

## KELOMPOK 1

# UAS DSP & ML

# ANALISI FASIES SEISMIK MENGGUNAKAN MACHINE LEARNING

DOSEN PENGAMPU MATAKULIAH : SUKARNO, M.SI.

seismic data

```
# read("C:/Users/ACER/Downloads/all_seismi  
= st[0].data
```

```
from scipy.ndimage import gaussian_filter1d
```

```
from scipy.signal import convolve
```

```
# Convolve with a window to further smooth the data  
curvature = convolve(curvature, np.ones((1, 11)) / 11)
```

```
return curvature
```

```
# Apply curvature to the seismic trace  
curvature_trace = calculate_curvature(trace)
```

# ANGGOTA KELOMPOK



- Azka Hilman Naufal (2106701406)
- Farrastha Hady (2106701116)
- Iyan Ismail (2106639724)
- Muhammad Azis Rizaldi (2106725690)





# OUTLINE

**01**

Teori Dasar

**02**

Processing

**03**

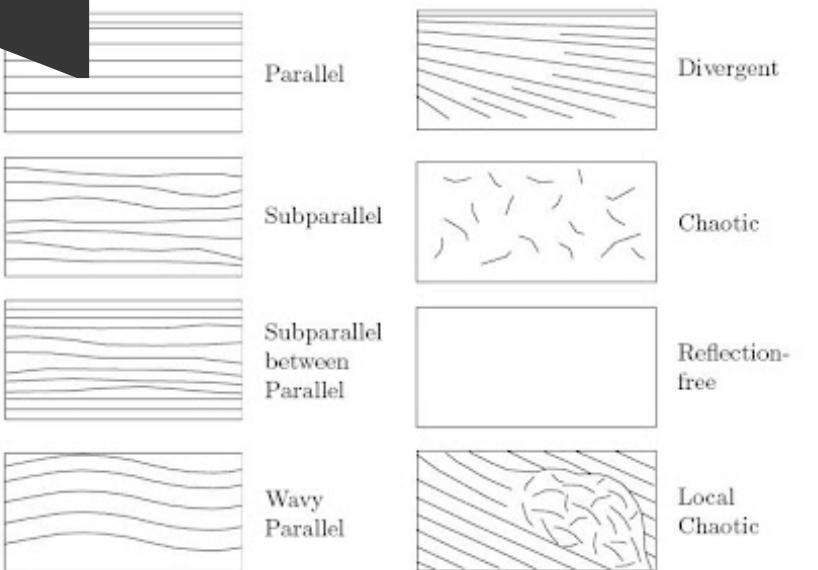
Hasil

**04**

Kesimpulan

# TEORI DASAR

## Fasies Seismik



Fasies seismik adalah unit seismik yang terdiri dari kumpulan refleksi seismik dengan parameter berbeda dari unit fasies yang berdekatan. Analisis fasies seismik dilakukan untuk mengetahui interpretasi geologi dari parameter refleksi seismik.

## Atribut Seismik

Atribut seismik merupakan penyajian dan analisa data seismik berdasarkan informasi utama, yaitu informasi waktu, frekuensi, amplitudo dan fase pada jejak seismik kompleks. Atribut seismik memberikan informasi parameter-parameter fisis batuan bawah permukaan. seperti amplitudo dan fase yang secara tidak langsung diperoleh melalui data seismik

# TEORI DASAR

## Atribut Seismik yang digunakan :

- **cosine of phase** : mengacu pada nilai kosinus dari sudut fasa antara dua sinyal atau vektor yang berhubungan dengan sinyal sinusoidal atau vektor dalam ruang berdimensi
- **instantaneous phase** : mengacu pada sudut fasa yang diukur pada titik waktu tertentu dalam suatu sinyal. Ini digunakan untuk menggambarkan bagaimana fasa (atau posisi relatif dalam siklus) dari sinyal berubah sepanjang waktu

## Support Vector Machine (SVM)

Algoritma Support Vector Machine(SVM) sendiri ialah algoritma yang bertujuan untuk menemukan hyperplane maksimal, hyperplane adalah suatu fungsi yang dapat memisahkan antara dua kelas. Pada prosesnya SVM akan memaksimalkan margin atau jarak antara pola pelatihan dan batas keputusan pembacaan atribut pada data sampling

```
## Mendefinisikan fungsi seismic attribute

def att_instan_phase(data):
    fourier_transform = np.fft.fft(data)
    magnitude_spectrum = np.abs(fourier_transform)
    phase_spectrum = np.angle(fourier_transform)
    instantaneous_phase = np.unwrap(phase_spectrum)
    return instantaneous_phase

def att_cosine_phase(data):
    spectrum = np.fft.fft(data)
    phase = np.angle(spectrum)
    cosine_phase = np.cos(phase)
    return cosine_phase

def att_max_curvature(data, window_size):
    # Apply Gaussian smoothing
    smoothed_data = gaussian_filter1d(data, sigma=1)

    # Calculate first and second derivatives
    gradient = np.gradient(smoothed_data)
    max_curvature = np.gradient(gradient)

    # Convolve with a window to further smooth the curvature
    max_curvature = convolve(max_curvature, np.ones(window_size)/window_size, mode='full')
    return max_curvature

def att_local_flatness(data, window_size):
    # Apply Gaussian smoothing
    smoothed_data = gaussian_filter1d(data, sigma=1)

    # Calculate first and second derivatives
    gradient = np.gradient(smoothed_data)
    curvature = np.gradient(gradient)

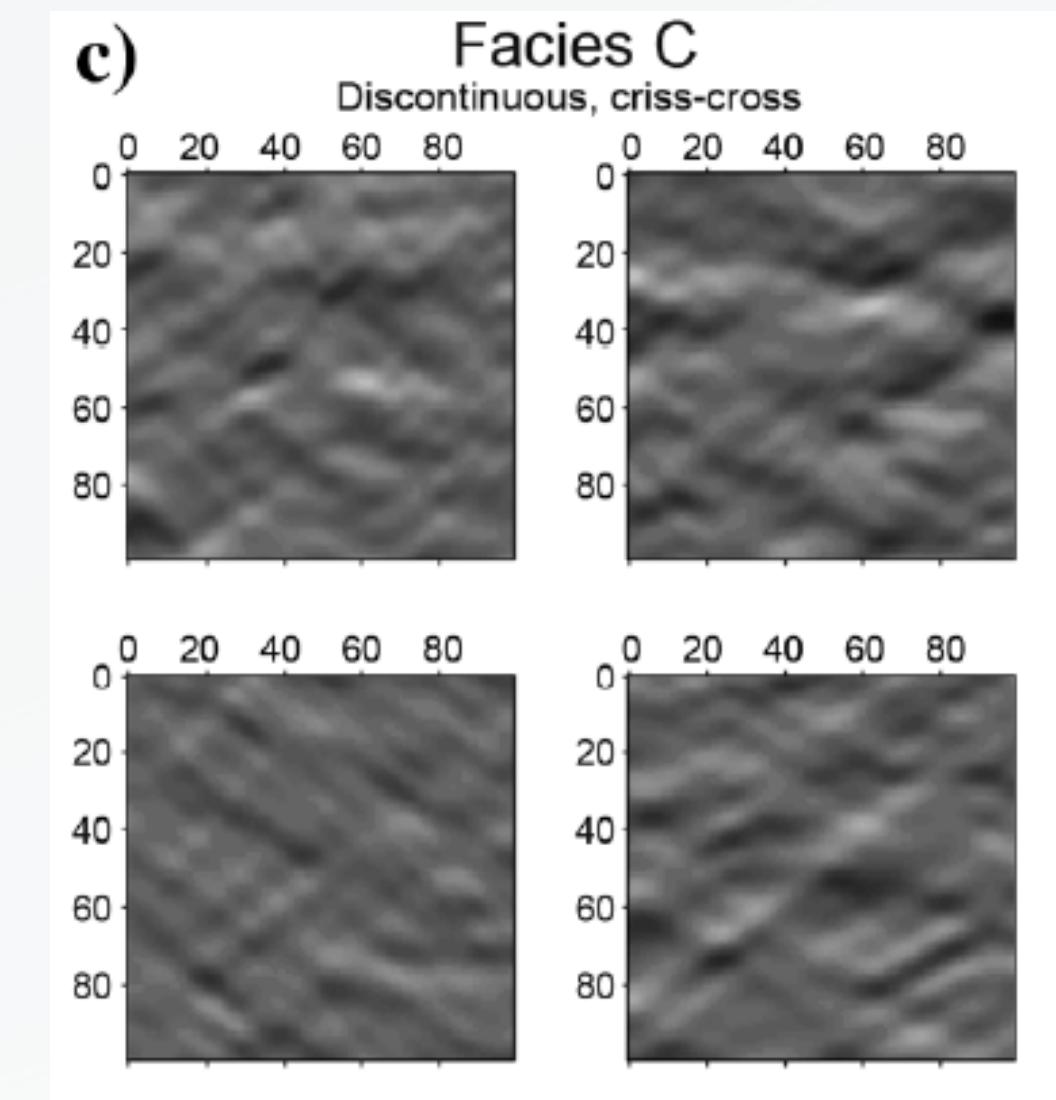
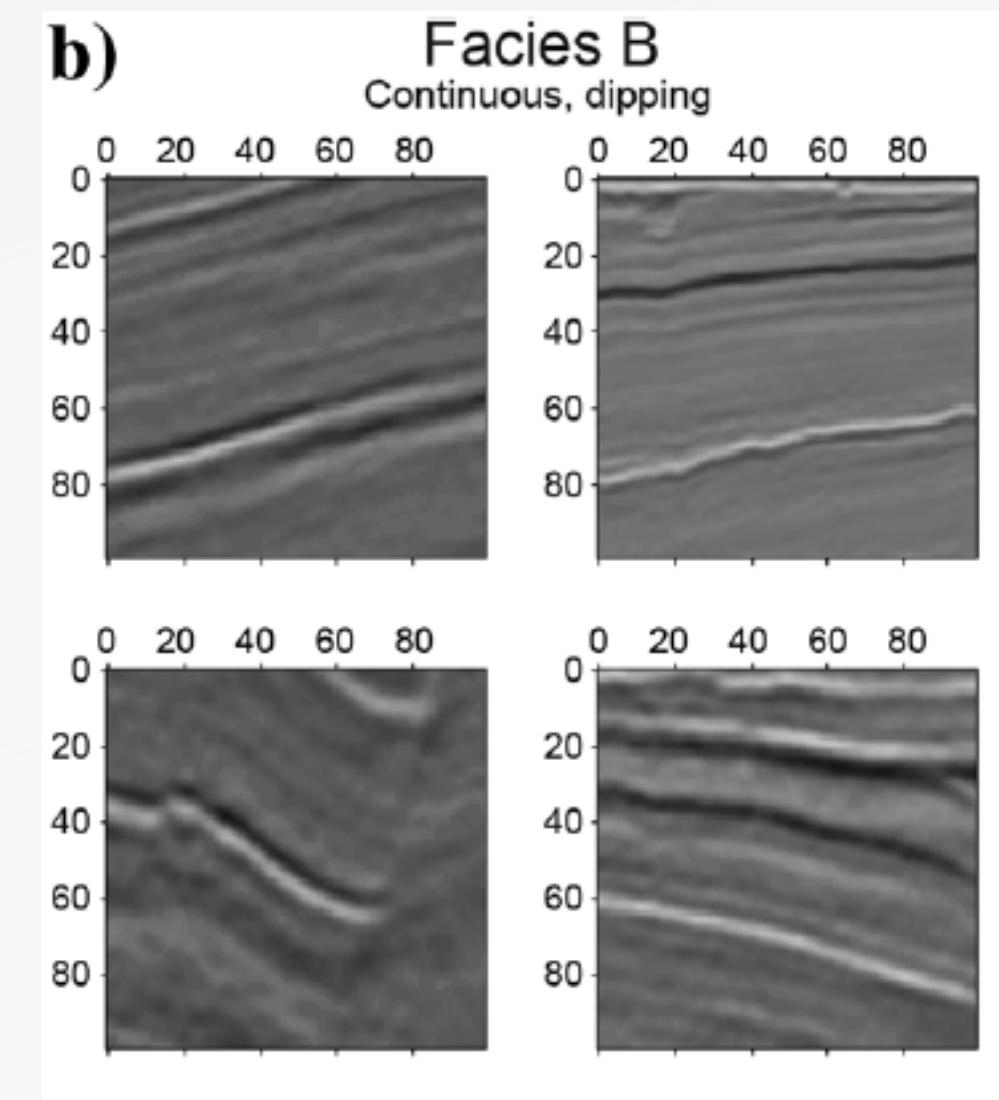
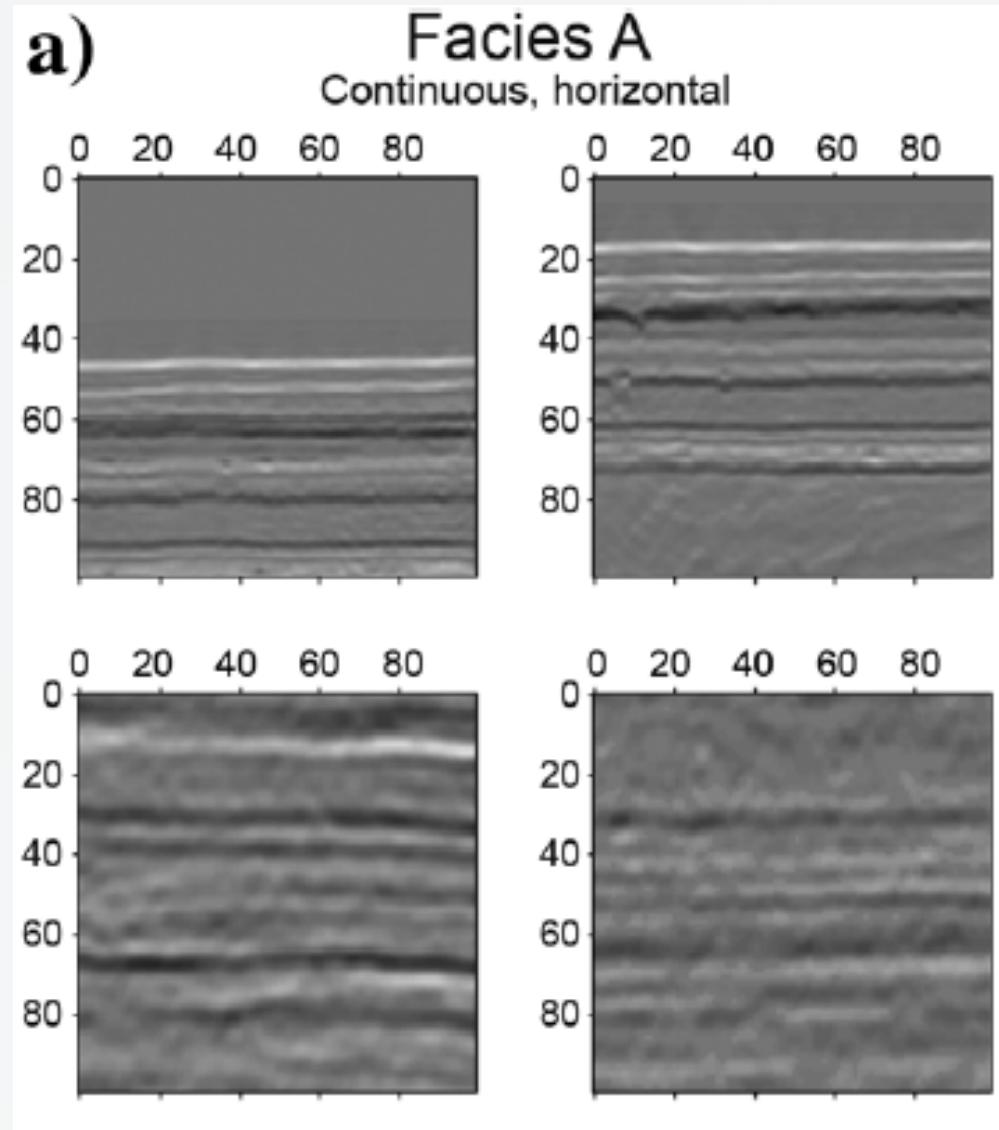
    # Convolve with a window to further smooth the curvature
    curvature = convolve(curvature, np.ones(window_size)/window_size, mode='full')

    # Calculate local flatness
    flatness = np.sqrt(np.sum(curvature**2, axis=0) / len(curvature))
    return flatness
```

## FFT (Fast Fourier Transform)

Transformasi Fourier digunakan untuk mengonversi sinyal atau data dari domain waktu ke domain frekuensi

# SAMPLE FACIES



# PROCESSING

## • Feature Extraction :

Mengekstraksi fitur/properti yang menjadi objek processing. fitur-fiturnya adalah properti statistik dari atribut seismik. Ekstraksi atribut seismik biasanya melibatkan transformasi nonlinier (misalnya transformasi Hilbert) dari data seismik asli :

$$\mathbf{X}_i = T_i(\mathbf{X}_0) \quad \text{dimana } \mathbf{X}_0 \text{ adalah data asli, } T_i \text{ adalah transformasi, dan } \mathbf{X}_i \text{ adalah atribut seismik yang dihasilkan dinormalisasi}$$

Properti statistik dihitung disemua sampel fasies, sesudah mendefinisika atribut :

$$f_{ij} = f(\mathbf{X}_{ij}) = (p_{20}(\mathbf{X}_{ij}), p_{80}(\mathbf{X}_{ij}), \dots)$$

Statistik yang kami gunakan meliputi (1) persentil ke-20, (2) persentil ke-80, (3) mean, (4) simpangan baku, (5) kesalahan baku mean, (6) skewness, dan (7) kurtosis.

```
import numpy as np
from scipy.stats import scoreatpercentile

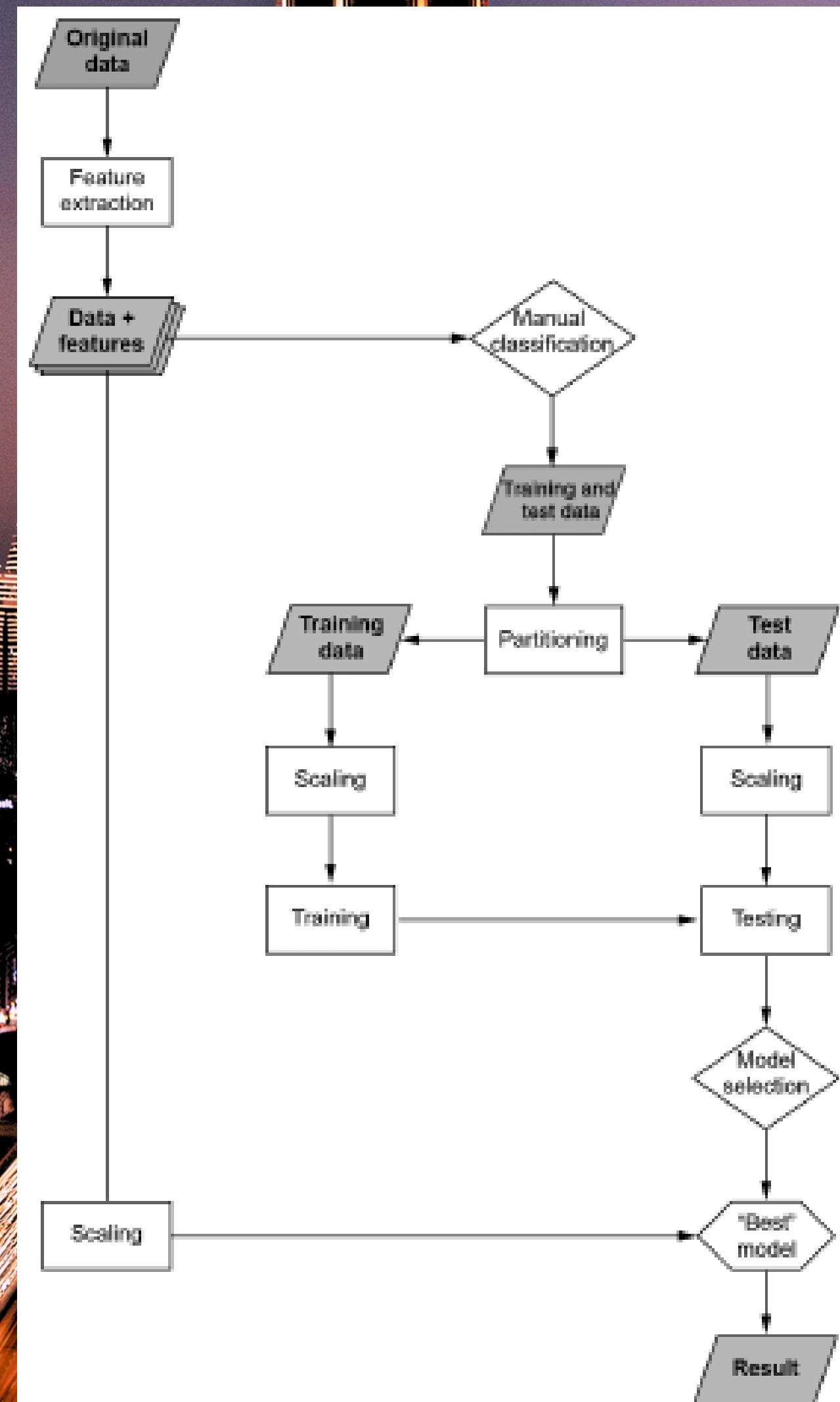
# Inisialisasi list untuk menyimpan hasil statistik
statistics_array1 = []

# Loop untuk menghitung statistik untuk masing-masing array
for phase_array in instantaneous_phase:
    # Menghitung percentiles
    ph_percentile_20 = scoreatpercentile(phase_array, 20)
    ph_percentile_80 = scoreatpercentile(phase_array, 80)

    # Menghitung mean, standard deviation, dan standard error of the mean
    ph_mean_value = np.mean(phase_array)
    ph_std_dev_value = np.std(phase_array)
    ph_std_error_value = ph_std_dev_value / np.sqrt(len(phase_array))

    # Menyimpan hasil ke dalam list
    statistics_array1.append([ph_percentile_20, ph_percentile_80, ph_mean_value, ph_std_dev_value, ph_std_error_value])

# Mengubah list menjadi array numpy
statistics_array1 = np.array(statistics_array1)
```



# PROCESSING

- **Regularization :**

Menggunakan sejumlah besar fitur dapat mengakibatkan overfitting model yang terlalu kompleks menggambarkan kesalahan acak atau gangguan dalam data. Untuk menghindari overfitting, kami mengatur model kami, jika memungkinkan, selama pelatihan.

Meskipun kami tidak melakukan pemilihan fitur secara eksplisit, regularisasi dapat dianggap sebagai metode implisit untuk membatasi fitur.

- **Training :**

Memilih fitur-fitur yang akan digunakan sebagai input model dan label yang akan diprediksi. Fitur adalah variabel input yang digunakan untuk membuat prediksi, sedangkan label adalah variabel target yang ingin diprediksi. Proses ini melibatkan pemahaman yang baik tentang data dan hubungan antara fitur dan label.

```
X = np.asarray(df[['20_Percentile','80_Percentile','Mean','STD','SE']])
y = np.asarray(df['Facies'])
```

- **validation :**

Validasi membagi data (X) menjadi dua subset: satu untuk pelatihan (Xtrain) dan satu lagi untuk pengujian (Xtest)

```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.1, random_state=4)
print ('Train set:', X_train.shape, y_train.shape)
print ('Test set:', X_test.shape, y_test.shape)
```

# HASIL

## Parameter sampling dari data seismik mentah :

- **sampling\_rate (frekuensi sampel per detik): 500**
- **num\_samples (jumlah sampel): 1601**
- **duration (durasi total data dalam detik): 3.202**

## Hasil statistik data dari fitur seismik yang sudah di definisikan atributnya :

	20_Percentile	80_Percentile	Mean	STD	SE
0	-7.845215	-3.991427	-5.983986e+00	3.009774	3.009774
1	0.804879	4.897726	2.991993e+00	2.081666	2.081666
2	-0.992486	7.275672	2.991993e+00	4.642008	4.642008
3	7.291697	11.440921	9.331463e+00	4.301147	4.301147
4	7.130412	27.751522	1.810156e+01	10.235017	10.235017
5	3.211678	28.144201	1.510956e+01	10.746774	10.746774
6	-0.067816	6.351002	2.991993e+00	3.149736	3.149736
7	4.750443	24.305034	1.510956e+01	8.918083	8.918083
8	3.532032	8.604989	6.133586e+00	2.537654	2.537654
9	-1.874969	1.874969	-1.057355e-17	2.421870	2.421870
10	-21.179182	-2.092812	-1.196797e+01	9.094452	9.094452

Note : Terdapat 59 hasil statistik dari fitur seismik sesuai atribut data sampling yang kita gunakan

```
import obspy
from obspy import read

seismic_data = read('C:/Users/ACER/Downloads/all_seismic_section/XL3560_2d_crossline.sgy')

seismic_trace = seismic_data[0] # Assuming it's a single trace

# Mengetahui Sampling parameters dari data seismik mentah
sampling_rate = seismic_trace.stats.sampling_rate
num_samples = len(seismic_trace)
duration = num_samples / sampling_rate

print("sampling rate: ", sampling_rate)
print("number of samples: ", num_samples)
print("duration: ", duration)
#t = np.arange(num_samples) / sampling_rate

sampling rate: 500.0
number of samples: 1601
duration: 3.202
```

```
import pandas as pd

df_instan_phase = pd.DataFrame(statistics_array1, columns=['20_Percentile', '80_Percentile', 'Mean', 'STD', 'SE'])
print(df_instan_phase)

import pandas as pd

df_cosine_phase = pd.DataFrame(statistics_array2, columns=['20_Percentile', '80_Percentile', 'Mean', 'STD', 'SE'])
print(df_cosine_phase)

result_df = pd.merge(df_instan_phase, df_cosine_phase, how='outer')
print(result_df)
```

# HASIL

## Hasil Statistik terhadap setiap Fasies Seismik

Didapatkan 60 data statistik yang mengklasifikasikan setiap fasies seismik hasil sampling. Gambar disamping sebagai representasi dari 25 data statistik yang terkласifikasi di 3 fasies seismik

```
In [34]: 1 yhat = clf.predict(X_test)
2 print(X_test)
3 yhat [0:7]

[[ 7.13041229e+00  2.77515224e+01  1.81015577e+01  1.02350170e+01
  1.02350170e+01]
 [ 2.62125148e+00  3.66193382e+00  3.14159265e+00  7.76951985e-01
  7.76951985e-01]
 [-8.95944230e-01  8.14109361e-01  7.53947663e-02  8.12342056e-01
  8.12342056e-01]
 [-1.11044591e+00  1.11044591e+00  7.93016446e-17  1.86310427e+00
  1.86310427e+00]
 [-1.54044133e+00  1.76142262e+00  1.49599650e-01  1.96728935e+00
  1.96728935e+00]
 [-8.10450394e-01  8.10450394e-01  -2.64338815e-17  1.53745537e+00
  1.53745537e+00]]
```

Out[34]: array([1, 1, 2, 3, 1, 3], dtype=int64)

```
In [35]: 1 from sklearn.metrics import classification_report, confusion_matrix
2 import itertools
3
4 print (classification_report(y_test, yhat))
```

	precision	recall	f1-score	support
1	0.33	1.00	0.50	1
2	0.00	0.00	0.00	1
3	1.00	0.50	0.67	4
accuracy			0.50	6
macro avg	0.44	0.50	0.39	6
weighted avg	0.72	0.50	0.53	6

	20_Percentile	80_Percentile	Mean	STD	SE	Facies
0	-7.845215	-3.991427	-5.983986e+00	3.009774	3.009774	1
1	0.804879	4.897726	2.991993e+00	2.081666	2.081666	1
2	-0.992486	7.275672	2.991993e+00	4.642008	4.642008	1
3	7.291697	11.440921	9.331463e+00	4.301147	4.301147	1
4	7.130412	27.751522	1.810156e+01	10.235017	10.235017	1
5	3.211678	28.144201	1.510956e+01	10.746774	10.746774	1
6	-0.067816	6.351002	2.991993e+00	3.149736	3.149736	1
7	4.750443	24.305034	1.510956e+01	8.918083	8.918083	1
8	3.532032	8.604989	6.133586e+00	2.537654	2.537654	1
9	-1.874969	1.874969	-1.057355e-17	2.421870	2.421870	1
10	-21.179182	-2.092812	-1.196797e+01	9.094452	9.094452	2
11	1.301343	4.796067	2.991993e+00	2.224087	2.224087	2
12	-1.540441	1.761423	1.495997e-01	1.967289	1.967289	2
13	4.703729	12.047278	9.125579e+00	4.068107	4.068107	2
14	3.835175	8.685755	6.133586e+00	2.171716	2.171716	2
15	-8.364041	-4.166009	-5.983986e+00	3.035779	3.035779	2
16	-0.463517	0.463517	-7.930164e-18	0.826998	0.826998	2
17	2.408702	9.783755	5.983986e+00	3.500077	3.500077	2
18	3.771701	8.426560	6.133586e+00	2.171730	2.171730	2
19	5.552852	12.803855	9.125579e+00	3.782655	3.782655	2
20	2.621251	3.661934	3.141593e+00	0.776952	0.776952	3
21	-4.719482	-1.526582	-2.991993e+00	1.913118	1.913118	3
22	2.366382	10.199988	6.133586e+00	4.539479	4.539479	3
23	-6.142637	0.160106	-2.842393e+00	3.480160	3.480160	3
24	-1.110446	1.110446	7.930164e-17	1.863104	1.863104	3
25	-7.732718	2.391944	-2.842393e+00	5.044787	5.044787	3

# KESIMPULAN

1. Facies yang digunakan ada 3 (ada 3 facies karena lingkungan pengendapannya sallow marine)
2. Facies ini digunakan untuk klasifikasi struktur geologi
3. Atribut seismik yang digunakan dalam pengolahan data adalah cosinus-phase dan instantaneous-phase
4. Ada beberapa atribut yang tidak bisa digunakan karena dimensi sampling faciesnya kecil terhadap volume seismik
5. hasil yang didapat dari pengolahan data seismik memiliki akurasi yang tidak bagus karena atribut yang digunakan tidak banyak dan sampling yang sedikit.

