**LAPORAN TUGAS IOT DENGAN KEMAMPUAN CERDAS**

**FILTERISASI SINYAL DAN DATA PROSESING MENGGUNAKAN MVA DAN LPF**

laporan ini dibuat untuk memenuhi kebutuhan nilai tugas mata kuliah IoT Dengan Kemampuan Cerdas

Dosen mata kuliah : **Viddi Mardiansyah, S.Si., M.T.**

**CII3O3-IF-43-PIL-CPS-02**



disusun oleh

**NAUFAL ADRIAN HIDAYAT (1301204056)**

**IF-44-11**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY**

**2022**

# BAB 1 PENDAHULUAN

## Latar Belakang

Internet of Things (IoT) adalah konsep dimana perangkat elektronik yang terhubung dengan internet dapat saling berkomunikasi untuk mengumpulkan dan mengirimkan data secara otomatis. Salah satu aplikasi IoT adalah pada filterisasi sinyal dan pengolahan data menggunakan teknologi cerdas seperti Moving Average (MVA) dan Low Pass Filter (LPF).

Signal Data Processing adalah teknologi pengolahan data yang digunakan untuk memproses sinyal analog dan digital. Tujuannya adalah untuk menghasilkan informasi yang lebih bermanfaat dan mudah dipahami. LPF adalah jenis filter sinyal yang digunakan untuk meredam atau menghilangkan komponen frekuensi tinggi pada sinyal. Sementara itu, MVA adalah teknik pengolahan data yang digunakan untuk menghaluskan data dan menghilangkan noise.

Pada tugas ini, kita akan menggunakan teknologi IoT dan kemampuan cerdas seperti LPF dan MVA untuk filterisasi sinyal dan pengolahan data. Dalam hal ini, kita akan memplot data mentah dan membandingkannya dengan data yang telah difilter dengan LPF dan MVA. Pada tugas ini, kita akan menggunakan teknologi IoT dan kemampuan cerdas seperti LPF dan MVA untuk filterisasi sinyal dan pengolahan data, dengan tujuan untuk menghasilkan informasi yang lebih bermanfaat dan mudah dipahami. Pada tugas ini, kami diminta menggunakan dataset yang berasal dari link berikut.

<https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer>

Setelah membuka dan mengunduh dataset tersebut, kami akan memilih salah satu dataset yang terkumpul di dalamnya untuk melakukan Filtering Sinyal Data Processing dengan menggunakan LPF dan MVA. Kemudian memilih salah satu kolom di dalam dataset yang ada seperti kolom B atau C atau D (X, y, Z axis dari accelerometer) yang sudah dilakukan filterisasi pada raw data yang akan digunakan/ Beberapa pertanyaan yang akan dijawab dalam tugas ini adalah sebagai berikut:

-) Plot data mentah dan bandingkan dengan LPF dan MVA Filter.

-) Frekuensi cut-off (fc) apa yang Anda gunakan untuk LPF? Bagaimana Anda menentukan fc tersebut? (tolong sediakan plot data yang sudah difilter dengan frekuensi cut-off yang Anda gunakan)

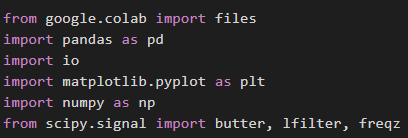
-) Berapa ukuran jendela yang Anda gunakan dalam MVA? (tolong pilih 3 ukuran jendela yang berbeda, kemudian plot dan bandingkan hasilnya). Ukuran jendela mana yang memberikan plot data yang halus? Mengapa?

# BAB 2 PEMBAHASAN

## Pembahasan Codingan Filter LPF dan MVA

Pada bagian berikut, akan dijelaskan mengenai codingan yang digunakan memakai *Google Collab* untuk membuat bagaimana filter dari LPF dan MVA tersebut bekerja.

### Import Library

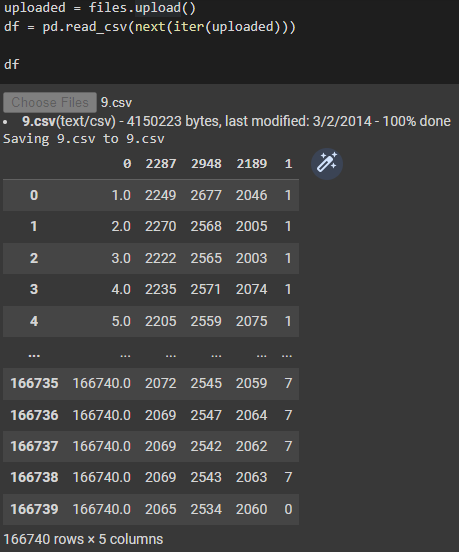


**Gambar 2.1** : Library pada Google Collabs.

Pada bagian ini, beberapa *library* digunakan untuk membantu kelancaran pembuatan tugas ini menjadi lebih baik. Kombinasi dari library-library tersebut memungkinkan untuk melakukan filterisasi sinyal data dengan menggunakan teknik LPF dan MVA, serta melakukan visualisasi dari hasil filterisasi tersebut. Berikut penjelasan dari *library* yang digunakan.

* google.colab: *library* ini digunakan untuk mengupload dan membaca file dari Google Colab, sehingga data dapat diakses dan diproses secara online.
* pandas: library ini digunakan untuk mengolah dan memanipulasi data yang sudah diupload.
* matplotlib: library ini digunakan untuk membuat visualisasi dari data dalam bentuk grafik, sehingga memudahkan untuk membandingkan hasil filterisasi sinyal data.
* numpy: library ini digunakan untuk melakukan operasi matematika seperti array, matriks, dan operasi lainnya pada data.
* scipy.signal.butter: fungsi ini digunakan untuk mendefinisikan filter Butterworth, yaitu jenis filter LPF yang sangat populer dan sering digunakan pada filterisasi sinyal data.
* scipy.signal.lfilter: fungsi ini digunakan untuk menerapkan filter LPF dan MVA pada data, sehingga bisa diproses lebih lanjut.
* scipy.signal.freqz: fungsi ini digunakan untuk menghitung respon frekuensi dari filter yang sudah dibuat, sehingga dapat dilihat frekuensi cut-off dari filter tersebut.

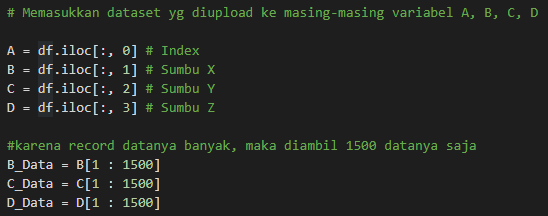
### Input File

****

**Gambar 2.2 :** Bagian Input File pada Google Collabs.

Pada bagian ini, kode program bermaksud untuk meminta kita memilih file mana yang akan diunggah dan dipakai dalam pengerjaan tugas. Setelah diunggah, file tersebut akan memasukkan dan menampilkan isi dataset yang sudah diunggah ke dalam folder Google Collab. Adapun dataset yang digunakan untuk tugas ini adalah “9.csv”.

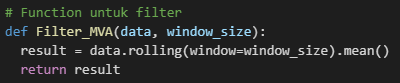
### Pembuatan Data dan Kolom Data

****

**Gambar 2.3** : Proses pembuatan masing-masing Kolom untuk menentukan Index dan Sumbu serta pengambilan data untuk melakukan filterisasi.

Pada bagian ini, akan dilakukan pembuatan masing-masing kolom untuk menentukan mana data yang Index, sumbu X, sumbu Y, dan sumbu Z dari dataset 9.csv yang dipakai. Karena data pada dataset tersebut recordnya terlalu banyak (Bisa dilihat pada gambar 2.2) yang mencapai baris ke-166740 maka akan diambil datanya menjadi 1500 record dari banyaknya record yang ada pada dataset yang digunakan. Data diambil dengan membagi 1:1500 dari tiap kolom yang dipakai untuk masing-masing sumbu, seperti Kolom B, C, dan D.

### Filterisasi MVA (Moving Average)

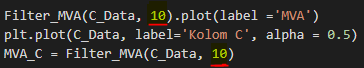


**Gambar 2.4 :** *Function* untuk melakukan filterisasi MVA.

Moving Average (MVA) adalah jenis filter yang mengambil rata-rata nilai sinyal pada jendela waktu tertentu yang bergerak secara terus-menerus. Filter ini digunakan untuk menghaluskan sinyal dan menghilangkan noise atau fluktuasi kecil pada sinyal. MVA sering digunakan pada analisis time-series dan data sensor.

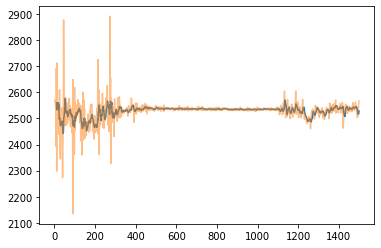
Pada bagian ini, dilakukannya pembuatan function untuk membuat filter dari MVA yang akan digunakan. Kemudian akan dilakukannya Filter pada kolom yang ingin digunakan dengan memasukkan tiga nilai *window size* yang berbeda-beda. Fungsi dari *Window Size* pada function ini adalah sebagai ukuran jendela (*rolling window*) pada data yang digunakan. Jadi, ketika melakukan filter, data akan dibagi menjadi beberapa jendela (*Window*) dengan ukuran tertentu (*Window Size*) lalu dihitung rata-ratanya untuk menghasilkan data baru yang lebih halus. Kolom yang akan dipakai untuk melakukan Filter ini adalah Kolom C.

#### *Window Size* = 10

****

**Gambar 2.4.1 :** Filter untuk *Window Size* = 10.

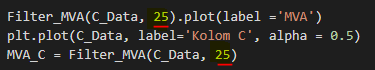
Pada tahap ini, MVA akan melakukan filterisasi Kolom C dengan menggunakan *window size* 10. Kemudian didapat hasil sebagai berikut.



**Gambar 2.4.2 :** Hasil Filterisasi MVA pada *Window Size* = 10.

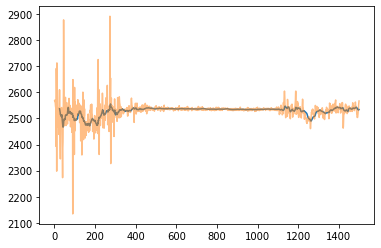
Pada hasil gambar tersebut, kita mendapat data baru yang lebih halus pada bagian garis yang berwarna gelap. Bisa diketahui bahwa data yang baru tersebut (berwarna gelap pada gambar 2.4.2) data yang sudah dilakukan filter MVA dan terlihat lebih baik dilihat dibanding data asli yang berwarna cerah (kuning) pada gambar 2.4.2.

#### *Window Size* = 25

****

**Gambar 2.4.3 :** Filter untuk *Window Size* = 25.

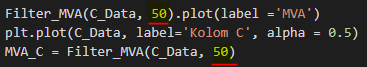
Pada tahap ini, MVA akan melakukan filterisasi Kolom C dengan menggunakan *window size* 25. Kemudian didapat hasil sebagai berikut.



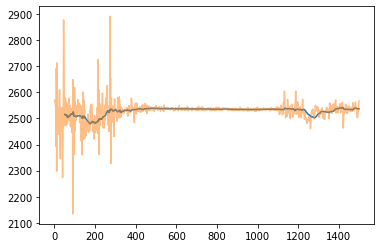
**Gambar 2.4.4 :** Hasil Filterisasi MVA pada *Window Size* = 25.

Pada hasil gambar tersebut, kita mendapat data baru yang lebih halus pada bagian garis yang berwarna gelap. Bisa diketahui bahwa data yang baru tersebut (berwarna gelap pada gambar 2.4.4) data yang sudah dilakukan filter MVA dan terlihat lebih baik dilihat dibanding data asli yang berwarna cerah (kuning) pada gambar 2.4.4.

#### *Window Size* = 50

****

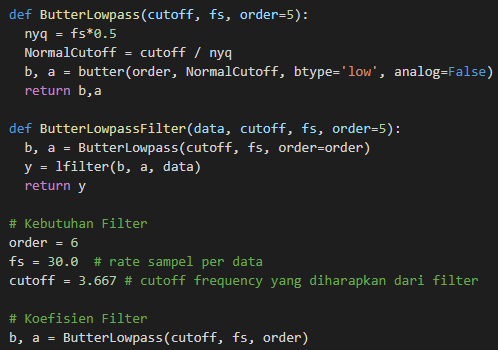
**Gambar 2.4.5 :** Filter untuk *Window Size* = 50.

Pada tahap ini, MVA akan melakukan filterisasi Kolom C dengan menggunakan *window size* 50. Kemudian didapat hasil sebagai berikut.

**Gambar 2.4.6 :** Hasil Filterisasi MVA pada *Window Size* = 50.

Pada hasil gambar tersebut, kita mendapat data baru yang lebih halus pada bagian garis yang berwarna gelap. Bisa diketahui bahwa data yang baru tersebut (berwarna gelap pada gambar 2.4.6) data yang sudah dilakukan filter MVA dan terlihat lebih baik dilihat dibanding data asli yang berwarna cerah (kuning) pada gambar 2.4.6.

### Filterisasi LPF (Low Pass Filter)



**Gambar 2.5 :** *Function* untuk melakukan filterisasi LPF.

Low Pass Filter (LPF) adalah jenis filter yang membiarkan frekuensi rendah melewati filter dan memblokir atau mengurangi frekuensi tinggi. Filter ini digunakan untuk mempertahankan atau meninggalkan komponen sinyal yang memiliki frekuensi rendah dan menghilangkan atau mengurangi komponen sinyal yang memiliki frekuensi tinggi. LPF sering digunakan pada pengolahan sinyal suara dan gambar.

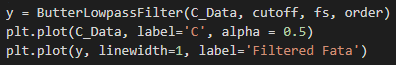
Pada bagian ini, dilakukannya pembuatan *function* untuk membuat filter dari LPF yang akan digunakan. Terdapat dua *function* untuk melakukan filterisasi LPF yang akan digunakan dan satu baris kode yang dipakai untuk menghitung koefisien filter. Pada LPF ini, kolom yang digunakan sama dengan kolom yang dipakai oleh MVA yakni kolom C. Berikut penjelasan dari kode program di atas.

Pertama adalah “ButterLowPass(cutoff, fs, order=5)”. *Function* ini digunakan untuk membuat filter Butterworth LPF dengan parameter “cutoff” (frekuensi cut-off), “fs” (frekuensi sampling), dan “order” (tingkat orde filter). Pertama-tama, *function* ini menghitung nilai frekuensi “Nyquist” (nyq) dari “fs”, kemudian menghitung nilai “NormalCutoff” yaitu “cutoff” dibagi “nyq”. Selanjutnya, *function* ini menggunakan fungsi butter dari library “scipy.signal” untuk membuat filter Butterworth dengan tingkat orde “order”, frekuensi cut-off “NormalCutoff”, dan tipe filter low-pass (“btype='low'”). *Function* ini mengembalikan koefisien filter “b” dan “a” yang akan digunakan dalam *function* “ButterLowpassFilter”.

Kedua adalah “ButterLowpassFilter(data, cutoff, fs, order=5)”. *Function* ini digunakan untuk menerapkan filter LPF pada data. *Function* ini memanggil *function* “ButterLowpass” untuk mendapatkan koefisien filter “b” dan “a”, kemudian menggunakan fungsi “lfilter” dari library “scipy.signal” untuk menerapkan filter pada data. *Function* ini mengembalikan data yang sudah difilter.

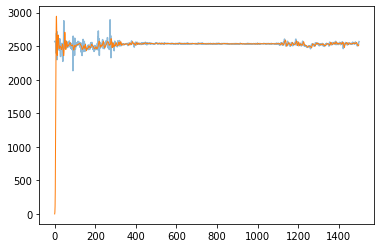
Ketiga adalah *line* kode “b, a = ButterLowpass(cutoff, fs, order)”. *Line* kode ini digunakan untuk menghitung koefisien filter “b” dan “a” dengan menggunakan *function* “ButterLowpass”. Koefisien filter tersebut akan digunakan dalam proses filterisasi sinyal data menggunakan fungsi “lfilter”.

#### Hasil filterisasi LPF

****

**Gambar 2.5.1 :** Filter untuk LPF.

Pada tahap ini, LPF akan melakukan filterisasi pada Kolom C dengan kebutuhan filter pada tahap sebelumnya yang telah dijelaskan dan dipaparkan pada gambar 2.5. Kemudian didapat hasil sebagai berikut.



**Gambar 2.5.2 :** Hasil Filterisasi LPF.

Pada hasil gambar tersebut, kita memplot data asli yakni “C\_Data” dengan data yang sudah difilter dengan Buttorworth LPF memakai *function* “ButterLowPassFilter” pada tahap sebelumbya. Data hasil filter ditandai dengan garis yang berwarna *orange* serta telah diberi label “Filtered Data”.

## Hasil Analisis

Pada bagian ini, akan dilakukan analisis dari tiap-tiap hasil filterisasi.

1. Pada filter LPF, nilai cutoff frequency yang saya pilih adalah 3.667. Nilai ini digunakan berdasarkan dari analisis sinyal awal dan pemilihan dari frekuensi yang relevan dengan aktivitas manusia. Pada gambar 2.5.2, didapat bahwa filter LPF dengan cutoff frequency = 3.667 dan sampling rate = 30.0 dengan memakai metode LPF Butterworth sudah berhasil membuat data sinyal yang lebih halus dan mudah diproses.
2. Pada filter MVA, dilakukan dengan pembagian tiga macam nilai *window siz*e yang digunakan. Adapun nilai dari *window size* tersebut adalah 10, 25, dan 50. Kita bisa memahami pada pengertian MVA yakni jenis filter yang mengambil rata-rata nilai sinyal pada jendela waktu tertentu yang bergerak secara terus-menerus, hubungannya sangat dekat dengan nilai yang berbeda pada tiap *window size* yang digunakan. Jika diperhatikan kembali pada hasil *window size* = 50, hasil filter terlihat lebih halus dan lebih mulus dibandingkan dengan nilai *window size* yang lainnya. Hal tersebut membuktikan bahwa semakin besar nilai *window size* maka akan semakin baik sinyal atau data yang akan difilternya.

# BAB 3 KESIMPULAN

Kesimpulan yang bisa didapat dari beberapa kegiatan yang saya lakukan adalah bahwa LPF dan MVA merupakan teknik pengolahan sinyal yang dipakai untuk mengurangi *noise* maupun gangguan pada data. LPF (Low Pass Filter) digunakan untuk memfilter sinyal data agar frekuensi tinggi pada data dapat dihilangkan, sehingga data menjadi lebih halus dan dapat diproses dengan lebih baik. Sementara itu, MVA (Moving Average Filter) digunakan untuk memfilter sinyal data dengan cara mengambil rata-rata beberapa nilai sebelumnya. *Window size* digunakan untuk menentukan jumlah data yang digunakan untuk menghitung rata-rata. Semakin besar *window size*, semakin halus plot data yang dihasilkan.

Pada hasil analisis, bisa dibandingkan plot dari data asli dengan yang sudah difilter untuk melihat perbedaannya. Pada LPF, frekuensi cut-off dapat ditentukan dengan cara melihat plot data asli dan plot filtered data, kemudian memilih nilai frekuensi yang dianggap tepat. Pada MVA, dapat dipilih beberapa nilai *window size* yang berbeda untuk dibandingkan hasilnya dan memilih *window size* yang menghasilkan plot data ataupun gambar garis data yang paling halus. Tetapi harus dilakukan lagi penyesuaian dan pengujian lebih lanjut tergantung dari tujuan analisis yang diharapkan tercapai

**Lampiran**

**Link Codingan (Google Collabs) :**

[*https://colab.research.google.com/drive/1ms5C4x\_DhfdSBebxGC\_9YVcjBKVPUtfL?usp=sharing*](https://colab.research.google.com/drive/1ms5C4x_DhfdSBebxGC_9YVcjBKVPUtfL?usp=sharing)

**Link Laporan :**

[*https://docs.google.com/document/d/1qc2GAc275j7kaCBY3gXMOQsTvkyOo5FS/edit?usp=sharing&ouid=100359807533626157288&rtpof=true&sd=true*](https://docs.google.com/document/d/1qc2GAc275j7kaCBY3gXMOQsTvkyOo5FS/edit?usp=sharing&ouid=100359807533626157288&rtpof=true&sd=true)