

# Laporan Tugas Akhir Perancangan Sistem Digital

Nama Anggota :

- Jason Adrian Mahalim
  - Muhamad Aliefian R.
  - Muhamad Rizky Febrianto
- 

## Pendahuluan

Pedometer merupakan sebuah alat untuk mengukur langkah ketika dipakai oleh pengguna. Terdapat 2 macam pedometer, yaitu pedometer mekanikal dan elektronik. Pedometer mekanikal menggunakan mekanisme seperti jam pendulum, ketika pengguna bergerak, maka pendulum akan mengayun satu kali dan dari ayunan tersebut langkah akan terhitung satu.

Beberapa pedometer elektronik menggunakan pendulum untuk menghitung langkah, namun ada juga pedometer elektronik yang menggunakan *accelerometer* untuk menghitung langkah. Cara pedometer elektronik dengan *accelerometer* bekerja adalah dengan menerima sinyal dari langkah pengguna, mengolah sinyal tersebut sehingga tidak memiliki banyak *noise*, dan akhirnya ada proses kalkulasi langkah yang dilakukan oleh komputer pada pedometer tersebut.

Pada tugas akhir kali ini, kami membuat coding matlab dan filter yang nantinya bisa digunakan untuk membuat pedometer. Coding matlab dan filter yang kami buat berguna untuk mereduksi dan menganalisis *noise* yang terdapat pada data yang kami ambil.

## Requirement

Ada beberapa peralatan yang dibutuhkan untuk membuat Coding pedometer sederhana, peralatan tersebut yaitu:

- Smartphone dengan *accelerometer*
- Aplikasi sensor untuk menangkap data
- Matlab

## Data

Data diambil menggunakan Smartphone yang memiliki aplikasi sensor. Terdapat 12 data yang diambil untuk mengetes tingkat akurat pedometer. Data yang diambil oleh kami adalah 10 langkah, 20 langkah, 30 langkah, dan variasi. Untuk variasi sendiri Jason mengambil 10 langkah berlari, Rizky mengambil 5 langkah berjalan & 5 berlari, dan Aliefian... .

## Source code dan Filter

Berikut ini adalah potongan *source code* dan filter yang kami gunakan untuk membuat pedometer:

### Source code

```
//inisialisasi
LRizky1020 = length(Rizky1020);
xRizky1020 = Rizky1020(:,1);
yRizky1020 = Rizky1020(:,2);
zRizky1020 = Rizky1020(:,3);
fsRizky1020 = 50;

//hitung magnititude
magRizky1020 = sqrt(sum(xRizky1020.^2 + yRizky1020.^2 + zRizky1020.^2, 2));
YmagRizky1020 = abs(fft(magRizky1020));

//analisis fft dari detrend magnititude
trendRizky1020 = detrend(magRizky1020);
fft_trendRizky1020 = abs(fft(trendRizky1020));
p2Rizky1020 = abs(fft_trendRizky1020/LRizky1020);
p1Rizky1020 = p2Rizky1020(1:LRizky1020/2+1);
p1Rizky1020(2:end-1)=2*p1Rizky1020(2:end-1);
fRizky1020 = fsRizky1020*(0:(LRizky1020/2))/LRizky1020;

//hasil dari sinyal yang sudah di filter
hasilRizky1020 = filtfilt(SOS, G, trendRizky1020);
```

### Filter

<b>Response Type</b> <input type="radio"/> Lowpass <input type="radio"/> Highpass <input checked="" type="radio"/> Bandpass <input type="radio"/> Bandstop <input type="radio"/> Differentiator <b>Design Method</b> <input checked="" type="radio"/> IIR Chebyshev Type II <input type="radio"/> FIR Equiripple	<b>Filter Order</b> <input checked="" type="radio"/> Specify order: 2 <input type="radio"/> Minimum order <b>Options</b> There are no optional parameters for this design method.	<b>Frequency Specifications</b> Units: Hz Fs: 50 Fstop1: 1.7 Fstop2: 2.0	<b>Magnitude Specifications</b> Units: dB Astop: 50
--	---	--	---

Dari *source code* diatas, kami menggunakan magnitude untuk mengkitung vektor dari sumbu x, y dan z. Kemudian, magnitude di detrend untuk merapikan data sehingga mudah untuk di fft. Hasil dari fft detrend tersebut nantinya akan digunakan untuk membuat filter. Terakhir, data kemudian di filter dengan menggunakan fungsi filtfilt dan parameter yang digunakan berasal dari design filter yang telah dibuat di atas.

Filter yang kami design merupakan filter *bandpass*, yang artinya kami memfilter signal yang ada di kiri dan kana frekuensi cutoff. Tipe filter yang kami gunakan adalah IIR Chebyshev type 2. Kemudian filter order yang kami gunakan adalah 2, dan frekuensi cutoff yang kami input 1,7 dan 2,1. Dan magnitude spesification yang digunakan adalah 50 dB.

## Analisis Data

Data yang kami dapatkan dan hasil yang kami peroleh dari perhitungan menggunakan *source code* matlab kami adalah sebagai berikut

	10 langkah	20 langkah	30 langkah	Variasi
Jason	13	23	31	13
Aliefian	9	19	28	23
Rizky	12	21	31	16

Spesifikasi langkah untuk variasi masing-masing anggota kelompok adalah sebagai berikut:

- Jason : 10 langkah berlari
- Aliefian : 5 langkah berjalan, 5 langkah berjalan jongkok
- Rizky : 5 langkah berjalan, 5 langkah berlari

Penempatan smartphone selama pengambilan data berada di tangan

Perhitungan persen error dengan menggunakan rumus berikut

$$Error = \frac{\text{experimental value} - \text{accepted value}}{\text{accepted value}} \times 100\%$$

Dari rumus tersebut didapatkan persentase error sebagai berikut

	10 langkah	20 langkah	30 langkah	Variasi
Jason	30%	15%	3.3%	30%
Aliefian	10%	5%	6.7%	130%
Rizky	20%	5%	3.3%	60%

Terdapat beberapa faktor yang menyebabkan error, seperti:

- Frekuensi cutoff yang kurang tepat untuk setiap data, karena kami hanya menggunakan 1 filter saja
- Design filter yang belum sempurna
- Perbedaan dari pola langkah dan tempo jalan setiap orang

## Kesimpulan

Pedometer yang kami buat masih memiliki tingkat error yang cukup besar, hal-hal tersebut karena frekuensi cutoff yang kurang tepat untuk setiap data, design filter yang belum sempurna, dan perbedaan dari pola langkah dan tempo jalan setiap orang

## Referensi

Helmenstine, T. (2014, May 16). *Calculate Percent Error*. Retrieved from Science Notes:  
<https://sciencenotes.org/calculate-percent-error/>

Woodford, C. (2018, September 2018). *Pedometers*. Retrieved from explainthatstuff!:  
<https://www.explainthatstuff.com/how-pedometers-work.html>