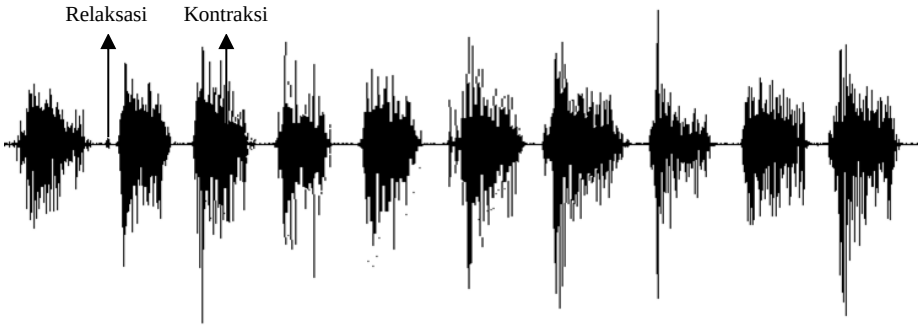
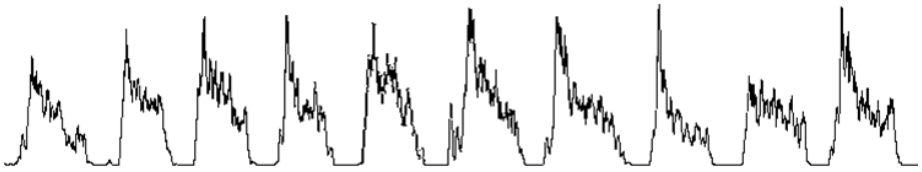
MODUL 1
PEMROSESAN SINYAL BIOPOTENSIAL

1.1 Electromyograph (EMG)



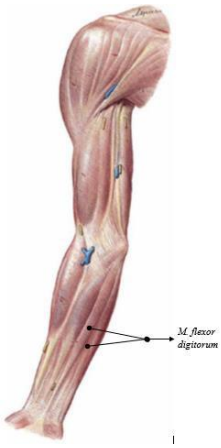
Gambar 1.1 Sinyal *Raw* EMG

Gambar 1.1 menunjukkan sinyal *raw* EMG yang merupakan sinyal biopotensial dari otot. Terdapat dua kondisi dari sinyal otot yaitu kontraksi dan relaksasi. Pada Gambar 1.1 kondisi kontraksi ditunjukkan dengan munculnya pola sinyal dan kondisi relaksasi ditunjukkan pada bagian yang tidak terdapat sinyal atau garis horizontal. Pada sinyal *raw* EMG terdapat nilai positif dan negatif dari sinyal.



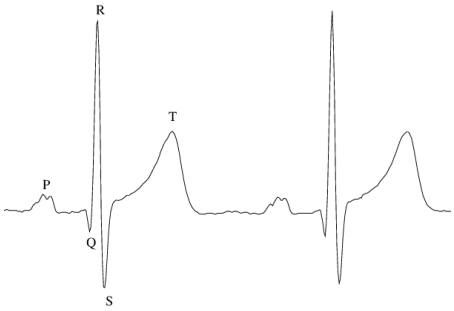
Gambar 1.2 Sinyal *Rectified* EMG

Gambar 1.2 menunjukkan sinyal EMG yang telah dilakukan rektifikasi. Rektifikasi adalah proses mengabsolutkan sinyal dengan mengubah nilai negatif dari sinyal menjadi nilai positif. Proses rektifikasi sering digunakan untuk melakukan pengolahan sinyal EMG lebih lanjut. Pengambilan sinyal EMG memerlukan 3 elektroda dimana dua elektroda ditempel pada *muscle belly* dan satu elektroda pada referensi (*Ground*) atau bagian alat gerak yang berperan menjadi *ground* seperti siku. Salah satu contoh otot yang sering digunakan yaitu *M. flexor digitorum* yang berfungsi untuk melakukan fleksi pada jari tangan.



Gambar 1.3 *M. Flexor digitorum* Tangan Kanan.

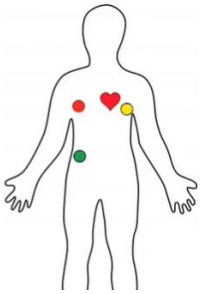
1.2 Electrocardiograph (ECG)



Gambar 1.4 Sinyal *Electrocardiography* (ECG).

Gambar 1.5 menunjukkan pola dari sinyal ECG. Sinyal ECG terbentuk akibat aktivitas listrik yang berasal dari proses depolarisasi dan repolarisasi atrium dan ventrikel jantung. Satu siklus jantung dalam sinyal ECG terdiri dari satu gelombang P-QRS-T.

Salah satu Teknik pengambilan sinyal ECG yaitu dengan menggunakan metode segitiga *Einthoven* yang merupakan tipe ECG *unipolar*. ECG *unipolar* hanya membutuhkan 3 buah elektroda yang dapat dipasangkan sesuai pada gambar 1.5.



Gambar 1.5 Pemasangan Elektroda ECG.

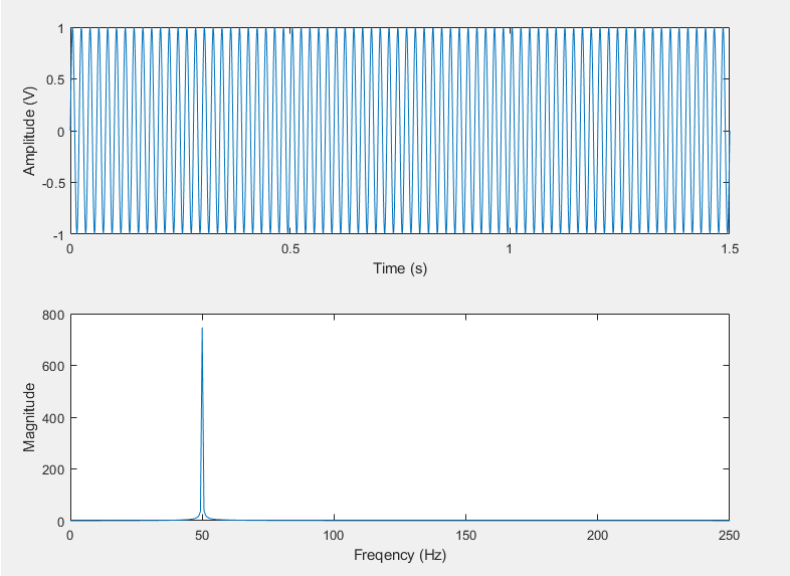
1.3 Spektrum Domain Frekuensi

Pada sinyal periodik, dekomposisi menjadi komponen sinusoidal disebut Deret Fourier. Sedangkan pada sinyal aperiodik (finite energy) disebut sebagai Transformasi Fourier. Transformasi Fourier dan Deret Fourier adalah alat matematis yang sangat penting dalam analisis. Dengan melakukan analisis frekuensi, kita dapat melihat representasi matematis komponen-komponen frekuensi yang terkandung dalam suatu sinyal. Rangkaian frekuensi yang terkandung dalam sinyal tersebut disebut spektrum. Persamaan 1.1 dapat digunakan untuk mengubah sinyal dengan domain waktu x(t) menjadi spektrum sinyal dalam domain frekuensi x(ω).

$$x(\omega) = \int_{-\infty}^{\infty} x(t).\, e^{-j\omega t} dt \quad \text{.....(1.1)}$$

x(t) = sinyal domain waktu.

t = waktu (second).



Gambar 1.6 Sinyal dengan Spektrum Frekuensi 50 Hz.

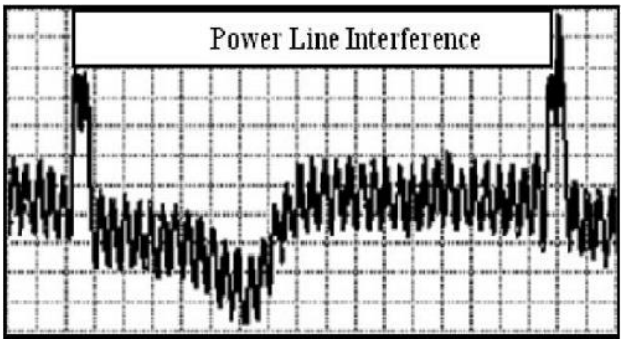
1.4 *Noise*

1.4.1 *Power Line Interference* (PLI)

Power Line Interference (PLI) adalah sumber gangguan sinyal atau *noise* dengan frekuensi 50/60 Hz yang dapat mempengaruhi hasil perekaman atau pengukuran sinyal. Penyebab dari PLI ada beberapa macam antara lain:

1. Gangguan elektromagnetik dari *power supply*.

2. Medan elektromagnetik dari mesin atau rangkaian dengan listrik yang menyebabkan frekuensi harmonik.
3. Efek simpang dari medan arus AC di dalam kabel.
4. *Grounding* yang buruk.
5. Alat elektronik.



Gambar 1.7 Sinyal dengan PLI

1.4.2 *Motion Artifact*

Noise ini terjadi akibat gerakan dari pasien atau subjek yang disebabkan oleh aktivitas otot saat proses perekaman. Hal ini mempengaruhi impedansi antara elektroda dan kulit yang menyebabkan terbentuknya potensial lain pada sinyal. *Noise* ini memiliki frekuensi dengan rentang 0-10 Hz.

1.4.3 *Wandering Baseline*

Noise ini memiliki frekuensi yang rendah yang disebabkan oleh adanya *offset* tegangan pada elektroda, respirasi, dan gerakan tubuh. Pada umumnya *noise* ini memiliki frekuensi diatas 1 Hz, namun adapun juga frekuensi dibawah 1 Hz (sangat kecil). Pada umumnya *noise ini* disebabkan oleh *offset* akibat tegangan DC.

1.5 Filter Digital *Finite Impulse Response* (FIR)

Filter FIR dapat direpresentasikan dengan persamaan berikut:

$$y[n] = \sum_{k=0}^{N-1} h[k].x[n-k] \quad (1.1)$$

Respon impuls dari filter FIR adalah koefisien filter h[k] atau b[k]. FIR tidak memiliki koefisien a kecuali a[0], sehingga tidak memiliki transfer fungsi dan penerapannya dilakukan secara konvolusi. Untuk mendesain filter FIR, dimulai dengan spektrum frekuensi yang diinginkan. Frekuensi yang diinginkan sendiri merupakan transformasi fourier dari koefisien b, sehingga untuk mendapatkan koefisien b dilakukan invers transformasi fourier dari spektrum. Bentuk kontinu

digunakan untuk mendapatkan kontinu *series* dari koefisien b(t) dan mengubahnya dalam bentuk disktit b(k).

$$b(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} B(\omega).e^{-j\omega t}d\omega = \frac{1}{2\pi}\int_{-\omega c}^{\omega c} 1.e^{-j\omega t}d\omega$$

dengan fungsi window yaitu 1.0 antara ωc dalam bentuk negatif dan positif. Maka dapat ditulis:

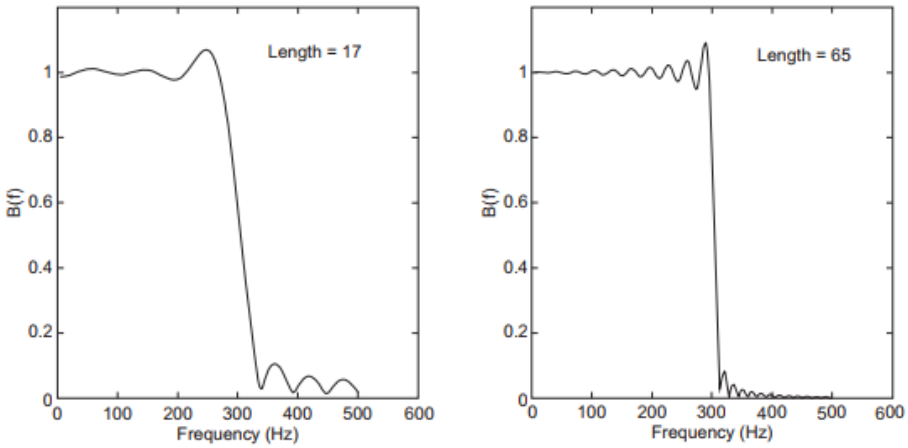
$$b(t) = \frac{1}{2\pi}\frac{e^{j\omega t}}{jt}\Big|_{-\omega c}^{\omega c} = \frac{1}{\pi t}\frac{e^{j\omega ct}-e^{-j\omega ct}}{2j} = \frac{\sin(\omega c.t)}{\pi t} \quad (1.2)$$

$$b(t) = \frac{\sin(2.\pi.fc.t)}{\pi t} \quad (1.3)$$

Respon impuls dari *rectangular window* memiliki bentuk umum dengan fungsi *sinc(x)*: *sin(x)/x*. Koefisien filter b[k] dapat diperoleh menggunakan persamaan 1.3 dengan mengubah dari variabel kontinu waktu (t), menjadi variabel diskrit (k). Filter FIR memiliki penjang filter yang simetris (memiliki sisi positif dan negatif sehingga berbentuk:

$$b(k) = h[k] = \frac{\sin\left(2.\pi.fc.\left(k-\frac{L}{2}\right)\right)}{\pi\left(k-\frac{L}{2}\right)} \quad (1.4)$$

fc adalah frekuensi *cut-off* dan L adalah panjang filter. Pada persamaan 1.3 atau 1.4, koefisien filter FIR masih bernilai *infinite* atau tidak terbatas. Sehingga untuk membatasi koefisien filter diperlukan pemotongan atau pembatasan nilai koefisien. Dampak yang ditimbulkan dengan adanya pembatasan nilai fungsi koefisien filter yaitu munculnya *Gibbs artifact* yang merupakan osilasi dari filter FIR yang dapat terlihat pada spektrum *magnitude* seperti Gambar 1.8.



Gambar 1.8 Spektrum *Magnitude* dari Dua Filter FIR dengan Pembatasan Koefisien, (a) 17 dan (b) 65.

Metode *windowing* digunakan untuk meminimalisir dampak dari pembatasan koefisien. Beberapa jenis fungsi *window* yang sering digunakan yaitu metode *hamming* dan *blackman*.

$$w[n]_{hamming} = 0.5 - 0.46\cos\left(\frac{2\pi n}{N}\right) \quad (1.5)$$

$$w[n]_{blackman} = 0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right) + 0.08\cos\left(\frac{4\pi n}{N}\right) \quad (1.6)$$

Sehingga koefisien Filter dengan nilai yang terbatas yaitu:

$$h[n] = h_d[n].w[n] \quad (1.7)$$

Beberapa respon impuls dari filter FIR berdasarkan jenis filter yang digunakan yaitu:

$$lowpass = h_d[n] = \begin{cases} \dfrac{\sin\left(2\pi.fc.\left(k-\frac{L}{2}\right)\right)}{\pi\left(k-\frac{L}{2}\right)}; k \neq 0 \\ \dfrac{2\pi.fc}{\pi}; k = 0 \end{cases}$$

$$highpass = h_d[n] = \begin{cases} -\dfrac{\sin\left(2\pi.fc.\left(k-\frac{L}{2}\right)\right)}{\pi\left(k-\frac{L}{2}\right)}; k \neq 0 \\ 1-\dfrac{2\pi.fc}{\pi}; k = 0 \end{cases}$$

$$bandpass = h_d[n] = \begin{cases} \dfrac{\sin\left(2\pi.fc2.\left(k-\frac{L}{2}\right)\right)}{\pi\left(k-\frac{L}{2}\right)} - \dfrac{\sin\left(2\pi.fc1.\left(k-\frac{L}{2}\right)\right)}{\pi\left(k-\frac{L}{2}\right)}; k \neq 0 \\ \dfrac{2\pi.fc2-2\pi.fc1}{\pi}; k = 0 \end{cases}$$

$$bandstop = h_d[n] = \begin{cases} \dfrac{\sin\left(2\pi.fc1.\left(k-\frac{L}{2}\right)\right)}{\pi\left(k-\frac{L}{2}\right)} - \dfrac{\sin\left(2\pi.fc2.\left(k-\frac{L}{2}\right)\right)}{\pi\left(k-\frac{L}{2}\right)}; k \neq 0 \\ 1-\dfrac{2\pi.fc2-2\pi.fc1}{\pi}; k = 0 \end{cases}$$

1.6 Filter Digital *Infinite Impulse Response* (IIR)

Filter FIR dapat direpresentasikan dengan persamaan berikut:

$$y[n] = \sum_{k=0}^{N-1} h[k].x[n-k] = \sum_{k=0}^{N-1} b_k.x[n-k] - \sum_{k=1}^{M} a_k.y[n-k] \quad (1.8)$$

$h[k]$ adalah respon impuls filter yang mempunyai panjang *infinite* atau tak hingga, b_k dan a_k adalah koefisien filter, $x[n]$ adalah masukan filter, dan $y[n]$ adalah keluaran filter. Dalam domain z, fungsi alih filter IIR dapat dituliskan sebagai:

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1+\sum_{k=1}^{M} a_k z^{-k}} \quad (1.9)$$

dapat difaktorkan menjadi:

$$H(z) = \frac{K(z-z_1)(z-z_2)\ldots(z-z_N)}{(z-p_1)(z-p_2)\ldots(z-p_M)} \quad (1.10)$$

dengan z_1, z_2,…, z_N adalah *zero* dan p_1, p_2,…, p_M adalah *pole H(z)*. Nilai-nilai *pole* haris berada dalam lingkaran dengan jari-jari 1 agar membentuk filter yang stabil, sehingga terdapat kemungkinan filter IIR memiliki ketidakstabilan dalam kinerjanya. Filter IIR juga memiliki *feedback* dari keluaran seperti pada persamaan 1.8 sehingga memiliki jumlah koefisien yang lebih sedikit dibandingkan dengan filter FIR. Salah satu metode yang dapat digunakan untuk menentukan koefisien dari filter IIR ini yaitu dengan metode *Bilinear Z-Transform* (BZT). Metode ini memanfaatkan fungsi transfer dari sistem filter IIR *lowpass* yang dilakukan denormalisasi dengan frekuensi *prewarp* $\omega'_p = k\tan\left(\frac{\omega_p}{2}\right)$ sesuai dengan jenis filter.

$$lowpass - lowpass = s = \frac{s}{\omega'_p} \quad (1.11)$$

$$lowpass - highpass = s = \frac{\omega'_p}{s} \quad (1.12)$$

$$lowpass - bandpass = s = \frac{s^2+{\omega_0}^2}{Ws} \quad (1.13)$$

$$lowpass - bandpass = s = \frac{Ws}{s^2+{\omega_0}^2} \quad (1.14)$$

dengan ${\omega_0}^2 = {\omega'_{p1}}^2.{\omega'_{p2}}^2$ dan $W = {\omega'_{p2}}^2 - {\omega'_{p1}}^2$. Setelah dilakukan denormalisasi dilanjutkan dengan transformasi z dengan mengubah $s = \frac{z-1}{z+1}$, sehingga diperoleh fungsi alih digital *H(z)*. Melalui *H(z)* dapat diperoleh persamaan differens untuk mendapatkan koefisien-koefisien dari filter IIR seperti pada contoh berikut:

$$H(s) = \frac{0{,}03952}{s^2+0{,}2811\,s+0{,}03952} \quad (1.15)$$

$$H(z) = \frac{0{,}03952}{\left(\frac{z-1}{z+1}\right)^2 + 0{,}2811\left(\frac{z-1}{z+1}\right) + 0{,}03952} x \frac{z^{-2}}{z^{-2}} \quad (1.16)$$

$$\frac{y(n)}{x(n)} = \frac{0{,}03 + 0{,}06\, z^{-1} + 0{,}03 z^{-2}}{1 - 1{,}45 z^{-1} + 0{,}57 z^{-2}} \quad (1.17)$$

$$y(n) = 1{,}45\, y(n-1) - 0{,}57\, y(n-2) + 0{,}03\, x(n) + 0{,}06\, x(n-1) + 0{,}03\, x(n-2) \quad (1.18)$$

Persamaan 1.18 adalah bentuk dari persamaan differens IIR.

ALAT DAN BAHAN:

1. Teensy 3.5
2. AD8232
3. Myoware 3.0
4. Breadboard
5. Kabel Jumper (Male-male), (Male-Female)
6. Electroda Ag-Cl
7. *Software* Arduino Uno (Install Teensyduino)
8. *Software* Tera Term
9. *Software* MATLAB

PERCOBAAN 1: Pengambilan Data Sinyal ECG dan EMG

Langkah-langkah:

1. Lakukan pengaturan pada Myoware yaitu:
    a. Hubungkan bagian + dari Myoware dengan 3,3 V dari Teensy.
    b. Hubungkan bagian - dari Myoware dengan GND dari Teensy.
    c. Hubungkan bagian SIG dari Myoware dengan A0 dari Teensy.
2. Lakukan pemasangan Myoware pada otot *M. Flexor digitorum* (Gambar 1.3) dengan bagian *ground*.
3. Hubungkan Teensy dengan *Personal Computer* (PC) menggunakan kabel USB.
4. Buka *software* Arduino IDE dan pada menu *tools* lakukan pengaturan sebagai berikut:
    a. Pilih Board Teensy 3.5
    b. Pilih Port Teensy 3.5
5. Lakukan pemrograman menggunakan Arduino IDE sebagai berikut:

a. Aktifkan komunikasi serial dari Teensy dengan *baudrate* 9600.
b. Lakukan pembacaan analog dari sinyal EMG melalui pin A0.
c. Lakukan konversi nilai data dari data digital menjadi nilai tegangan 0-5V.
d. Tampilkan bentuk sinyal menggunakan *serial plotter* Arduino IDE.
e. Tampilkan waktu dan nilai dari sinyal ECG pada *serial monitor*.

6. Catat jumlah data yang muncul selama 1 detik (*sampling frequency*).
7. Tutup jendela *serial monitor* dan *serial plotter* kemudian buka *software* Tera Term.
8. Pada *software* Tera Term lakukan:
    a. Pilih serial dan klik OK.
    b. Pada menu file pilih Log dan pilih direktori folder untuk menyimpan data sinyal EMG
    c. Beri nama file dengan format .*csv*
9. Lakukan kontraksi dan relaksasi otot secukupnya.
10. Tutup *software* Tera Term.
11. Buka *software* Matlab dan lakukan pemrograman sebagai berikut:
    a. Panggil *file.csv* dari EMG dan ECG yang telah tersimpan.
    b. Lakukan *plotting* grafik dengan sumbu x adalah waktu (*second*) dan sumbu y adalah nilai sinyal.

Lakukan pengambilan data sinyal ECG dengan melakukan pengaturan pada AD8232 yaitu:

a. Hubungkan bagian *Power* dari AD8232 dengan 3,3 V dan GND dari Teensy.
b. Hubungkan bagian OUTPUT dari AD8232 dengan A0 dari Teensy.

Lakukan pemasangan elektoda seperti pada Gambar 1.5 dan lakukan langkah 5 hingga langkah 11.

PERCOBAAN 2: Penambahan *Noise* Pada Sinyal ECG dan EMG dan Penerapan *Fast Fourier Transform* (FFT)

Langkah-langkah:

1. Buka *software* Matlab dan panggil *file.csv* dari sinyal ECG dan EMG.
2. Lakukan *plotting* data dalam domain waktu dan domain frekuensi.
3. Buatlah sinyal sinusoidal dengan frekuensi 50 Hz (PLI) dengan amplitudo 1 V
4. Tambahkan data nilai sinyal ECG dan EMG dengan sinyal sinusoidal 50Hz untuk menambahkan *power line interference.* (**Pastikan bentuk dimensi antara data sinyal EMG dan ECG sama dengan sinyal *noise***).
5. Lakukan *plotting* dalam domain waktu dan domain frekuensi dan bandingkan hasilnya dengan *plotting* pada langkah 2.
    a. Atur nilai frekuensi (bins) pada sumbu x sehingga dimulai dari frekuensi 0.
    b. Lakukan konversi dari frekuensi(bins) menjadi frekuensi(Hz) dengan mengkalikan frekuensi(bins) pada langkah 5a dengan resolusi frekuensi.

$$Resolusi\ Frekuensi = \frac{Frekuensi\ sampling}{Banyak\ data}$$

    (*Pada MATLAB dapat digunakan fungsi **abs()** dan **fft()** untuk melakukan plotting dalam domain frekuensi*)
6. Simpan Data Nilai Sinyal ECG dan EMG yang telah ditambahkan *noise* PLI dalam *file_noise.csv*

Lakukan penambahan *noise motion artifact* dan *wandering baseline* pada sinyal EMG dan ECG yang bersih dan lakukan langkah 5 dan 6 untuk masing-masing *noise* dengan melakukan hal berikut (**Pastikan bentuk dimensi antara data sinyal EMG dan ECG sama dengan sinyal *noise***):

a. *Motion artifact noise*: Buatlah sinyal sinusoidal dengan amplitude 1V dan frekuensi dibawah 1 Hz dengan nilai frekuensi acak.
   (*Pada MATLAB dapat digunakan fungsi rand(1,1) untuk membentuk nilai*

antara 0 hingga 1 secara acak)

b. *Wandering baseline*: Buatlah satu variabel yang memuat informasi tegangan DC 1,5 V.

PERCOBAAN 3: Penerapan Filter FIR dan IIR

Finite Impulse Response

Langkah-langkah:

1. Buka *software* MATLAB dan panggil *file_noise.csv*.
2. Masukkan inisialisasi frekuensi sampel (*fs*), panjang data, waktu, frekuensi *cut-off* atau frekuensi *high* dan *low* sesuai dengan jenis filter.
   (*Filter Lowpass dan Highpass hanya menggunakan fc, sedangkan Bandpass dan Bandstop menggunakan fh dan fl*)
3. Masukkan panjang filter (jumlah koefisien filter)
4. Lakukan normalisasi pada *fc* atau *fl* dan *fh* dengan melakukan pembagian dengan frekuensi sampel (*fs*).
5. Lakukan penghitungan $h_d[n]$ sesuai dengan jenis filter yang digunakan sebanyak panjang filter **(Penghitungan harus simetrik, jika panjang filter sebanyak L, maka harus memuat $\boldsymbol{h_d[-L/2]}$ $\boldsymbol{hingga}$ $\boldsymbol{h_d[L/2]}$)**.
6. Lakukan penghitungan fungsi *window* (*w(n)*).
7. Kalikan $h_d[n]$ dengan $w[n]$.
   (*Pada MATLAB dapat digunakan perkalian vektor dengan operasi* **($\boldsymbol{h_d[n]}$.* $\boldsymbol{w[n]}$)**).
8. Lakukan proses filter sinyal dengan melakukan konvolusi antara sinyal *input* dengan koefisien filter.
   (*Pada MATLAB dapat digunakan fungsi* (**conv(*input, koef_filter,* 'same')**) untuk melakukan konvolusi).
9. Lakukan plotting sinyal sebelum dan sesudah filter dalam domain waktu dan frekuensi.

Infinite Impulse Response

Langkah-langkah:

1. Buka *software* MATLAB dan panggil *file_noise.csv*.
2. Masukkan inisialisasi frekuensi sampel (*fs*), panjang data, waktu, orde filter, dan periode sampel (*Ts*).
3. Lakukan desain filter IIR sesuai dengan jenis filter yang diperlukan.
   (*Pada MATLAB dapat digunakan toolbox* (**designfilt()**))
   Diperlukan informasi berikut sesuai dengan jenis filter yang digunakan.
   a. Lowpass = nama filter (contoh: ‘lowpassiir’), orde filter, frekuensi *passband*, *ripple passband*, dan frekuensi sampel (*fs*).
      (*designfilt('lowpassiir','FilterOrder',8,'PassbandFrequency',35e3,'PassbandRipple',0.2, 'SampleRate',200e3);*)
   b. Highpass = nama filter (contoh: ‘highpassiir’), orde filter, frekuensi *passband*, *ripple passband*, dan frekuensi sampel (*fs*).
      (*designfilt('highpassiir','FilterOrder',8,'PassbandFrequency',35e3,'PassbandRipple',0.2, 'SampleRate',200e3);*).
   c. Bandpass = nama filter (contoh: ‘bandpassiir’), orde filter, frekuensi 1, frekuensi 2, dan frekuensi sampel (*fs*).
      (*designfilt('bandpassiir','FilterOrder',20,'HalfPowerFrequency1',500,'HalfPowerFrequency2',560, 'SampleRate',1500);*)
   d. Bandstop = nama filter (contoh: ‘bandstopiir’), orde filter, frekuensi 1, frekuensi 2, dan frekuensi sampel (*fs*).
      (*designfilt('bandstopiir','FilterOrder',20,'HalfPowerFrequency1',500,'HalfPowerFrequency2',560, 'SampleRate',1500);*)
4. Lakukan proses filter.
   (*Pada MATLAB dapat digunakan toolbox* (**filter(desainfilter, sinyal_input)**))
5. Lakukan *plotting* sinyal sebelum dan sesudah filter.