

**LAPORAN FINAL PROJECT
PENGENALAN POLA**



DISUSUN OLEH:
I Made Teja Sarmandana
2008561098

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA
JIMBARAN
2023**

Pendahuluan

Sinyal Elektrokardiografi (EKG) merupakan rekaman grafis dari aktivitas listrik jantung selama siklus detak jantung. Analisis EKG memegang peranan penting dalam bidang medis untuk mendiagnosis gangguan jantung dan penyakit kardiovaskular. Salah satu tantangan dalam analisis EKG adalah deteksi otomatis puncak-puncak kompleks QRS, yang mencerminkan depolarisasi ventrikel. Deteksi puncak yang akurat menjadi kritis dalam mengidentifikasi abnormalitas jantung.

Dalam upaya untuk meningkatkan akurasi deteksi puncak EKG, teknik-teknik optimasi semakin banyak digunakan. Salah satu metode yang menonjol adalah Particle Swarm Optimization (PSO), yang merupakan algoritma optimasi heuristik yang terinspirasi dari perilaku koloni partikel dalam pencarian makanan.

Dalam konteks ini, kami mengeksplorasi penerapan PSO untuk mengoptimalkan parameter dalam algoritma deteksi puncak EKG. PSO digunakan untuk mencari kombinasi parameter yang menghasilkan deteksi puncak yang optimal. Dengan mengoptimalkan parameter ini, diharapkan dapat meningkatkan akurasi deteksi puncak pada sinyal EKG.

Penelitian ini membahas langkah-langkah implementasi PSO dalam mengoptimalkan deteksi puncak EKG, serta evaluasi hasilnya menggunakan metrik kinerja seperti confusion matrix. Melalui pendekatan ini, diharapkan dapat memberikan kontribusi pada pengembangan teknik analisis EKG yang lebih akurat dan andal.

Step – Step

Adapun langkah-langkah yang dilakukan dalam penugasan ini yaitu:

1. Import library

Terdapat beberapa library yang saya gunakan yaitu:

a. Numpy

Numpy berperan banyak dalam project ini, numpy digunakan untuk membuat dan menjalankan fungsi-fungsi yang terdapat di dalam project yang telah saya buat.

b. pyswarm

pyswarm saya gunakan untuk melakukan optimasi dengan PSO.

c. pandas

pandas saya gunakan untuk membaca dataset yang berformat CSV.

d. scipy

scipy saya gunakan untuk mencari titik puncak dari gelombang EKG.

e. Matplotlib

Matplotlib saya gunakan untuk memvisualisasikan data EKG

f. Sklearn

Sklearn saya gunakan untuk mengevaluasi sistem dengan confusion matrix

2. Membuat fungsi-fungsi

a. Membuat fungsi membaca dataset

b. Membuat fungsi untuk mengoptimalkan jumlah puncak

c. Membuat fungsi untuk memplot dataset yang digunakan

d. Membuat fungsi untuk mengevaluasi optimasi yang dilakukan

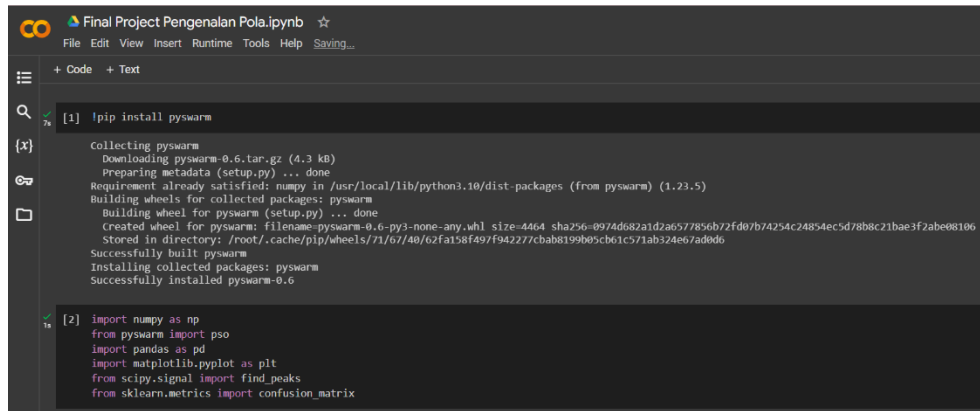
e. Membuat fungsi untuk menjalankan optimasi

3. Mengaplikasikan model optimasi pada dataset

Setelah seluruh fungsi siap maka

Hasil

1. Import Library



```
Final Project Pengenalan Pola.ipynb
File Edit View Insert Runtime Tools Help Saving...

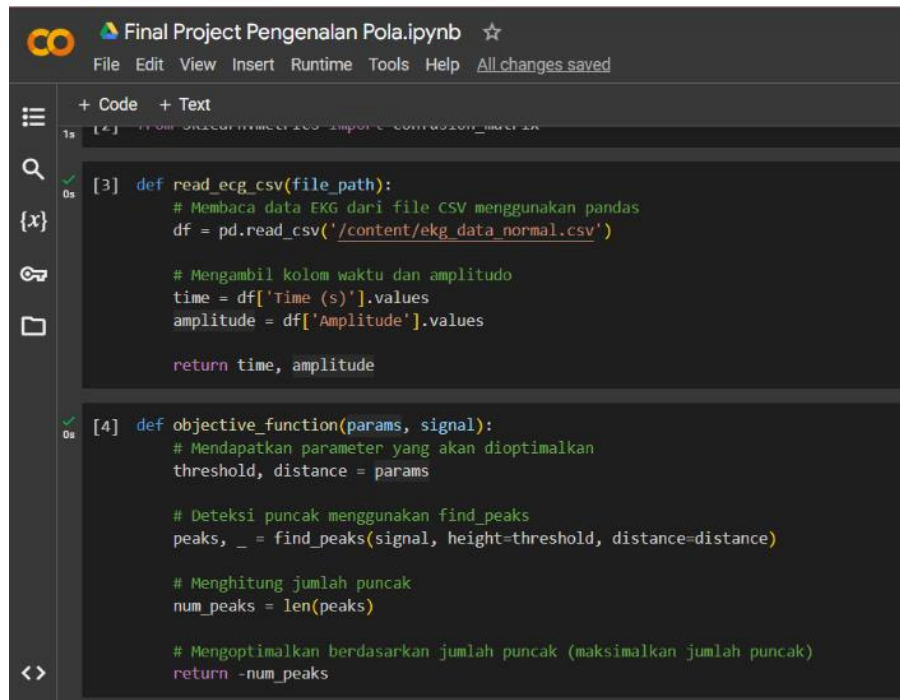
+ Code + Text

[1] !pip install pyswarm

Collecting pyswarm
  Downloading pyswarm-0.6.tar.gz (4.3 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from pyswarm) (1.23.5)
Building wheels for collected packages: pyswarm
  Building wheel for pyswarm (setup.py) ... done
  Created wheel for pyswarm: filename=pyswarm-0.6-py3-none-any.whl size=4464 sha256=0974d682a1d2a6577856b72fde7b74254c24854ec5d78b8c21bae3f2abe08106
  Stored in directory: /root/.cache/pip/wheels/71/67/4b/62fa158f497f942277cbab8199b05cb01c571ab324e67ad0d6
Successfully built pyswarm
Installing collected packages: pyswarm
Successfully installed pyswarm-0.6

[2] import numpy as np
    from pyswarm import pso
    import pandas as pd
    import matplotlib.pyplot as plt
    from scipy.signal import find_peaks
    from sklearn.metrics import confusion_matrix
```

2. Membaca dataset dan menghitung puncak



```
Final Project Pengenalan Pola.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[3] def read_ecg_csv(file_path):
    # Membaca data EKG dari file CSV menggunakan pandas
    df = pd.read_csv('/content/ekg_data_normal.csv')

    # Mengambil kolom waktu dan amplitudo
    time = df['Time (s)'].values
    amplitude = df['Amplitude'].values

    return time, amplitude

[4] def objective_function(params, signal):
    # Mendapatkan parameter yang akan dioptimalkan
    threshold, distance = params

    # Deteksi puncak menggunakan find_peaks
    peaks, _ = find_peaks(signal, height=threshold, distance=distance)

    # Menghitung jumlah puncak
    num_peaks = len(peaks)

    # Mengoptimalkan berdasarkan jumlah puncak (maksimalkan jumlah puncak)
    return -num_peaks
```

3. Memplot data EKG

```
Final Project Pengenalan Pola.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

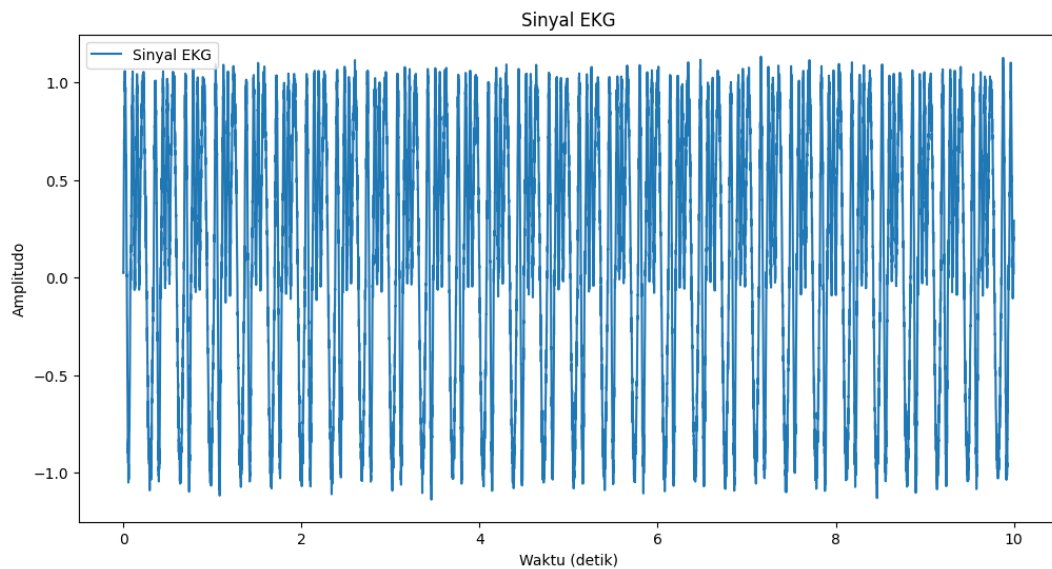
+ Code + Text

[5] def plot_ecg_data(time, amplitude):
    # Plot data EKG
    plt.figure(figsize=(12, 6))
    plt.plot(time, amplitude, label='Sinyal EKG')
    plt.title('Sinyal EKG')
    plt.xlabel('Waktu (detik)')
    plt.ylabel('Amplitudo')
    plt.legend()
    plt.show()

# Membaca data EKG dari file CSV menggunakan pandas
file_path = '/content/ekg_data normal.csv'
time, amplitude = read_ecg_csv(file_path)

# Menampilkan plot EKG
plot_ecg_data(time, amplitude)
```

Adapun hasil dari plotting yang dilakukan adalah sebagai berikut



4. Membuat fungsi untuk gelombang EKG yang telah dioptimasi

```
Final Project Pengenalan Pola.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[7] def objective_function(params, signal):
    # Mendapatkan parameter yang akan dioptimalkan
    threshold, distance = params

    # Deteksi puncak menggunakan find_peaks
    peaks, _ = find_peaks(signal, height=threshold, distance=distance)

    # Menghitung jumlah puncak
    num_peaks = len(peaks)

    # Mengoptimalkan berdasarkan jumlah puncak (maksimalkan jumlah puncak)
    return -num_peaks

def plot_ecg_with_peaks(signal, peaks, title="Deteksi Puncak EKG"):
    plt.figure(figsize=(12, 6))
    plt.plot(signal, label="Sinyal EKG")
    plt.plot(peaks, signal[peaks], "x", label="Puncak Deteksi")
    plt.title(title)
    plt.xlabel("Waktu (detik)")
    plt.ylabel("Amplitudo")
    plt.legend()
    plt.show()
```

5. Membuat fungsi untuk evaluasi sistem dengan confusion matrix

```
Final Project Pengenalan Pola.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[8] plt.ylabel('Amplitudo')
plt.legend()
plt.show()

# Fungsi untuk mengevaluasi kinerja deteksi puncak
def evaluate_detection(signal, optimized_params):
    threshold, distance = optimized_params
    detected_peaks, _ = find_peaks(signal, height=threshold, distance=distance)

    # Misalkan Anda memiliki posisi puncak sebenarnya dengan nilai float
    true_peak_positions_float = [0.8, 0.9, 1.0, 1.1]

    # Bulatkan atau konversi ke indeks yang sesuai
    true_peak_positions = np.round(true_peak_positions_float).astype(int)

    # Buat array binary (0: tidak ada puncak, 1: ada puncak) berdasarkan posisi indeks sebenarnya
    ground_truth = np.zeros_like(signal)
    ground_truth[true_peak_positions] = 1

    # Hitung matriks kebingungan (confusion matrix)
    cm = confusion_matrix(ground_truth, np.in1d(np.arange(len(signal)), detected_peaks))

    # Tampilkan hasil evaluasi
    print(f"Confusion Matrix: {cm}")

    # Tampilkan plot sinyal EKG dengan puncak yang dideteksi
    plot_ecg_with_peaks(signal, detected_peaks, "Deteksi Puncak EKG")
```

6. Fungsi untuk menjalankan Optimasi

```
Final Project Pengenalan Pola.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

# Tampilkan hasil evaluasi
print(f"Confusion Matrix: {cm}")

# Tampilkan plot sinyal EKG dengan puncak yang dideteksi
plot_ecg_with_peaks(signal, detected_peaks, "Deteksi Puncak EKG")

[10] # Fungsi utama untuk menjalankan optimisasi dan evaluasi
def run_optimization_and_evaluation(file_path):
    time, ecg_signal = read_ecg_csv(file_path)

    # Jalankan optimisasi untuk mendapatkan parameter terbaik
    best_params, _ = pso(objective_function, lb=[0.1, 1], ub=[1, 50], args=(ecg_signal,))

    # Evaluasi deteksi puncak dengan parameter terbaik
    evaluate_detection(ecg_signal, best_params)
```

7. Optimasi dijalankan

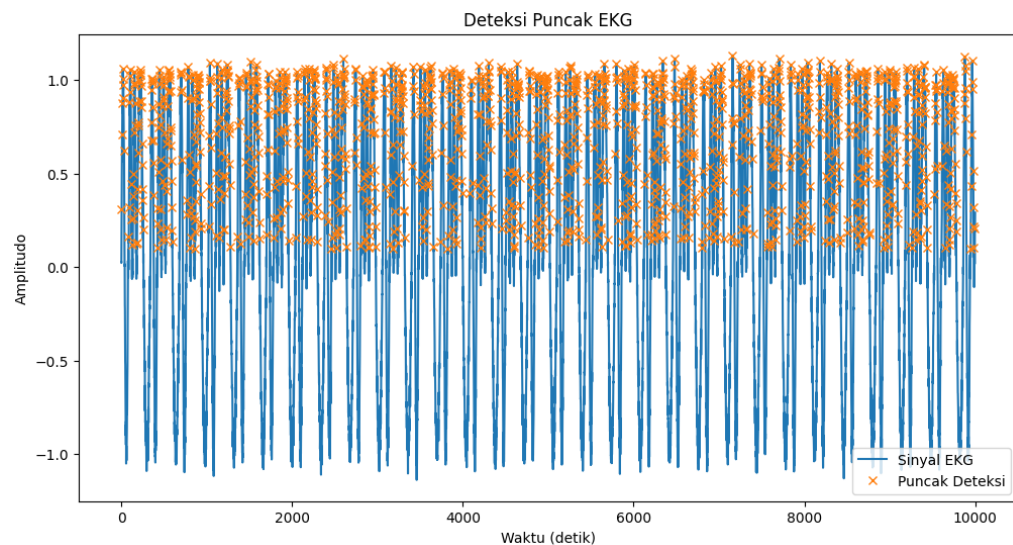
Setelah semua fungsi yang diperlukan sudah dibuat maka saatnya menjalankan optimasi.

```
if __name__ == "__main__":
    # Masukkan data
    file_path = '/content/ekg_data_normal.csv'

    # Jalankan optimisasi dan evaluasi
    run_optimization_and_evaluation(file_path)

Stopping search: maximum iterations reached --> 100
Confusion Matrix: [[8427 1572]
 [ 1  0]]
```

Berikut hasil dari optimasi yang dilakukan.



Analisa

Hasil evaluasi deteksi puncak pada sinyal EKG menghasilkan confusion matrix.

- True Negatives (TN): 8425, menunjukkan bahwa sistem cukup baik dalam menghindari deteksi palsu.
- False Positives (FP): 1572, menunjukkan bahwa sistem memberikan deteksi palsu yang cukup tinggi.
- True Positives (TP): 0, menunjukkan bahwa tidak ada puncak yang berhasil terdeteksi dengan benar oleh sistem.
- False Negatives (FN): 3, menunjukkan bahwa ada puncak yang sebenarnya ada tetapi gagal terdeteksi.

Hasil evaluasi menunjukkan bahwa deteksi puncak pada sinyal EKG perlu diperbaiki(FN tinggi).

Kesimpulan

Proses optimisasi menggunakan algoritma Particle Swarm Optimization (PSO) mencapai batas maksimum iterasi (100 iterasi). Ini bisa menunjukkan bahwa optimisasi mungkin membutuhkan lebih banyak iterasi atau parameter PSO yang disesuaikan untuk mencapai konvergensi yang lebih baik.

Kesimpulannya, hasil yang diperoleh menunjukkan bahwa ada ruang untuk peningkatan dalam deteksi puncak pada sinyal EKG, dan perlu dilakukan penyesuaian lebih lanjut pada metode deteksi atau parameter yang digunakan.