

## **KLASIFIKASI TUTUPAN LAHAN DENGAN MENGGUNAKAN ALGORITMA MULTI LAYER PERCEPTRON (MLP)**

### **Deskripsi :**

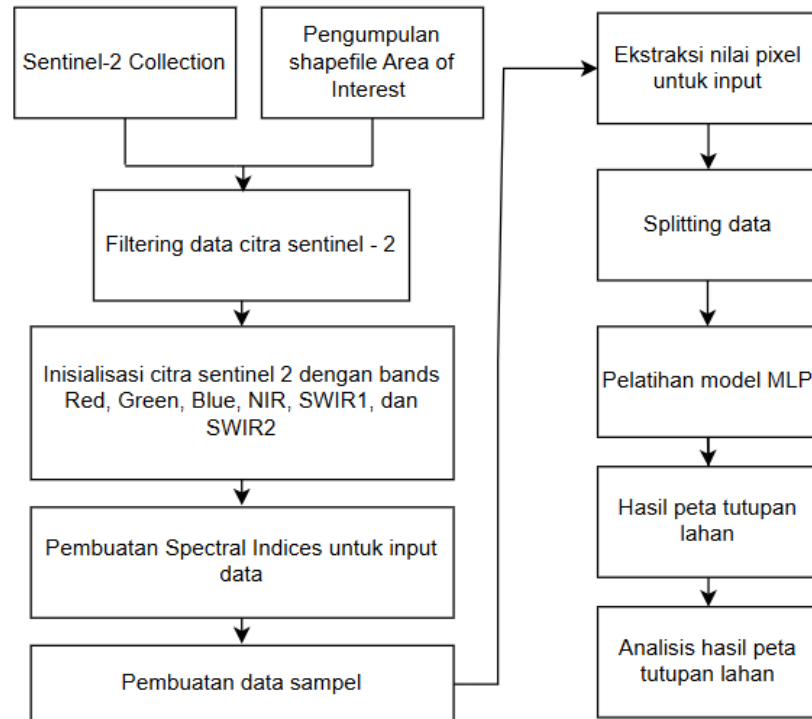
Proyek ini bertujuan untuk membangun sebuah model machine learning untuk membantu proses pengklasifikasian penutup lahan dengan memanfaatkan algoritma MLP. Model yang telah dilatih diharapkan dapat membantu dalam melakukan analisis dan pengelolaan data tutupan lahan secara efisien.

### **Data yang digunakan :**

- Sumber data

Data yang digunakan untuk klasifikasi tutupan lahan berasal dari koleksi citra satelit sentinel-2 yang telah tersedia melalui platform google earth engine dengan cakupan wilayah yang digunakan berada disekitar daerah pantai losari, kota makassar, provinsi Sulawesi Selatan. Daerah ini dipilih karena memiliki kombinasi tutupan lahan yang beragam meliputi area terbangun, perairan, mangrove, dan berbagai vegetasi lainnya serta posisinya sebagai kawasan pesisir metropolitan terbesar di Indonesia timur dengan dinamika perubahan lahan yang cukup signifikan akibat pesatnya urbanisasi di kawasan pantai sehingga dapat menantang keseimbangan antara pembangunan dan konservasi lingkungan seperti mangrove yang sangat berperan vital dalam menjaga ketahanan daerah pesisir. Hasil klasifikasi tutupan lahan ini akan bermanfaat sebagai dasar dalam upaya pemantauan perubahan penggunaan lahan pesisir, perencanaan tata ruang berkelanjutan, serta menjadi dasar pengambilan kebijakan dalam pengelolaan wilayah pesisir yang mempertimbangkan keseimbangan pembangunan dan pelestarian lingkungan.

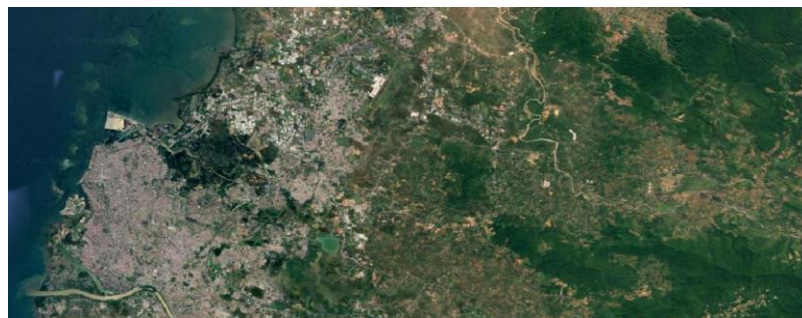
### **Tahapan Pengerjaan Proyek Akhir**



#### A. Pengumpulan Shapefile Area Of Interest (AOI) & Sentinel- 2 Collection

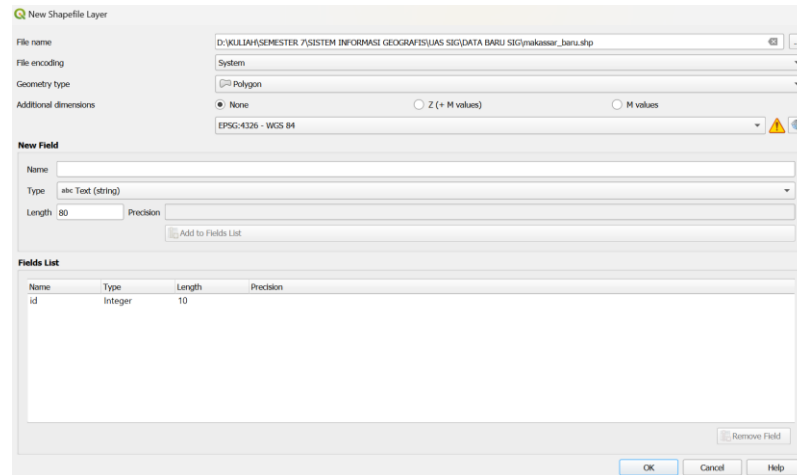
##### Pembuatan Shapefile Area Of Interest (AOI)

Shapefile Area of Interest (AOI) yang akan digunakan untuk klasifikasi tutupan lahan pada proyek ini adalah daerah disekitar pantai losari, kota makassar, Sulawesi Selatan yang akan dibuat menggunakan aplikasi quantum GIS. Pada proses pembuatan shapefile, untuk memudahkan pengerjaan akan menggunakan bantuan google satellite untuk menampilkan citra satelite wilayah akan dijadikan tujuan penelitian proyek. Caranya adalah dengan menggunakan fitur **Web -> QuickMapSevice -> Google -> Google satellite** Sehingga akan menghasilkan tampilan seperti Gambar 1.1 berikut ini.



Gambar 1. 1 Citra google satelite

- Setelah citra satelit ditampilkan, Selanjutnya akan dilakukan Pembuatan *shapefile* terhadap area yang diinginkan. Caranya adalah dengan menggunakan fitur **layer -> create layer -> New shapefile layer** kemudian akan dilakukan penamaan terhadap nama shapefile yang akan dibuat serta penentuan geometry type yang digunakan. Pada proyek ini akan menggunakan *geometry type* dengan tipe polygon.







Gambar 1. 2 Pembuatan shapefile AOI

- Setelah file shp berhasil dibuat, selanjutnya akan dilakukan pembuatan *Area of Interest* yang akan disimpan dalam file shp tersebut. Pada proses pembuatannya, akan menggunakan menu **Toggle editing -> Add Polygon Feature -> pilih bentuk polygon yang diinginkan**. Kemudian akan dilakukan penentuan batasan wilayah serta proses penyimpanan terhadap area yang akan digunakan kedepannya dalam pengerjaan proyek. Hasil shp AOI yang diperoleh dan akan digunakan adalah seperti Gambar 1.3 berikut.



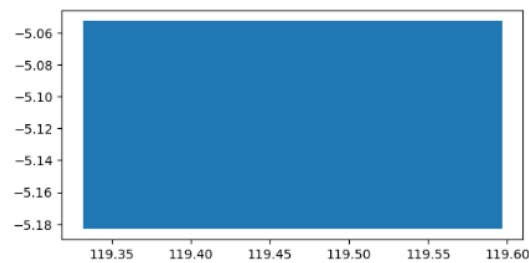
Gambar 1. 3 AOI

- Setelah pembuatan file shp berhasil dilakukan, langkah selanjutnya adalah mengupload file shp kedalam google drive untuk dilakukan pemrosesan lebih lanjut menggunakan tools google collab. file AOI yang akan diupload kedalam google drive dapat dilihat pada Gambar 1.4 berikut.

 makassar_baru.dbf	2/7/2025 3:42 PM	DBF File	1 KB
 makassar_baru.prj	2/7/2025 3:40 PM	PRJ File	1 KB
 makassar_baru.shp	2/7/2025 3:42 PM	SHP File	1 KB
 makassar_baru.shx	2/7/2025 3:42 PM	SHX File	1 KB

*Gambar 1. 4 List file AOI*

- Pada google collab, akan digunakan beberapa *library* seperti **geopandas**, **ee**, dan **geemap**. geopandas akan digunakan untuk memproses file shp berupa data vektor yang akan digunakan. Gambar 1.5 berikut merupakan contoh visualisasi sederhana dari data vektor yang digunakan



*Gambar 1. 5 Visualisasi AOI*

Gambar 1.5 diatas menampilkan visualisasi dari *Area of Interest* (AOI) yang berada pada garis bujur (longitude) berkisar antara 119.35 hingga 119.60 dan garis lintang (latitude) yang berkisar antara -5.06 hingga -5.18. sedangkan area berwarna biru menunjukkan batas wilayah AOI yang telah ditentukan sebelumnya.

- Kemudian dengan menggunakan library ee dan geemap seperti yang telah disebutkan sebelumnya, akan dilakukan pengkoneksian ke google cloud project serta menampilkan basemap menggunakan citra satellite yang berasal dari *google earth engine* yang nantinya akan digunakan dalam proses pengklasifikasian. Gambar 1.6 berikut adalah contoh dari *basemap* yang akan digunakan.



Gambar 1. 6 Basemap

Setelah *basemap* berhasil ditampilkan, terdapat beberapa langkah yang akan dilakukan meliputi hal – hal berikut ini :

- Mengubah file shp AOI yang telah dibuat menjadi bentuk *earth engine object* agar dapat digunakan didalam *google earth engine*. Proses pengubahan menggunakan kode berikut ini

```
AOI_ee = geemap.shp_to_ee("/content/drive/MyDrive/KULIAH/SEMESTER 7/SISTEM
INFORMASI GEOGRAFIS/UAS SIG/DATA_BARU_SIG/makassar_baru.shp")

AOI_ee = AOI_ee.geometry()
```

- Melakukan penambahan AOI yang telah diubah kedalam *earth engine object* kedalam *basemap*. Setelah berhasil diubah kedalam bentuk *earth engine object*, maka akan dilakukan penambahan pada basemap yang sebelumnya telah dibuat kemudian akan dilakukan pemfokusan pada wilayah AOI tersebut.

Pada Gambar 1.7 berikut ini adalah hasil dari base map yang telah berhasil dilakukan penambahan AOI serta pemfokusan pada wilayah tersebut.



Gambar 1. 7 AOI pada basemap

- Setelah wilayah telah berhasil di fokuskan pada area yang akan diteliti, pada proyek kali ini selanjutnya akan digunakan *ImageCollection* dari citra satellite sentinel-2 yaitu **COPERNICUS/S2\_HARMONIZED**.

## B. Filtering Data Citra Sentinel-2

- Setelah berhasil memperoleh Kumpulan citra dari satellite sentinel – 2, selanjutnya akan dilakukan *filtering* terhadap koleksi citra tersebut. Beberapa penyaringan yang dilakukan meliputi hanya memilih data dengan tutupan awan berada dibawah 20%, koleksi citra yang berada dalam rentang waktu '2024-01-01' sampai dengan '2025-01-31', serta pemfokusan koleksi citra hanya berada di wilayah kajian. Hasil dari proses filtering menghasilkan jumlah citra yang sesuai sebanyak 21 citra. Untuk melakukan filtering dan pengecekan, dilakukan dengan menggunakan kode berikut ini

```
S2_filtered =S2_Col.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',
20)).filter(ee.Filter.date('2024-01-01', '2025-01-31')).filter(ee.Filter.bounds(AOI_ee))

print(S2_filtered.size().getInfo())
```

- Setelah dilakukan proses *filtering*, selanjutnya jumlah koleksi citra yang diperoleh akan diubah menjadi 1 citra saja sehingga menghasilkan komposit dari beberapa citra yang diperoleh sebelumnya dengan menggunakan fungsi `median()`. hasil visualisasi dari komposit citra yang



diperoleh dengan menggunakan *True color* (432) dapat dilihat pada Gambar 1.8 berikut.



Gambar 1. 8 True color komposit citra sentinel-2

### C. Inisialisasi Citra Satelite Sentinel – 2 Dengan Band Yang Diperlukan

- Pada tahap ini akan dilakukan pemilihan band dari citra sentinel - 2 komposit sebelumnya yang dibutuhkan dalam proses pengklasifikasian tutupan lahan kedepannya yang kemudian data tersebut akan disimpan dalam format .tif didalam drive, beberapa band yang dipilih adalah band Red (B4), Green (B3), Blue (B2), NIR (B8), SWIR1 (B11), dan SWIR2 (B12). Berikut adalah kode yang digunakan pada tahapan ini

```
selected_bands = S2_medianComposite.select(['B2', 'B3', 'B4', 'B8', 'B11', 'B12'])
selected_bands

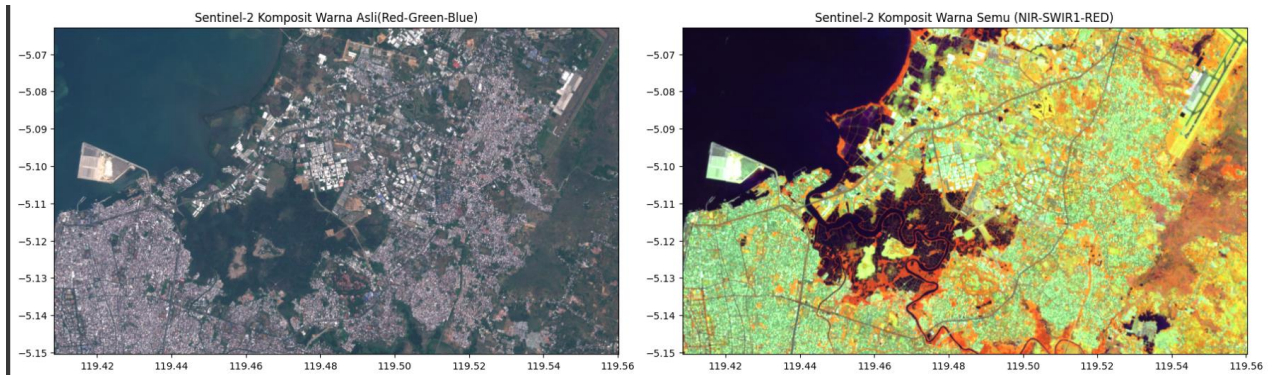
geemap.ee_export_image_to_drive(selected_bands,description="MAKASSAR_BARU",
folder="DATA_BARU_SIG", region=AOI_ee, scale=10)
```

- Setelah data berhasil disimpan, selanjutnya data raster tersebut dapat diproses dengan menggunakan library **rasterio**. Library ini akan digunakan untuk membuka file raster sebelumnya dengan nama MAKASSAR\_BARU yang telah tersimpan di drive. Berikut adalah contoh kode penggunaanya

```
raster_file_path = '/content/drive/MyDrive/KULIAH/SEMESTER 7/SISTEM INFORMASI
GEOGRAFIS/UAS SIG/DATA_BARU_SIG /MAKASSAR_BARU.tif'
dataset = rasterio.open(raster_file_path)
```

- Kemudian untuk memvisualisasikan data raster, kita perlu memastikan data berbentuk numpy array terlebih dahulu dengan menggunakan fungsi `.read()`. Setelah berhasil dipastikan, lalu akan dibuat visualisasi data dengan 2 buah visualisasi yaitu visualisasi dengan menggunakan (B4, B3, B2) dan

komposit warna semu (B8, B11, B4). Gambar 1.9 berikut merupakan hasil dari visualisasi yang dilakukan



Gambar 1. 9 Visualisasi area kajian dengan true color dan komposit warna semu

Dari hasil visualisasi diatas, terlihat bahwa dengan warna semu vegetasi seperti mangrove lebih mudah terdeteksi dengan warna yang lebih mencolok sehingga dapat membantu memudahkan dalam membedakannya dengan vegetasi lain ketika proses pembuatan sampel data yang akan dilakukan kedepannya.

#### D. Pembuatan Spectral Indices untuk Input Data

- Pada tahap ini, selanjutnya akan dilakukan pembuatan *spectral index* yang akan digunakan sebagai *input feature* tambahan dalam input data yang dimiliki. *Spectral index* akan menampilkan fitur tertentu yang ada pada permukaan bumi seperti vegetasi, air, lahan terbangun dan sebagainya melalui perpaduan dua atau lebih band spektral. Pada proyek ini, beberapa *Spectral index* yang akan dibuat adalah NDVI (*Normalized Difference vegetation Index*), NDBI (*Normalized Difference Built-up*), MNDWI (*Modified Normalized Difference Water index*), NDMI (*Normalized Different Mangrove Index*), dan CMRI (*Combined Mangrove Recognition Index*). *Spectral index* tersebut kemudian dapat digunakan untuk mendeteksi kesehatan vegetasi, deteksi area terbangun, badan air, maupun ekosistem mangrove. Berikut ini merupakan kode yang digunakan dalam pembuatan *spectral index* tersebut.



**# NDVI: (NIR - Red) / (NIR + Red)**

$$\text{NDVI} = (\text{nir} - \text{red}) / (\text{nir} + \text{red})$$

**# NDBI: (SWIR1 - NIR) / (SWIR1 + NIR)**

$$\text{NDBI} = (\text{swir1} - \text{nir}) / (\text{swir1} + \text{nir})$$

**# MNDWI: (Green - SWIR1) / (Green + SWIR1)**

$$\text{MNDWI} = (\text{green} - \text{swir1}) / (\text{green} + \text{swir1})$$

**# NDMI: (SWIR2 - Green) / (SWIR2 + Green)**

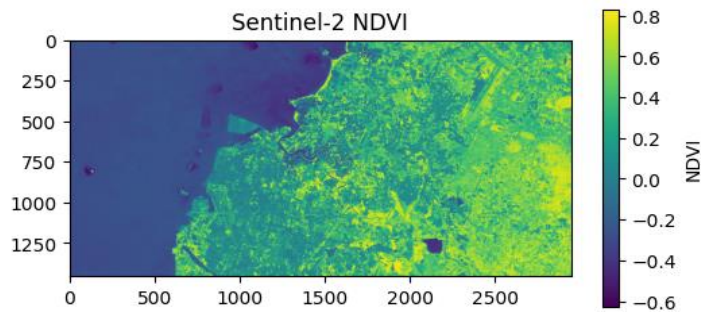
$$\text{NDMI} = (\text{swir2} - \text{green}) / (\text{swir2} + \text{green})$$

**# CMRI: NDVI - NDWI**

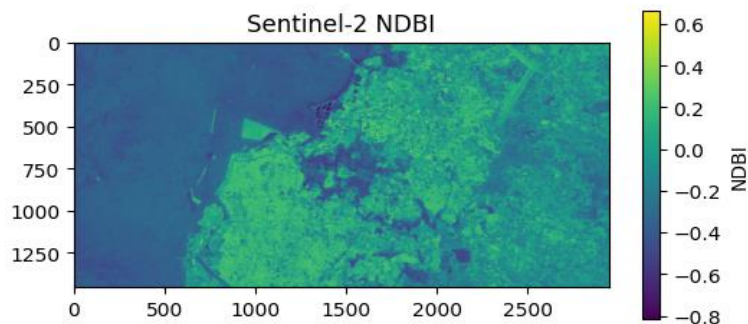
$$\text{NDWI} = (\text{green} - \text{nir}) / (\text{green} + \text{nir})$$

$$\text{CMRI} = \text{NDVI} - \text{NDWI}$$

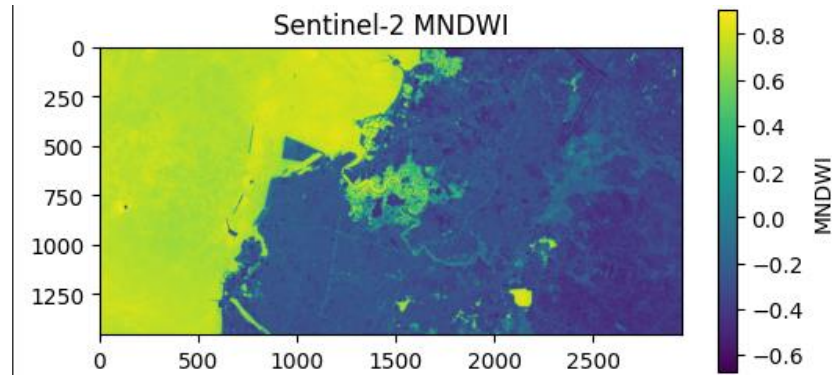
Berikut ini merupakan contoh visualisasi dari hasil penerapan *spectral index* yang telah dilakukan.



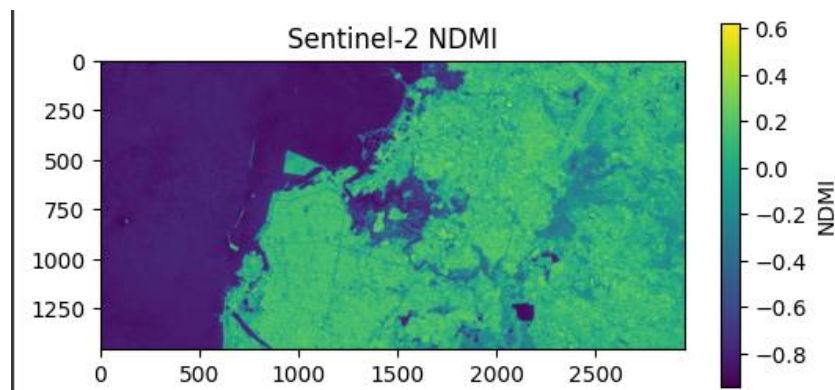
Gambar 1. 10 NDVI pada wilayah kajian



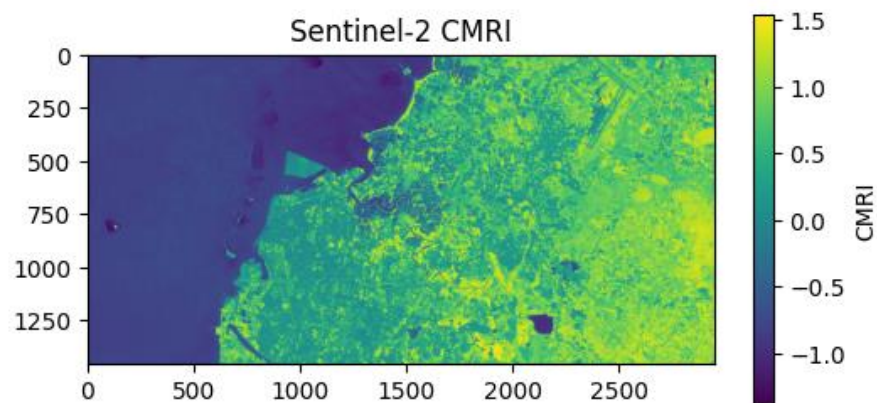
Gambar 1. 11 NDBI pada wilayah kajian



*Gambar 1. 12 MNDWI pada wilayah kajian*



*Gambar 1. 13 NDMI pada wilayah kajian*



*Gambar 1. 14 CMRI pada wilayah kajian*

- Pada visualisasi NDVI, nilai yang semakin mendekati 1 menunjukkan bahwa wilayah tersebut memiliki vegetasi hijau yang sehat begitu juga sebaliknya. Pada NDBI, nilai tinggi (mendekati 1) menandakan bahwa wilayah tersebut merupakan wilayah pemukiman atau area terbangun, sedangkan pada visualisasi MNDWI, nilai yang semakin mendekati 1 menunjukkan bahwa wilayah

tersebut memiliki kandungan air yang tinggi. Pada NDMI, semakin rendah nilainya menunjukkan bahwa kawasan tersebut memiliki kandungan air yang tinggi. Sehingga dapat dilihat pada visualisasi NDMI bahwa kawasan mangrove dideteksi memiliki warna citra yang mirip dengan badan air. Kemudian untuk visualisasi CMRI mengindikasikan bahwa semakin besar nilai yang diperoleh (mendekati 1), maka akan menunjukkan bahwa kemungkinan wilayah tersebut merupakan kawasan mangrove.

Setelah dilakukan pembuatan *spectral index* selanjutnya bands tersebut akan dimasukkan kedalam citra sentinel- 2 yang dimiliki sebelumnya. Proses pendefinisian nama bands kedalam citra satelit akan dilakukan dengan menggunakan kode berikut.

```
# Mendefinisikan deskripsi nama bands (original bands + spectral indices)
band_descriptions = ['B2', 'B3', 'B4', 'B8', 'B11', 'B12',
                    'NDVI', 'NDBI', 'MDNWI', 'NDMI', 'CMRI']

# Tempat untuk menyimpan file tif.
output_file_path_S2New = '/content/drive/MyDrive/KULIAH/SEMESTER 7/SISTEM
INFORMASI GEOGRAFIS/UAS
SIG/DATA_BARU_SIG/Sentinel2_Spectral_Indices_MAKASSAR_BARU.tif'

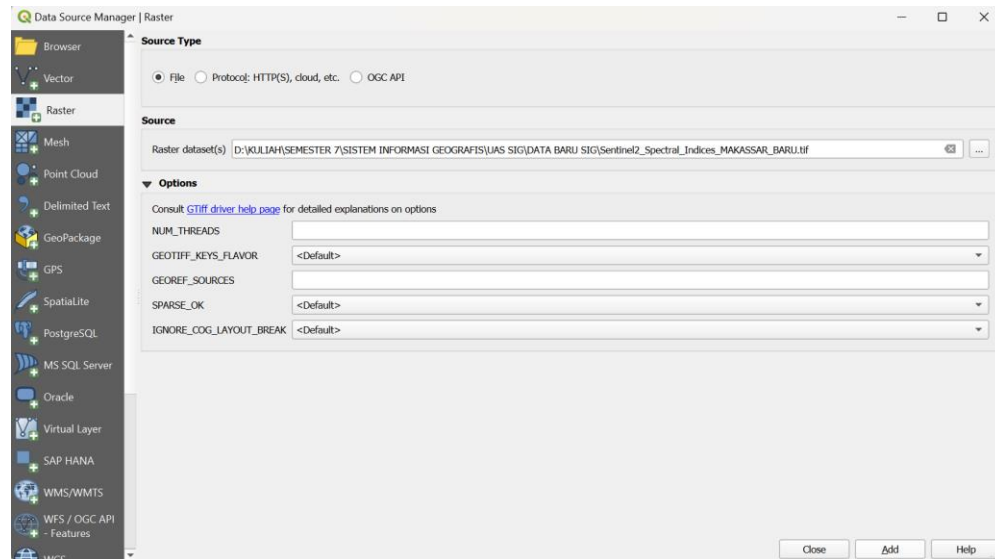
# Export Sentinel-2 dengan sepctral indices
with rasterio.open(output_file_path_S2New, 'w', **meta) as dst:
    for i in range(11):
        dst.write(S2_image_with_indices[:, :, i], i + 1) # Write each band

# Menambahkan deskripsi band names untuk setiap bands
dst.set_band_description(i + 1, band_descriptions[i])
```

Setelah kode tersebut berhasil diterapkan, maka deskripsi dari bands pada citra satellite yang digunakan yang sebelumnya berjumlah 6 bands akan berubah menjadi memiliki 11 bands yang terdiri dari 'B2', 'B3', 'B4', 'B8', 'B11', 'B12', 'NDVI', 'NDBI', 'MDNWI', 'NDMI', dan 'CMRI'.

## E. Pembuatan Data Sampel

- Selanjutnya akan dilakukan pembuatan data sampel menggunakan aplikasi QGIS. Sampel yang akan dibuat berupa sampel titik dari masing - masing kelas yang ada sebanyak **100 sampel** untuk setiap kelasnya. pada proses pembuatan sampel, akan digunakan data sentinel - 2 yang telah ditambahkan *spectral indeks* sebelumnya. cara menambahkan data raster pada QGIS adalah dengan klik menu **layer -> Add Raster Layer -> Pilih data -> kemudian klik Add**. Gambar 1.13 berikut adalah contoh tampilan dari proses penambahan data raster tersebut.

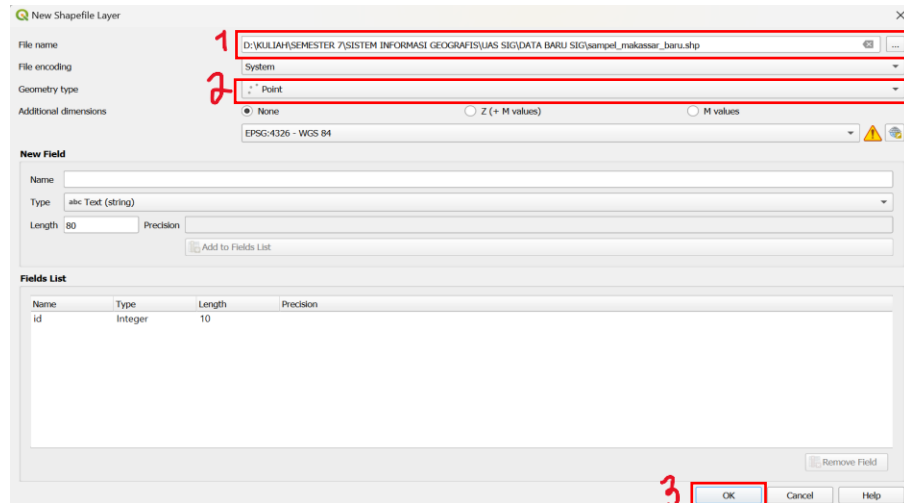


Gambar 1. 15 Upload data raster pada QGIS



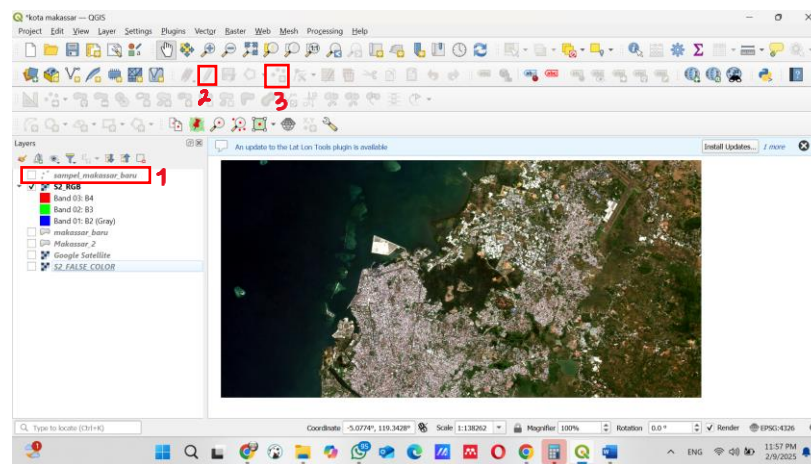
Gambar 1. 16 Hasil upload data raster pada QGIS

- Setelah layer data berhasil ditambahkan, selanjutnya akan dibuat sampel berupa titik. Setiap titik yang dibuat akan mewakili kelas yang ada. kelas yang akan dibuat adalah **kelas 0 : Tubuh Air, kelas 1 : Mangrove, kelas 2 : Vegetasi Non - Mangrove, kelas 3 : Lahan Terbuka, kelas 4 : Lahan Terbangun**. Cara pembuatan file untuk menampung sampel titik yang akan dibuat adalah dengan mengklik **tab Layer -> Create layer -> New Shapefile Layer -> pilih geometry type point -> OK**.



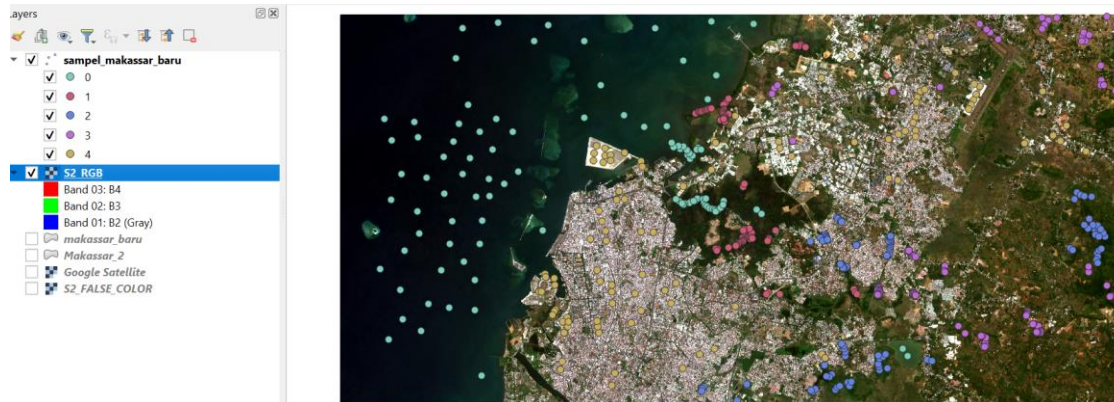
Gambar 1. 17 Proses pembuatan shapefile data sampel

- Setelah *shapefile* dengan *geometry type point* berhasil dibuat, selanjutnya akan dimulai proses pembuatan titik sampel dari masing - masing kelas yang ada. Cara pembuatan titiknya adalah dengan **klik layer shapefile point -> Toggle Editing -> Add Point Feature** seperti gambar 1.16 berikut.



Gambar 1. 18 Pembuatan titik sampel menggunakan QGIS

Hasil dari sampel titik yang telah dibuat melalui proses tersebut adalah seperti Gambar 1.17 berikut.



Gambar 1. 19 Titik sampel setiap kelas penutup lahan

## F. Ekstraksi Nilai Pixel Untuk Input

- Setelah mendapatkan 100 sampel untuk setiap kelas sehingga total data 500 sampel, selanjutnya akan dilakukan pemrosesan data *shapefile* sampel kelas tersebut dengan data input raster yang dimiliki. Berikut adalah kode yang digunakan untuk melakukan pemrosesan data *shapefile* (sampel) dengan data input raster yang dimiliki.

```
def extract_pixels_from_shapefile(shapefile, raster):
    training_samples = []
    for index, row in shapefile.iterrows():
        geometry = [row['geometry']]
        id_class = row['id']

        out_image, out_transform = mask(raster, geometry, crop=True)

        out_image = out_image.reshape(raster.count, -1).T # Sekarang (pixels, bands)

        valid_pixels = out_image[~np.isnan(out_image).any(axis=1)]

        for pixel in valid_pixels:
            training_samples.append((pixel, id_class))

    features = np.array([sample[0] for sample in training_samples])
    labels = np.array([sample[1] for sample in training_samples])
    return features, labels

features, labels = extract_pixels_from_shapefile(sampil_point, input_raster)
print(features.shape)
print(labels.shape)
```

- Melalui kode diatas, akan dilakukan pengekstrakan data sehingga setiap sampel yang ada akan memperoleh nilai pixel dari setiap band. Setelah dilakukan pengekstrakan data, selanjutnya akan



dilakukan pengecekan terhadap nilai kosong yang ada pada data sekaligus menghitung total dari masing – masing sampel setelah dilakukan pembersihan. Berikut adalah kode yang digunakan.

```
# Membuat mask untuk cek Nan values dari feature dan label
nan_mask = np.isnan(features).any(axis=1) | np.isnan(labels)

# Apabila ada Nan value maka akan dihapuskan
features = features[~nan_mask]
labels = labels[~nan_mask]

# Menghitung final total sampel masing-masing kelas
sample_new = pd.DataFrame({'class': labels})
print(sample_new['class'].value_counts())
```

Melalui penggunaan kode diatas, pada penelitian ini diperoleh sampel dari masing masing kelas tetap berjumlah 100 sampel. Hal ini menandakan bahwa didalam data tidak ditemukan adanya nilai yang kosong.

## G. Splitting Data

- Setelah data sampel berhasil untuk memperoleh nilai pixel dari setiap band sekaligus bersih terhadap nilai kosong, selanjutnya akan dilakukan proses *splitting data* pada data yang dimiliki. Proses *splitting data* akan menggunakan *library train\_test\_split* dari sklearn. Berikut adalah kode yang digunakan untuk melakukan *split data* terhadap dataset yang dimiliki.

```
from sklearn.model_selection import train_test_split

# Split data: 80% for training and 20% for testing
x_train, x_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, stratify=labels,
random_state=42)
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

- Kemudian pada proyek ini, data akan dibagi menjadi 80% untuk *training data* dan 20% untuk *testing data*. Hasil dari *splitting data* yang dilakukan menghasilkan 400 sampel data akan digunakan untuk *training model* dan 100 sampel data akan digunakan untuk *testing model*. Data yang telah berhasil di *split* ini kemudian dapat disimpan maupun dimuat ulang dengan menggunakan fungsi *np.save* atau *np.load* yang ada pada *library numpy* .

## H. Pelatihan Model Multi Layer Perceptron (MLP)

- Pada tahapan selanjutnya, akan dilakukan pelatihan model *Multi Layer Perceptron* (MLP) menggunakan data yang telah dilakukan proses *splitting* sebelumnya. Untuk memperoleh model MLP, diperlukan terlebih dahulu proses *import* model menggunakan library *sklearn.neural\_network*.

```
# Import MLP classifier dari Library Scikit-learn
from sklearn.neural_network import MLPClassifier

# model MLP dengan default parameter
model_mlp = MLPClassifier(random_state=42)
```

- Setelah model berhasil di import, selanjutnya akan dilakukan proses *training* dan *testing* model. selain itu, juga akan dilakukan evaluasi terhadap kinerja model dengan menggunakan beberapa metrik evaluasi seperti *confusion metrix* dan *classification report*. Berikut ini merupakan kode yang digunakan dalam proses pelatihan dan pengujian serta evaluasi dari kinerja model yang telah dilatih.

```
#Training MLP Model
model_mlp.fit(x_train, y_train)

# Evaluation assessment
# Menggunakan scikit-learn library untuk menghitung overall accuracy, classification report,
# dan confusion metrix
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Menerapkan model_mlp untuk memprediksi kelas dari tetsing dataset
y_pred_default = model_mlp.predict(x_test)

# Membuat evaluation function untuk mempermudah kedepannya
def evaluate_classification(y_true, y_pred):
    # Perhitungan overall accuracy
    OA = accuracy_score(y_true, y_pred)

    # Print Overall Accuracy
    print(f"Overall Accuracy: {OA:.4f}")

    # Menghitung classification report: precision, recall, f1-score
    report = classification_report(y_true, y_pred, digits=4)
    print(report)
```

```
# Menghitung confusion matrix
class_names = ['Tubuh Air', 'Mangrove', 'Vegetasi Non-Mangrove', 'Lahan Terbuka', 'Lahan Terbangun']
cm = confusion_matrix(y_true, y_pred)

# Visualisasi confusion matrix menggunakan seaborn heatmap
plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix', fontsize=14)
plt.show()

# Menerapkan function evaluation assessment dengan data yang sebenarnya atau y_test dan
data hasil prediksi
evaluate_classification(y_test, y_pred_default)
```

Melalui proses pelatihan dan pengujian diatas, model MLP berhasil mendapatkan performa yang sangat baik dalam mengklasifikasikan data tutupan lahan dengan nilai kesalahan yang sangat kecil. Visualisasi dari hasil evaluasi kinerja model MLP dalam mengklasifikasikan penutup lahan dapat dilihat pada Gambar 1.18 berikut ini.

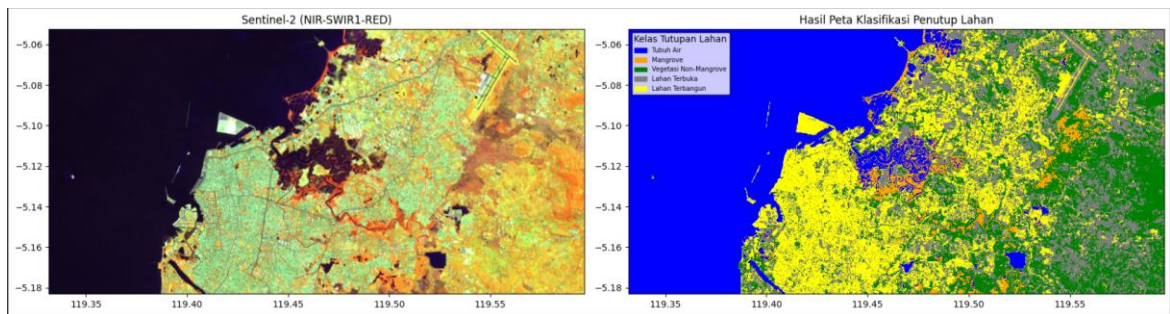


Gambar 1. 20 Hasil evaluasi model MLP

Dari hasil kinerja model MLP pada Gambar 1.18 diatas, model MLP mampu mengklasifikasikan dengan benar dari hampir keseluruhan data uji yang diberikan dengan perolehan akurasi sebesar **96%**. Model hanya mengalami beberapa kesalahan kecil dalam mengklasifikasikan data penutup lahan seperti pada data lahan terbangun, lahan terbuka, dan vegetasi non – mangrove pada data uji. Hal ini membuktikan bahwa performa model MLP sangat baik digunakan dalam mengklasifikasikan data penutup lahan pada wilayah yang di teliti.

#### i. Hasil Peta Penutup Lahan

- Setelah model berhasil dilatih dan mendapatkan performa yang baik dalam proses pengujian, selanjutnya model akan digunakan untuk mengklasifikasikan keseluruhan citra yang ada. Model akan diberikan data raster yang telah di *reshape* menjadi nilai (*pixel, bands*) yang kemudian akan digunakan oleh model MLP dalam mengklasifikasikan kelas dari setiap piksel tersebut. Hasil klasifikasi keseluruhan peta penutup lahan yang dihasilkan oleh model MLP adalah sebagai berikut.



Dari visualisasi diatas, terlihat model MLP mampu mengklasifikasikan setiap penutup lahan yang ada pada peta yang diberikan. Setiap penutup lahan diwakili oleh warna yang berbeda beda. Warna biru digunakan untuk mengklasifikasikan wilayah badan air, orange untuk kelas Mangrove, hijau untuk Vegetasi Non Mangrove, dan kuning untuk kelas Lahan terbangun.

#### j. Analisis Hasil Peta Penutup Lahan

- Pada tahapan selanjutnya akan dilakukan perhitungan terhadap luasan dari masing – masing penutup lahan yang telah berhasil di klasifikasikan oleh model. Proses perhitungan luasan penutup lahan dapat ditemukan dengan menggunakan kode berikut ini.

```
# Mendefinisikan kelas penutup lahan

labels = ['Tubuh Air', 'Mangrove', 'Vegetasi Non-Mangrove', 'Lahan Terbuka', 'Lahan Terbangun']
classes = [0, 1, 2, 3, 4]

# Sentinel-2 pixel size in hectares (each pixel covers 10m x 10m or 100 m² = 0.01 hectares)
pixel_area_ha = 0.01
```

```

# Membuat function untuk menghitung luasan per kelas dan total area
def calculate_class_areas(classified_map, classes, pixel_area_ha):
    areas = {}
    total_area = 0
    for cls in classes:
        # Menghitung total piksel untuk setiap kelas
        pixel_count = np.sum(classified_map == cls)
        # Convert menjadi area Ha
        area = pixel_count * pixel_area_ha
        areas[cls] = area
        total_area += area # Menghitung total luasan
    return areas, total_area

# Menerapkan fungsi yang telah dibuat
area_by_class, total_area = calculate_class_areas(classified_map, classes, pixel_area_ha)

# Print hasil luasan setiap kelas
for cls, area in area_by_class.items():
    print(f"{labels[cls]}: {area:.2f} Ha")

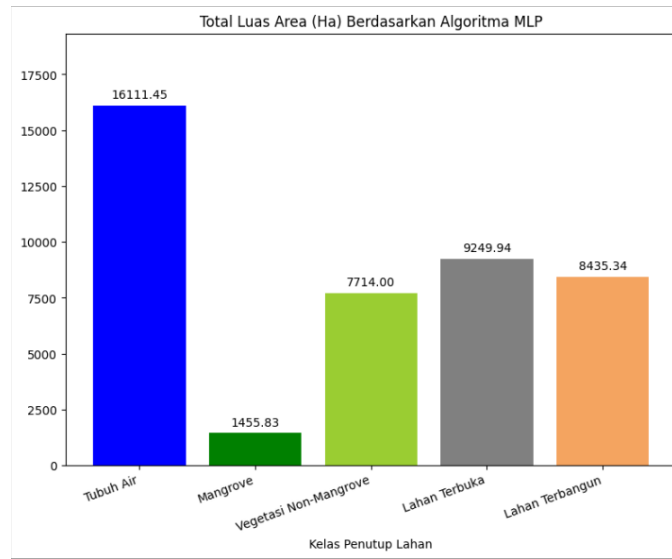
# Print the total area
print(f"Total Area: {total_area:.2f} Ha")

```

Dari hasil perhitungan yang dilakukan, diperoleh hasil luasan sebagai berikut.

Kelas	Luasan
Tubuh Air	16111.45 Ha
Mangrove	1455.83 Ha
Vegetasi Non - Mangrove	7714.00 Ha
Lahan Terbuka	9249.94 Ha
Lahan Terbangun	8435.34 Ha
Total Area	42966.56 Ha

Kemudian agar data tersebut lebih mudah untuk dipahami, akan dilakukan visualisasi data dengan menggunakan library matplotlib pada python. Hasil visualisasi yang dihasilkan adalah sebagai berikut.



*Gambar 1. 21 Visualiasi luas penutup lahan*

Berdasarkan Hasil perolehan diatas, klasifikasi penutup lahan dengan menggunakan model MLP terbukti mampu memperoleh akurasi yang tinggi sebesar 96%. Hasil klasifikasi menunjukkan bahwa luas area mangrove hanya 1.455,83 Ha yang jauh lebih kecil dibandingkan kelas lainnya. Hal ini mengindikasikan bahwa ekosistem mangrove di wilayah Pantai Losari, Makassar, berada dalam kondisi yang terbatas atau mengalami degradasi. Mangrove memiliki peran penting dalam menjaga kestabilan pesisir, mencegah abrasi, serta menjadi habitat bagi berbagai spesies laut. Luas yang kecil ini dapat menjadi perhatian bagi pengambil kebijakan untuk melakukan upaya konservasi dan rehabilitasi.

Di sisi lain, hasil klasifikasi juga menunjukkan bahwa lahan terbuka memiliki luas 9.249,94 Ha, sedangkan lahan terbangun mencapai 8.435,34 Ha. Luasnya area lahan terbuka dapat mencerminkan adanya lahan kosong yang belum dimanfaatkan. Sementara itu, luasnya lahan terbangun menunjukkan tingginya tingkat urbanisasi di wilayah pesisir ini. Konversi lahan terbuka atau bahkan lahan mangrove menjadi lahan terbangun dapat berdampak pada lingkungan pesisir, seperti penurunan keanekaragaman hayati.

Dengan luas mangrove yang kecil dan luas lahan terbangun yang signifikan, diperlukan kebijakan yang seimbang antara pembangunan dan konservasi lingkungan. Jika tren urbanisasi terus berlanjut tanpa pengelolaan yang baik, maka keberadaan mangrove dapat semakin terancam. Rehabilitasi mangrove dan perencanaan tata ruang yang berkelanjutan sangat diperlukan agar pembangunan pesisir tetap dapat dilakukan tanpa mengorbankan keseimbangan ekologi wilayah pantai losari.

Dengan adanya peta dan model klasifikasi yang telah dibangun, diharapkan dapat membantu dalam upaya pengambilan kebijakan dengan memperoleh gambaran yang jelas mengenai distribusi berbagai jenis lahan. Informasi ini dapat digunakan untuk merancang program rehabilitasi ekosistem serta mengembangkan



kebijakan tata ruang yang lebih berkelanjutan. Selain itu, klasifikasi penutup lahan juga dapat membantu dalam mengidentifikasi wilayah prioritas yang membutuhkan intervensi segera, seperti area mangrove yang mengalami degradasi atau lahan terbuka yang berpotensi dikembangkan secara ramah lingkungan. Dengan demikian, hasil klasifikasi ini tidak hanya menjadi dasar dalam pengambilan keputusan, tetapi juga sebagai alat untuk memantau efektivitas kebijakan yang telah diterapkan dalam menjaga keseimbangan antara pembangunan dan konservasi lingkungan di pantai losari.