

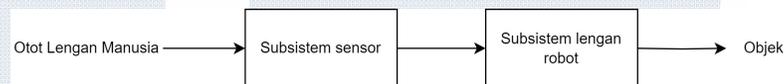
BAB III

PERANCANGAN DAN IMPLEMENTASI SISTEM

3.1 Tinjauan Desain Sistem

3.1.1 Desain Sistem Keseluruhan

Desain sistem secara keseluruhan produk *EMG Controlled Assistive Robotic Arm* memiliki dua subsistem, yakni subsistem sensor (*elbow sleeve*) dan lengan robot seperti yang ditunjukkan oleh *Data Flow Diagram (DFD)* pada Gambar 3.1.



Gambar 3.1 DFD *EMG Controlled Assistive Robotic Arm*

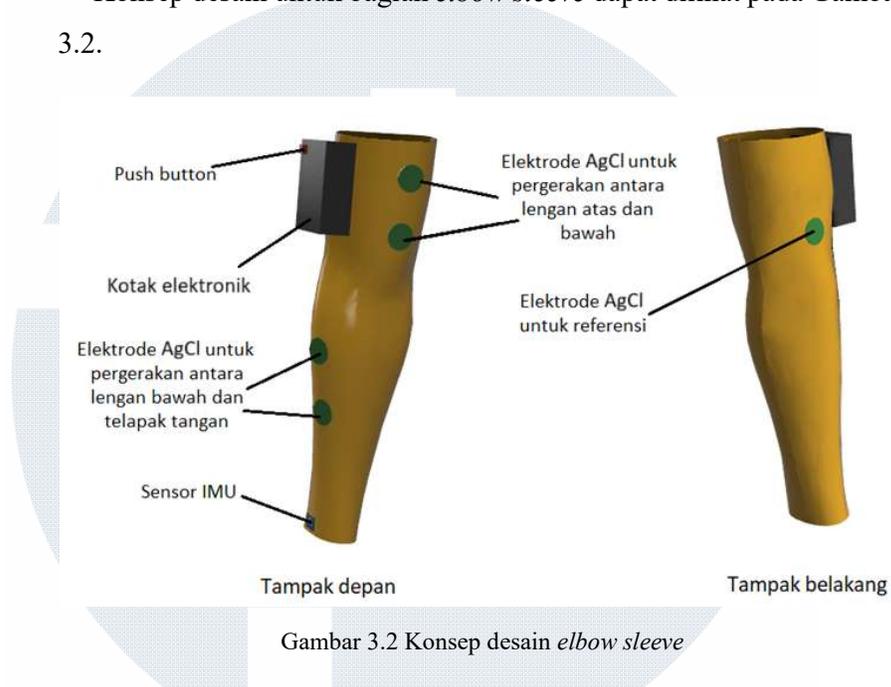
Tabel 3.1 Penjelasan Input/Output Sistem Keseluruhan

Parameter	Keterangan
Input	<ul style="list-style-type: none">● Pergerakan lengan manusia● Konfigurasi awal pengguna
Output	<ul style="list-style-type: none">● Sistem bergerak meniru pergerakan lengan manusia● Sistem mengambil atau meletakkan objek
Fungsi	<ul style="list-style-type: none">● Melakukan pergerakan untuk mengambil dan meletakkan objek

Berdasarkan Gambar 3.1, *EMG Controlled Assistive Robotic Arm* menerima input dari lengan manusia dan diproses oleh subsistem *elbow sleeve*. Pada subsistem *elbow sleeve*, pergerakan lengan pengguna dapat dibaca oleh sensor yang ada. Data tersebut kemudian diproses dan dikirimkan pada subsistem lengan robot dengan menggunakan komunikasi nirkabel, yaitu *bluetooth*. Setelah subsistem lengan robot menerima data yang dikirimkan oleh subsistem *elbow sleeve*, subsistem lengan robot mengolah data yang diterima menjadi pergerakan motor *servo* pada lengan robot menyesuaikan dengan gerak lengan pengguna untuk melakukan sesuatu terhadap objek yang dikehendaki.

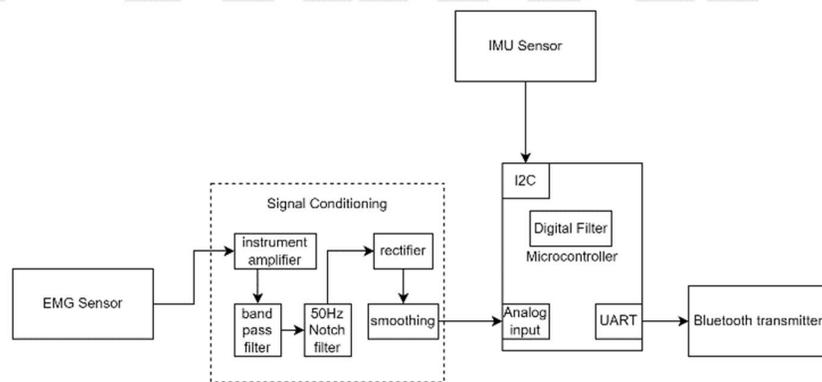
3.1.2 Desain Subsistem Elbow Sleeve

Konsep desain untuk bagian *elbow sleeve* dapat dilihat pada Gambar 3.2.



Subsistem sensor (*elbow sleeve*) terdiri dari *elbow sleeve* dan kotak elektronik. *Elbow sleeve* digunakan sebagai tempat untuk menempatkan sensor EMG sehingga sensor EMG dapat menempel pada kulit. Kotak elektronik digunakan untuk menyimpan baterai, *push button*, mikrokontroler, dan rangkaian pada subsistem pengkondisi sinyal.

Hubungan antar komponen yang digunakan pada subsistem *elbow sleeve* dapat dilihat pada Gambar 3.3.



Gambar 3.3 DFD subsistem elbow sleeve

Tabel 3.2 Penjelasan Input/Output Subsistem Elbow Sleeve

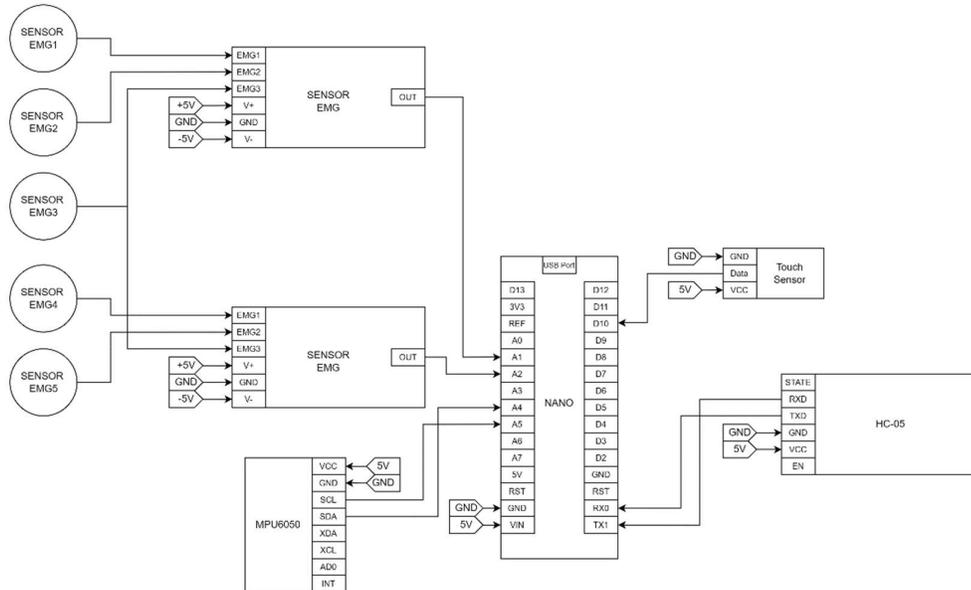
Parameter	Keterangan
Input	<ul style="list-style-type: none"> • Sinyal sensor EMG • Sensor IMU
Output	<ul style="list-style-type: none"> • Data dari sensor EMG dan IMU yang sudah difilter
Fungsi	<ul style="list-style-type: none"> • Mendeteksi pergerakan lengan manusia melalui sensor EMG dan IMU • Menyaring data hasil bacaan sensor EMG dan IMU • Mengirimkan data hasil filter melalui modul <i>transmitter bluetooth</i>

Pada subsistem sensor terdapat sensor EMG yang mendeteksi pergerakan lengan berdasarkan beda potensial yang dihasilkan antara 2 otot lengan dan sensor IMU yang mendeteksi pergerakan lengan berdasarkan mekanisme *Micro-Electromechanical System* (MEMS). Data yang dihasilkan oleh sensor EMG kemudian diproses secara analog di dalam subsistem pengkondisian sinyal sebelum diterima oleh mikrokontroler Arduino Nano, sedangkan data yang dihasilkan oleh sensor IMU langsung dikirimkan ke Arduino Nano melalui protokol *Inter-Integrated Circuit* (I2C), kemudian diproses secara digital. Setelah kedua data telah diterima dan difilter, Arduino Nano akan mengirimkan data tersebut ke *transmitter bluetooth* melalui protokol *Universal Asynchronous Receiver/Transmitter* (UART) untuk dikirimkan ke *receiver bluetooth* yang terdapat pada subsistem lengan robot.

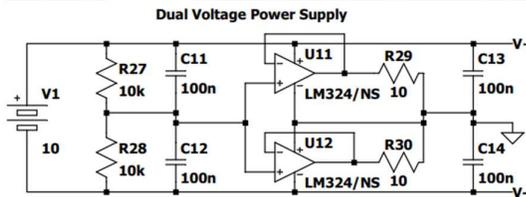
U M I

3.1.3 Diagram Subsistem Lengan Robot

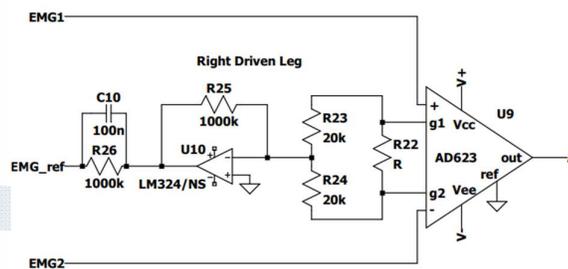
Diagram subsistem elbow sleeve terdapat pada Gambar 3.4.



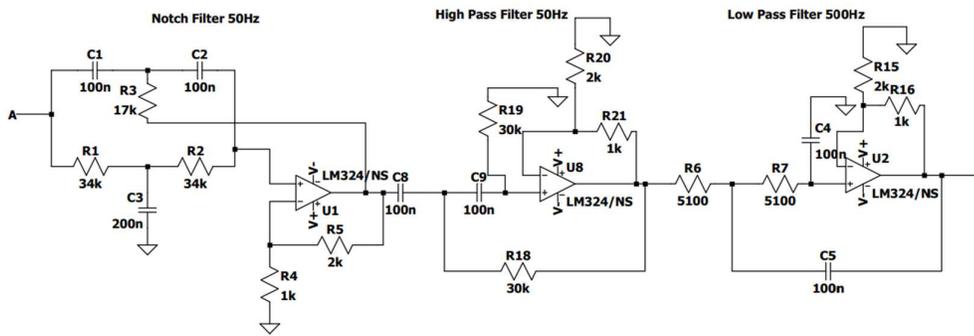
Gambar 3.4 *Wiring diagram elbow sleeve*



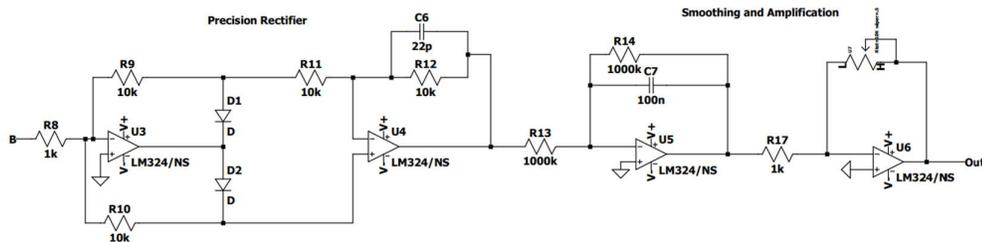
Gambar 3.5 *Circuit diagram rangkaian dual voltage power supply*



Gambar 3.6 *Circuit diagram rangkaian right driven leg dan instrument amplifier*



Gambar 3.7 Circuit diagram rangkaian filter



Gambar 3.8 Circuit diagram rangkaian precision rectifier, smoothing, and amplifier

Berdasarkan *wiring diagram* pada Gambar 3.4, pergerakan tangan akan dibaca oleh sensor EMG dan sensor MPU6050. Hasil pembacaan sensor EMG akan diolah terlebih dahulu pada rangkaian pengkondisi sinyal pada Gambar 3.5 untuk menghilangkan *noise* pada data EMG, sedangkan pembacaan sensor MPU6050 akan diolah di dalam Arduino Nano. Hasil pembacaan sensor kemudian dikirim ke subsistem lengan robot dengan menggunakan modul *bluetooth*.

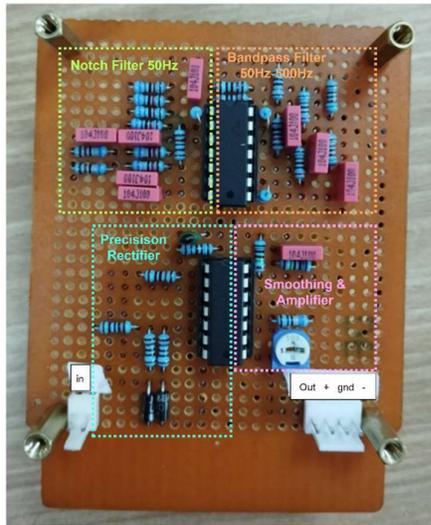
3.2 Implementasi Sistem

Pengembangan subsistem elbow sleeve dilakukan dengan menggunakan aplikasi LTspice VII, Tina-Ti, dan Arduino IDE. Aplikasi LTspice VII dan Tina-Ti digunakan untuk melakukan simulasi rangkaian pengolahan sinyal dari sensor EMG. Hasil yang didapatkan dari kedua aplikasi ini adalah grafik frekuensi respon dan grafik osiloskop sinyal keluaran dari rangkaian. Hasil grafik tersebut akan dibandingkan dengan pengukuran grafik yang terbaca melalui osiloskop pada rangkaian fisik. Kemudian, hasil keluaran rangkaian pengolahan sinyal EMG ini akan dibaca oleh mikrokontroler Arduino Nano

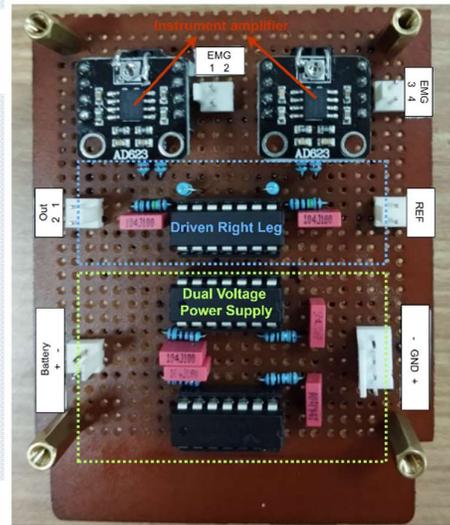
dengan Aplikasi Arduino IDE. Arduino IDE ini menggunakan bahasa pemrograman C++. Selain digunakan untuk membaca sensor EMG, Arduino Nano juga digunakan untuk mengolah pembacaan sensor MPU6050.

3.2.1 Hasil Implementasi Rangkaian Filter

Secara fisik, bentuk rangkaian filter yang sudah jadi dapat dilihat pada Gambar 3.9 dan 3.10.



Gambar 3.9a PCB rangkaian *filter*, *rectifier*, dan *smoothing*



Gambar 3.9b PCB rangkaian *instrument amplifier*, *right driven leg* dan *dual voltage power supply*

Sebelum mengimplementasi rangkaian *elbow sleeve* seperti pada Gambar 3.9a dan Gambar 3.9b, dilakukan simulasi terlebih dahulu untuk rangkaian sensor EMG. Perancangan rangkaian sensor EMG, dimulai dengan menambahkan *bandpass filter* untuk memblokir frekuensi di bawah 50 Hz dan di atas 500 Hz [19]. Pada awalnya digunakan *sallen key bandpass filter* yang memiliki *cut off frequency* pada 50 Hz dan 500 Hz. Frekuensi 50 Hz diberi filter akibat munculnya noise dari listrik PLN berdasarkan pengujian sensor EMG pada osiloskop sebelumnya. Filter yang digunakan merupakan *filter butterworth* dengan nilai *quality factor* (Q) bernilai 0,707. Kapasitor yang digunakan adalah tipe MKM dengan nilai 100 nF. Sehingga untuk

mencari nilai resistor pada *high pass filter* dan *low pass filter* digunakan Persamaan 3.1.

$$f_c = \frac{1}{2\pi RC} \quad (3.1)$$

Sehingga didapatkan nilai resistor 31,83 k Ω untuk *high pass filter* dan 3,183 k Ω untuk *low pass filter*. Karena ingin didapatkan nilai Q = 0.707, maka dibutuhkan penguatan sebesar 1,585 kali seperti pada Persamaan 3.2.

$$Q = \frac{1}{3 - A} \quad (3.2)$$

$$A = \frac{3Q - 1}{Q} = \frac{1,121}{0,707} = 1,585$$

Maka dengan dimisalkan salah satu resistor bernilai 2 k Ω , didapatkan resistor kedua dengan nilai 1170 Ohm seperti pada Persamaan 3.3.

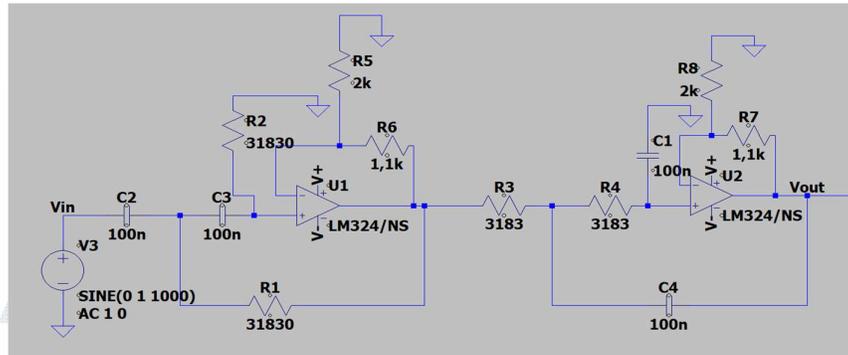
$$A = \frac{V_{out}}{V_{in}} = 1 + \frac{R_1}{2000} \quad (3.3)$$

$$1 + \frac{R_1}{2000} = 1,585$$

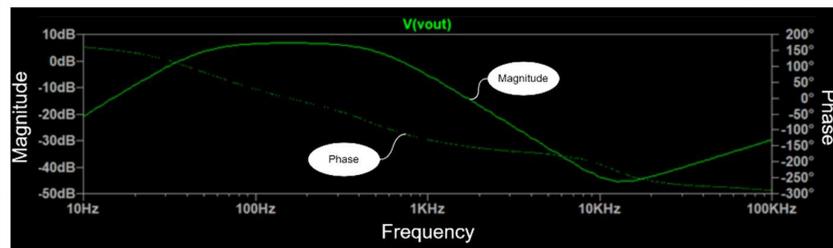
$$\frac{R_1}{2000} = 0,585$$

$$R_1 = 1170 \text{ Ohm}$$

Nilai resistor tersebut akan diubah menjadi 1,1 k Ω untuk menyesuaikan dengan nilai resistor yang ada di pasaran. Sehingga didapatkan rangkaian *bandpass filter* seperti pada Gambar 3.10. Grafik frekuensi respon rangkaian *bandpass filter* dapat dilihat pada Gambar 3.11.



Gambar 3.10 Rangkaian *Bandpass Filter* Subsystem *Elbow Sleeve*



Gambar 3.11 Grafik Frekuensi Respon *Bandpass Filter* Subsystem *Elbow Sleeve*. Garis lurus hijau menunjukkan *voltage gain* sinyal dalam satuan “dB” dan garis putus-putus hijau menunjukkan pergeseran fasa sinyal dalam satuan “°”

Nilai -3 dB pada grafik tersebut berada pada frekuensi 30 Hz pada *high pass filter* 50 Hz dan 865 Hz pada *low pass filter* 500 Hz. Namun setelah rangkaian tersebut diuji pada simulasi osiloskop, bentuk grafik tegangan yang terbaca tidak berubah nilai amplitudonya ketika nilai frekuensi pada *function generator* diubah. Seperti yang ditunjukkan pada Gambar 3.12.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

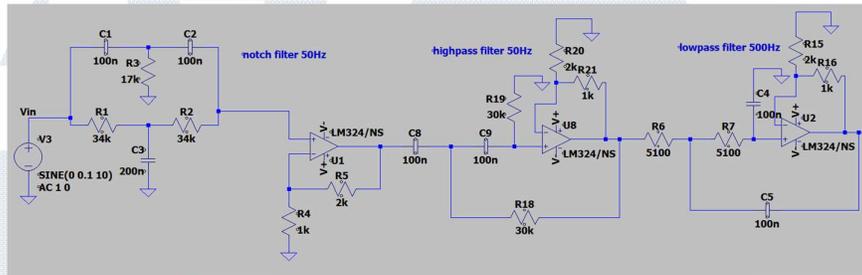


Gambar 3.12 Grafik Sinyal Output *Band pass filter* Subsistem *Elbow Sleeve* Pada Frekuensi 7 Hz dengan resolusi waktu 50ms/div, resolusi tegangan sinyal input (biru) sebesar 500mV/div dan sinyal output (kuning) sebesar 2V/div

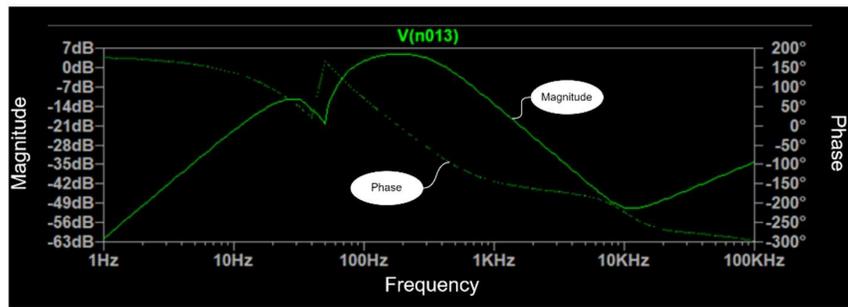
Berdasarkan Gambar 3.12, dapat dilihat bahwa input sinyal rangkaian *band pass filter* (biru) memiliki amplitudo 50 mV dengan frekuensi 7 Hz. Berdasarkan perhitungan frekuensi *cut off* yang dilakukan, frekuensi 7 Hz tidak dapat dilewatkan sehingga *output* yang diharapkan adalah sinyal keluaran rangkaian (kuning) memiliki amplitudo kurang dari 50 mV. Namun, hasil yang tampak pada Gambar 3.12 adalah sinyal output mengalami *clipping* yang berarti sinyal output memiliki amplitudo sinyal yang jauh lebih besar dari sinyal input.

Hal ini mungkin disebabkan oleh adanya sinyal *noise* pada frekuensi rendah sehingga tegangan yang terbaca pada osiloskop tetap terjadi *clipping*. Oleh karena itu, dibuat rangkaian baru dengan menambahkan *bandstop twin-t active filter* pada frekuensi 50 Hz. Pada *low pass sallen key filter* sebelumnya dengan frekuensi *cut off* 500 Hz, didapatkan nilai -3 dB pada frekuensi 1,25 kHz. Maka dengan mengubah nilai resistor menjadi 5,1 k Ω didapatkan nilai -3dB pada frekuensi 775 Hz. Dengan menggunakan persamaan 1 untuk menghitung nilai resistor pada *twin-t active filter*, didapatkan nilai resistor yang sama, yakni 31830 Ohm. Nilai resistor tersebut diubah menjadi 34 k Ω untuk menyesuaikan dengan ketersediaan komponen. Selain nilai 34 k Ω , diperlukan satu nilai resistor lagi untuk *twin-t active filter* bernilai setengahnya, yakni dipasangkan 17 k Ω (menyesuaikan

nilai yang ada pada pasaran). Hasil rangkaian filter yang telah ditambahkan filter *bandstop* dapat dilihat pada Gambar 3.13. Hasil grafik frekuensi respon rangkaian filter yang telah ditambahkan filter *bandstop* dapat dilihat pada Gambar 3.14.



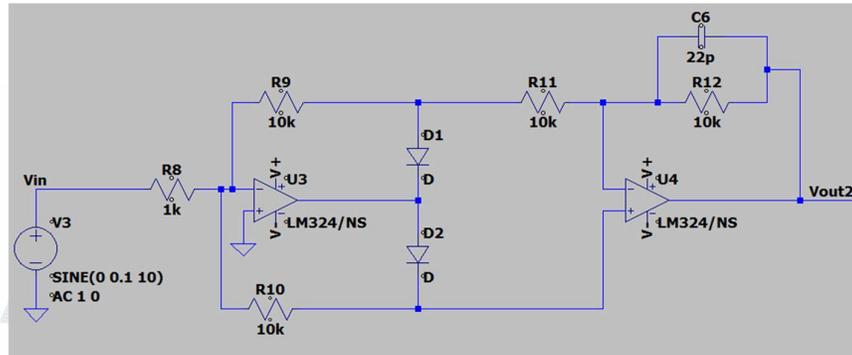
Gambar 3.13 Rangkaian Twin-t Active Filter dan Bandpass Filter Subsystem Elbow Sleeve



Gambar 3.14 Grafik Frekuensi Respon Rangkaian *Twin-t Active Filter* dan *Bandpass Filter* Subsystem *Elbow Sleeve*. Garis lurus hijau menunjukkan *voltage gain* sinyal dalam satuan “dB” dan garis putus-putus hijau menunjukkan pergeseran fasa sinyal dalam satuan “°”

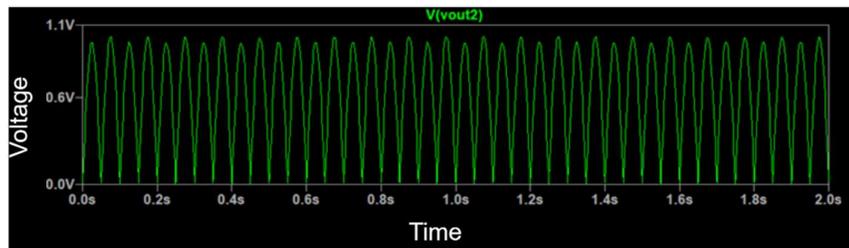
Nilai -3dB pada grafik tersebut berada pada frekuensi 535 Hz untuk *low pass filter*, serta 60 Hz untuk *high pass filter*. *Gain* yang digunakan pada *bandstop filter* bernilai 3 kali dengan resistor 1 k Ω dan 2 k Ω didapatkan dari hasil *fine tuning* agar rentang frekuensi nilai -3dB tidak terlalu jauh.

Selanjutnya pada rangkaian sensor EMG, terdapat rangkaian *precision rectifier* yang berfungsi untuk menyearahkan sinyal yang bernilai negatif. Selain itu, *precision rectifier* digunakan agar gain tidak berkurang, melainkan bertambah. Rangkaian tersebut dapat dilihat pada Gambar 3.15.



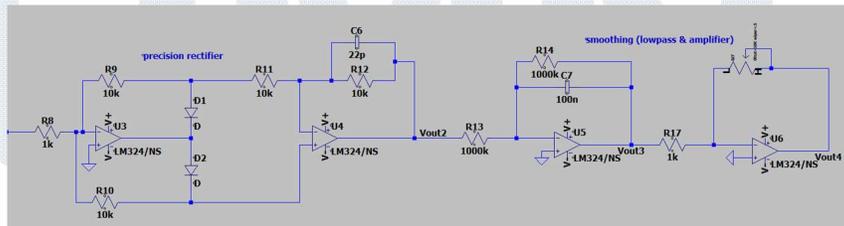
Gambar 3.15 Rangkaian *Precision Rectifier* Subsystem *Elbow Sleeve*

Berdasarkan rangkaian tersebut, didapatkan nilai gain sebesar 10 kali seperti Gambar 3.17. Amplitudo pada Gambar 3.16 memiliki nilai mendekati 1 V dengan input 0,1 V.



Gambar 3.16 Grafik Sinyal Rangkaian *Precision Rectifier* Subsystem *Elbow Sleeve*

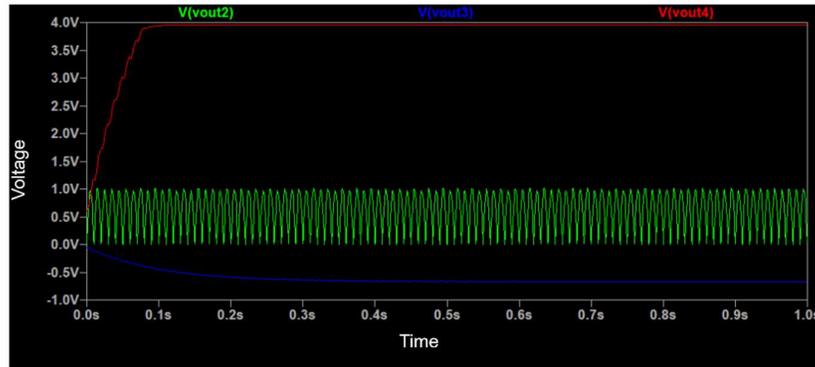
Setelah sinyal disearahkan, dilakukan *smoothing* dan *amplifier* sinyal dengan menggunakan *lowpass filter* dan penguat *inverting* seperti pada Gambar 3.17.



Gambar 3.17 Rangkaian *Precision Rectifier* Beserta *Smoothing* dan *Amplifikasi* Akhir Subsystem *Elbow Sleeve*

Dari rangkaian tersebut didapatkan grafik seperti pada Gambar 3.18. Dimana *Vout2* (hijau) merupakan hasil keluaran dari rangkaian *rectifier*, *Vout3* (biru) merupakan keluaran dari rangkaian *lowpass filter* dan *Vout4* (merah) merupakan keluaran dari rangkaian *smoothing* dan

amplifier. *Low pass filter* di sini berfungsi untuk membuat nilai sinyal menjadi DC (mendekati 0 Hz).



Gambar 3.18 Grafik Sinyal Rangkaian *Precision Rectifier* Beserta *Smoothing* dan Amplifikasi Subsistem *Elbow Sleeve*

Berdasarkan grafik pada Gambar 3.18, terdapat nilai *offset* negatif yang ditunjukkan oleh garis berwarna biru. *Offset* ini timbul akibat nilai rata-rata sinyal warna hijau atau nilai *output* rangkaian *precision rectifier* yang memiliki nilai positif. Sehingga besar *offset* dapat dihitung dengan menggunakan persamaan rata-rata fungsi, seperti pada persamaan 3.4.

$$f_{ave} = \frac{1}{b-a} \int_a^b f(x) dx \quad (3.4)$$

Karena sinyal *output* rangkaian *precision rectifier* merupakan fungsi sinus dengan setengah siklus, maka didapatkan persamaan 3.5.

$$f_{ave} = \frac{1}{\pi-0} \int_0^\pi \sin(x) dx \quad (3.5)$$

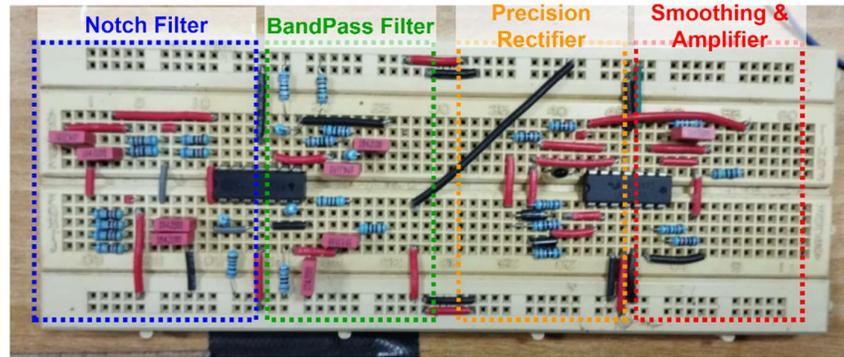
$$f_{ave} = \frac{1}{\pi} [-\cos(x)]_0^\pi$$

$$f_{ave} = \frac{1}{\pi} (-\cos(\pi) - (-\cos(0)))$$

$$f_{ave} = \frac{2}{\pi} \approx 0,636$$

Sehingga didapatkan nilai *offset* sebesar -0,636 V untuk hasil keluaran rangkaian *smoothing* dengan *gain* sebesar -1 kali.

Selama simulasi dilakukan, dibuat rangkaian pada breadboard dan menguji rangkaian yang telah dirancang seperti pada Gambar 3.19 terlebih dahulu. Rangkaian diuji dengan menggunakan osiloskop dan function generator.



Gambar 3.19 Rangkaian Sensor EMG pada breadboard

Hasil implementasi simulasi pada rangkaian ini kemudian dilakukan peningkatan dengan menambahkan rangkaian *driven right leg*. Berdasarkan [27] Pengukuran sinyal *biopotential* sangat sensitif terhadap gangguan elektromagnetik/*electromagnetic interference* (EMI). Oleh karena itu, penting untuk mengurangi impedansi elektroda *ground* atau mengimbangi arus yang dihasilkan EMI. Hasil ini dapat dicapai dengan menggunakan elektroda *driven right leg* (DRL). Elektroda ini biasanya dipasang pada kaki kanan, tetapi dapat juga dipasang di tempat lain pada tubuh pasien. Hal ini dilakukan untuk mengurangi *noise* lingkungan yang masuk ke tubuh khususnya pada frekuensi 50 Hz.

3.2.2 Hambatan dan Solusi Implementasi Rangkaian Filter

Dalam mengimplementasi subsistem *elbow sleeve* pada bagian rangkaian filter terdapat beberapa masalah dalam pengembangan yang dilakukan. Pertama, pembacaan sinyal EMG masih mendeteksi adanya sinyal 50Hz yang cukup besar. Solusi untuk permasalahan pertama adalah menambah rangkaian *driven right leg* pada elektrode yang menjadi referensi. Kedua, rangkaian sensor EMG pada perancangan

awal bagian *bandpass filter*, tidak menunjukkan perubahan amplitudo tegangan ketika nilai frekuensi diubah pada layar osiloskop. Sehingga solusinya adalah dengan menambahkan filter *band stop* pada 50Hz. Terakhir, terdapat *load* pada rangkaian sensor EMG sehingga membuat penguatan rangkaian tidak maksimal pada bagian *smoothing*. Solusi untuk permasalahan terakhir adalah dengan menambahkan resistor untuk meningkatkan *Gain* sinyal pada op-amp (tidak menggunakan rangkaian *unity gain*).

3.2.3 Hasil Implementasi Pembacaan Sensor IMU

Selanjutnya dilakukan pengujian untuk menentukan posisi ujung lengan dalam koordinat kartesian 3 dimensi dengan menggunakan sensor MPU-6050. Sebelum pengujian dilakukan, kalibrasi dilakukan terlebih dahulu dalam pembacaan percepatan melalui *accelerometer*. MPU-6050 memiliki fitur DMP (*Digital Motion Processor*).

DMP merupakan sejenis *hardware accelerator, motion processor* ini tertanam dalam sensor MEMS dan mampu menggabungkan data dari sensor untuk memberikan pengukuran yang lebih akurat terhadap orientasi, gerakan, dan posisi perangkat di ruang angkasa. Komponen tersebut sering diintegrasikan ke dalam mikrokontroler atau mikroprosesor dan digunakan dalam aplikasi pendeteksi gerakan seperti smartphone, jam tangan pintar, dan pengontrol game. DMP dirancang untuk membebaskan beberapa tugas pemrosesan dari prosesor utama, seperti penggabungan data sensor dan pemrosesan gerakan, serta memberikan kemampuan deteksi gerakan yang lebih akurat dan andal. Dengan kata lain, DMP sudah menjadi pengganti penuh untuk algoritma pendeteksi gerakan untuk sensor MEMS, yang diimplementasikan dalam perangkat keras sebagai koprocesor [28].

Pembacaan nilai *roll* dan *pitch* pada MPU-6050 beserta nilai akselerasi nya dibantu dengan menggunakan library “MPU6050_6Axis_MotionApps612.h”. Library tersebut memanfaatkan fitur DMP yang ada pada MPU-6050. Sehingga

pemrosesan data perhitungan bisa dilakukan di luar Arduino Nano. Hal ini dilakukan untuk mengurangi beban kerja pada Arduino Nano.

Nilai perpindahan ditentukan dengan melakukan integral dua kali dari nilai percepatan yang ada. Walaupun sudah menggunakan bantuan DMP MPU-6050 pada Arduino Nano, hasil pembacaan nilai akselerometer terlalu sensitif terhadap getaran seperti pada Gambar 3.22 dengan garis berwarna oranye, sehingga ditambahkan *median filter* (Gambar 3.20) dan *moving average* (Gambar 3.21) untuk memperhalus bentuk sinyal yang terbaca sehingga hasilnya tampak seperti pada Gambar 3.22 dengan garis berwarna biru.

```
readings_x[readIndex] = acc[x][n];
readings_y[readIndex] = acc[y][n];
readings_z[readIndex] = acc[z][n];
readIndex = (readIndex + 1) % FILTER_N;

for(int i = 0; i < FILTER_N; i++) {
    filter_sum_x += readings_x[i];
    filter_sum_y += readings_y[i];
    filter_sum_z += readings_z[i];
}

filter_sum_x = filter_sum_x / FILTER_N;
filter_sum_y = filter_sum_y / FILTER_N;
filter_sum_z = filter_sum_z / FILTER_N;
```

Gambar 3.20 Implementasi Program *Moving Average* Filter pada Arduino Nano

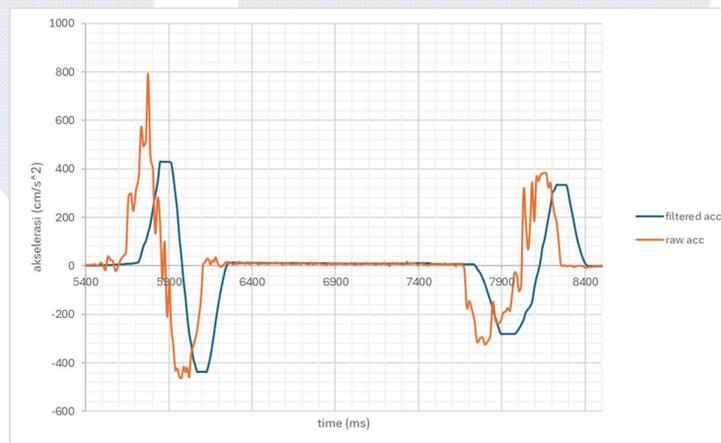
```
median_filter_x[medianIndex] = filter_sum_x;
median_filter_y[medianIndex] = filter_sum_y;
median_filter_z[medianIndex] = filter_sum_z;
medianIndex = (medianIndex + 1) % MEDIAN_FILTER_WINDOW;
for(int i = 0; i < FILTER_N; i++) {
    filter_buf[x][i] = median_filter_x[i];
    filter_buf[y][i] = median_filter_y[i];
    filter_buf[z][i] = median_filter_z[i];
}
for(int j = 0; j < FILTER_N - 1; j++) {
    for(int i = 0; i < FILTER_N - 1 - j; i++) {
        if(filter_buf[x][i] > filter_buf[x][i + 1]) {
            filter_temp_x = filter_buf[x][i];
            filter_buf[x][i] = filter_buf[x][i + 1];
            filter_buf[x][i + 1] = filter_temp_x;
        }
        if(filter_buf[y][i] > filter_buf[y][i + 1]) {
            filter_temp_y = filter_buf[y][i];
        }
    }
}
```

```

        filter_buf[y][i] = filter_buf[y][i + 1];
        filter_buf[y][i + 1] = filter_temp_y;
    }
    if(filter_buf[z][i] > filter_buf[z][i + 1]) {
        filter_temp_z = filter_buf[z][i];
        filter_buf[z][i] = filter_buf[z][i + 1];
        filter_buf[z][i + 1] = filter_temp_z;
    }
}
}
median_filtered_value_x = filter_buf[x][(FILTER_N - 1) / 2];
median_filtered_value_y = filter_buf[y][(FILTER_N - 1) / 2];
median_filtered_value_z = filter_buf[z][(FILTER_N - 1) / 2];

```

Gambar 3.21 Implementasi Program *Median Filter* pada Arduino Nano



Gambar 3.22 Hasil Implementasi Filter pada Pembacaan Akselerasi

3.2.4 Hambatan dan Solusi Implementasi Pembacaan Sensor IMU

Dalam pengimplementasian subsistem *elbow sleeve* pada bagian pembacaan sensor IMU, terdapat masalah yang harus dihadapi. Pertama, terbacanya percepatan gravitasi pada akselerometer. Hal ini menyebabkan ketidakakuratan dalam menentukan lokasi perpindahan akibat adanya percepatan diluar pergerakan. Solusi yang diterapkan untuk menyelesaikan masalah ini adalah dengan menerapkan *high pass filter*. Kedua, pembacaan sensor akselerometer terhadap percepatan terlalu sensitif dan tidak halus. Sehingga solusi yang diberikan untuk mengatasi permasalahan ini adalah dengan menambahkan *median filter* dan *moving average* untuk memperhalus sinyal tersebut.