localhost/sdgs_desa"
SQLALCHEMY_ECHO = False
SQLALCHEMY_TRACK_MODIFICATIONS = False
# max_allowed_packet = 128M <----- In 'my.ini' Server MySQL
```

LAMPIRAN 5. Script Python init.py

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager

db = SQLAlchemy()

def create_app():
    app = Flask(__name__)

    app.config.from_pyfile('config.py')
```

```
    db.init_app(app)

    login_manager = LoginManager()
    login_manager.login_view = 'auth.login'
    login_manager.login_message = 'Silahkan Login untuk mengakses
halaman'
    login_manager.init_app(app)

    from .models import User

    @login_manager.user_loader
    def load_user(user_id):
        return User.query.get(int(user_id))

    from .auth import auth as auth_blueprint
    app.register_blueprint(auth_blueprint)

    from .main import main as main_blueprint
    app.register_blueprint(main_blueprint)

    return app

# >>> open file "config.inc.php"

# >>> py
# >>> from app import db, create_app, models
# >>> db.create_all(app=create_app())
```

LAMPIRAN 6. Script Python auth.py

```
from flask import Blueprint, render_template, redirect, url_for,
request, flash
from werkzeug.security import generate_password_hash,
check_password_hash
from flask_login import login_user, logout_user, login_required,
current_user
from .models import User
from . import db

auth = Blueprint('auth', __name__)

@auth.route("/masuk")
def login():
```

```
    return render_template("login.html")

@auth.route('/masuk', methods=['POST'])
def login_post():
    email = request.form.get('email')
    password = request.form.get('password')

    user = User.query.filter_by(email=email).first()

    if not user or not check_password_hash(user.password, password):
        flash('Email atau Password salah')
        return redirect(url_for('auth.login'))

    login_user(user)

    return redirect(url_for('homepage.index'))

@auth.route('/daftar')
def daftar():
    return render_template('register.html')

@auth.route('/daftar', methods=['POST'])
def daftar_post():
    name = request.form.get('name')
    nomorhp = request.form.get('nomorhp')
    email = request.form.get('email')
    password = request.form.get('password')
    repassword = request.form.get('repassword')

    user = User.query.filter_by(email=email).first()

    admin = User.query.filter_by(lvl='1').first()
    if not admin:
        lvl = "1"
    else:
        lvl = "5"

    if user:
        flash('Email telah digunakan')
        return redirect(url_for('auth.daftar'))

    if password != repassword:
        flash(u'Password berbeda', 'pass-error')
        return redirect(url_for('auth.daftar'))
```

```
    new_user = User(email=email, nama=name, nomorhp = nomorhp,
password=generate_password_hash(password, method='sha256'), lvl=lvl)

    db.session.add(new_user)
    db.session.commit()

    return redirect(url_for('auth.login'))

@auth.route('/logout')
@login_required
def logout():
    logout_user()
    return redirect(url_for('homepage.home'))
```

LAMPIRAN 7. Link Github Script base-maps.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/base-maps.html

LAMPIRAN 8. Link Github Script base.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/base.html

LAMPIRAN 9. Link Github Script forms.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/forms.html

LAMPIRAN 10. Link Github Script gis-maps.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/gis-maps.html

LAMPIRAN 12. Link Github Script hasil_analisis.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/hasil_analisis.html

LAMPIRAN 13. Link Github Script hasil_analisis_persentase.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/hasil_analisis_persentase.html

LAMPIRAN 14. Link Github Script homepage.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/homepage.html

LAMPIRAN 15. Link Github Script index.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/index.html\

LAMPIRAN 16. Link Github Script input_data_shp.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/input_data_shp.html

LAMPIRAN 17. Link Github Script login.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/login.html

LAMPIRAN 18. Link Github Script main.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/main.html

LAMPIRAN 19. Link Github Script maps.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/maps.html

LAMPIRAN 20. Link Github Script register.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/register.html

LAMPIRAN 21. Link Github Script tables.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/tables.html

LAMPIRAN 22. Link Github Script usermanager.html

https://github.com/albarq21/Rancang-Bangun-Sistem-Informasi-SDGS-Desa-Dengan-WebGis-Berbasis-Flask-Python/blob/main/templates/usermanager.html

LAMPIRAN 23. Link File Full Project

Dikarenakan file keseluruhan proyek tidak dapat disimpan di Github maka diunggah ke Google Drive.

https://drive.google.com/drive/folders/18cP31KcDp1DavYh-yoOLMIuOOlCy0Tj-?usp=sharing

LAMPIRAN 24. Cara Memasang Project Yang Telah Didownload

1. Extrak file RAR
2. Membuat Virtual Environment :

   python -m venv env

   Aktifasi virtual environment:

   env\scripts\activate

   Install library:

   pip install -r requirements.txt
3. Menjalankan Program

   set FLASK_APP=app.py

   set FLASK_DEBUG=1

   flask run