

BAB I

PENDAHULUAN

A. Latar Belakang

Teknologi merupakan salah satu bagian penting dalam kehidupan manusia, tanpa adanya perkembangan teknologi, maka perubahan zaman tidak akan secanggih dan secepat seperti saat sekarang ini, dalam perkembangan ilmu teknologi sekarang tentu banyak manfaat yang dapat kita rasakan, salah satunya untuk mempermudah berbagai macam pekerjaan manusia mulai dari bidang industri, pemerintahan, pendidikan, dan lain sebagainya, tak terkecuali pada bidang robotika. Seiring dengan naiknya kebutuhan manusia dan teknologi yang semakin hari semakin canggih, maka dibuatlah suatu sistem yang dapat membantu pekerjaan manusia yang disebut dengan robot (Saefullah dkk, 2015). Robot menjadi suatu yang sangat penting di dunia saat ini dikarenakan robot dapat mengemban tugas dan fungsi yang fleksibel dalam membantu pekerjaan manusia. Berbagai macam jenis robot telah dibuat oleh manusia mulai dari robot bergerak (*Mobile*), lengan robot (*Robotic Arm*), robot menyerupai manusia (*Humanoid*), robot berkaki (*Legged*) dan sebagainya.

Robot merupakan suatu sistem mekanik yang dikendalikan oleh komputer dan memiliki kemampuan untuk melakukan tugas-tugas tertentu. Dalam dunia industri, robot sangat berguna karena dapat menjalankan tugas yang berulang-ulang dengan efisien dan akurat. Dalam beberapa tahun terakhir, penggunaan robot semakin berkembang dan banyak digunakan dalam berbagai bidang, seperti bidang teknologi, manufaktur, dan jasa.

Kendali suara adalah salah satu metode yang populer digunakan untuk mengendalikan robot. Metode ini memanfaatkan perintah suara untuk memberikan instruksi kepada robot. Kendali suara memberikan kemudahan bagi pengguna karena tidak perlu memasukkan perintah melalui keyboard atau mouse. Salah satu teknik yang digunakan untuk mengolah sinyal suara adalah Mel Frequency Cepstral Coefficient (MFCC) dan Adaptive Neuro-Fuzzy Inference System (ANFIS).

Tujuan dari skripsi ini adalah untuk mengembangkan sistem lengan robot dengan perintah suara menggunakan metode MFCC dan ANFIS. Metode MFCC akan digunakan untuk mengolah sinyal suara dan membuat representasi fitur suara. Sedangkan, ANFIS akan digunakan sebagai sistem pembuat keputusan untuk menentukan aksi yang harus dilakukan oleh lengan robot.

B. Rumusan Masalah

Berdasarkan uraian yang dijelaskan pada bagian latar belakang maka ditentukan rumusan masalah pada penelitian ini yaitu apakah lengan robot dengan perintah suara menggunakan metode mel frequency cepstral coefficient (mfcc) dan adaptive neuro-fuzzy inference system (anfis) dapat bekerja sebagaimana mestinya?

C. Tujuan Penelitian

1. Merancang sistem mekanik lengan robot yang efisien dan akurat dalam melakukan tugas.
2. Menggunakan metode MFCC untuk mengolah sinyal suara dan membuat representasi fitur suara.

3. Menggunakan metode ANFIS sebagai sistem pembuat keputusan untuk menentukan aksi yang harus dilakukan oleh lengan robot.
4. Sistem memberikan feedback berupa visual untuk mengonfirmasi bahwa tindakan yang dilakukan oleh lengan robot sudah sesuai dengan perintah suara.
5. Mengetahui kehandalan dan keefektifan dari sistem lengan robot dengan perintah suara menggunakan metode MFCC dan ANFIS.
6. Meningkatkan Keselamatan Manusia akibat paparan langsung terhadap bahan kimia berbahaya seperti iritasi kulit, mata, dan gangguan pernapasan

D. Batasan Masalah

Agar penelitian tidak keluar pada batasan yang telah ditentukan dan tetap terarah maka perlu adanya batasan masalah. Batasan masalah dalam penelitian ini yaitu :

1. Mikrokontroler yang digunakan yaitu Arduino Uno pada lengan robot.
2. Alat kontroler lengan robot menggunakan perintah suara dengan metode MFCC dan ANFIS.
3. Software yang digunakan MATLAB.
4. Berfokus pada pergerakan mekanik lengan robot.
5. Batasan lingkungan: Batasan lingkungan harus ditentukan dengan benar untuk memastikan bahwa sistem dapat berfungsi dengan baik dan memenuhi kebutuhan pengguna. Lingkungan harus memenuhi batasan seperti suhu, kelembaban, dan tingkat kebisingan.

E. Manfaat Penelitian

Manfaat dari penelitian ini yaitu Memperoleh pengetahuan tentang tingkat keakurasian metode yang digunakan dalam mengontrol lengan robot dengan beberapa Perintah yang berbeda, mengurangi risiko kecelakaan laboratorium dan melindungi dari paparan langsung terhadap bahan kimia berbahaya, selain itu penelitian ini juga dapat mendorong kolaborasi antar bidang kimia, robotika, dan elektro untuk menciptakan solusi lintas disiplin ilmu seperti kesehatan, farmasi, dan manufaktur. selanjutnya yaitu sebagai artikel ataupun referensi untuk penelitian selanjutnya yang berkaitan dengan lengan robot perintah suara dan juga merupakan salah satu syarat untuk memperoleh gelar sarjana program studi teknik elektro.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Kinematika Robot

Kinematika robot adalah ilmu yang mempelajari analisa pergerakan kaki atau lengan robot relatif terhadap kerangka acuan yang bergerak atau diam tanpa mempertimbangkan gaya yang menyebabkan pergerakan tersebut. Dikenal dua istilah dalam kinematika: forward kinematika dan invers kinematika. Forward kinematika adalah metode untuk menentukan posisi dan orientasi efektor akhir dari sudut sambungan dan panjang sambungan lengan robot. Invers kinematika, di sisi lain, adalah metode untuk menentukan nilai sudut sambungan yang diperlukan efektor akhir untuk mencapai posisi yang diinginkan. Hal ini merupakan kebalikan dari kinematika maju (Setiawan et al., 2015).

2. Lengan Robot

Lengan robot (*arm robot*) adalah robot mekanik yang gerakan pada robotnya dibuat hampir mirip dengan lengan manusia pada umumnya. Lengan robot memiliki beberapa sendi pergerakan tergantung kebutuhan dari lengan robot yang dibuat dan juga memiliki komponen-komponen lain yaitu aktuator dan kontroler. Lengan robot juga memiliki sistem DOF (*Degree of Freedom*) atau derajat kebebasan yang memiliki fungsi menentukan banyaknya gerakan pada robot tersebut (Cempaka dkk, 2016). Jumlah derajat kebebasan tergantung dari jumlah gerakan yang dapat dilakukan oleh lengan robot.

Lengan robot memiliki 6 bagian, Berikut adalah bagian-bagian dan definisi dari lengan robot 5 DOF:

Base (DOF 1): Base adalah bagian dasar dari lengan robot yang memungkinkan untuk rotasi pada sumbu Z. Base dapat diputar menggunakan motor atau actuator, memungkinkan lengan robot untuk mengarahkan tangan ke berbagai arah.

Shoulder (DOF 2): Shoulder adalah bagian dari lengan robot yang memungkinkan untuk rotasi pada sumbu Y. Shoulder dapat diputar menggunakan motor atau actuator, memungkinkan lengan robot untuk membungkuk dan mengangkat tangan.

Elbow (DOF 3): Elbow adalah bagian dari lengan robot yang memungkinkan untuk rotasi pada sumbu X. Elbow dapat diputar menggunakan motor atau actuator, memungkinkan lengan robot untuk membentuk siku dan membuat tangan menjauh dari tubuh.

Wrist Pitch (DOF 4): Wrist Pitch adalah bagian dari lengan robot yang memungkinkan untuk rotasi pada sumbu Y. Wrist Pitch dapat diputar menggunakan motor atau actuator, memungkinkan lengan robot untuk membungkuk dan memutar tangan.

Wrist Roll (DOF 5): Wrist Roll adalah bagian dari lengan robot yang memungkinkan untuk rotasi pada sumbu Z. Wrist Roll dapat diputar menggunakan motor atau actuator, memungkinkan lengan robot untuk memutar tangan dan melakukan gerakan putar.

Gripper adalah alat mekanik pada lengan robot yang digunakan untuk menangkap, memegang, dan melepaskan benda (J. J. Craig dkk, 2005).Gripper robot biasanya terdiri dari dua atau lebih elemen yang dapat bergerak relatif terhadap satu sama lain untuk menangkap atau memegang benda(S. K. Singh dkk, 2013).

Gripper robot digunakan dalam berbagai aplikasi, seperti pengumpulan, manufaktur, dan pengemasan, untuk memegang, memindahkan, dan melepaskan benda.Gripper robot dapat dibuat dalam berbagai bentuk dan ukuran, tergantung pada ukuran dan jenis benda yang harus ditangani(R. P. Paul , 1981).

Salah satu bentuk dari lengan robot dapat dilihat pada gambar dibawah ini.

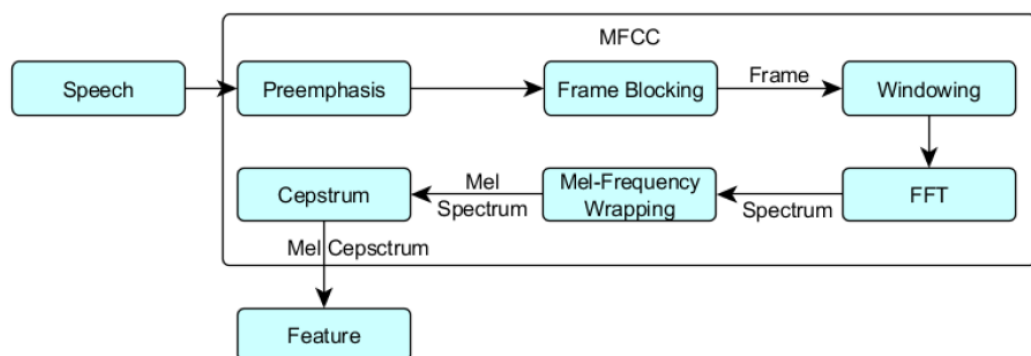


Gambar 2.1 Lengan Robot (*Sumber : Dokumentasi Pribadi*)

Lengan robot dapat dikendalikan menggunakan sensor dan aktuator (Didi dkk, 2016). Aktuator adalah perangkat elektronik yang dapat menghasilkan daya, salah satu contohnya yaitu dapat dibuat dari sistem motor listrik yaitu diantaranya motor dc, motor dc servo, motor dc stepper, ultrasonik motor dan lain sebagainya juga dapat berupa sistem pneumatik yaitu perangkat kompresi berbasis udara atau nitrogen (Jaya, 2016). Masing-masing jenis aktuator memiliki tingkat kontrol yang berbeda-beda.

3. Mel Frequency Cepstrum Coefficients (MFCC)

Metode Mel Frekuensi Cepstral Coefisien (MFCC) adalah salah satu metode ekstraksi fitur yang paling umum digunakan dalam pengolahan audio. Metode ini mengubah sinyal audio menjadi beberapa parameter. MFCC bekerja berdasarkan persepsi pendengaran manusia yang dipandu oleh perbedaan frekuensi yang didengar oleh panca indera (Umar, R., 2018). Proses ekstraksi fitur MFCC adalah sebagai berikut:



Gambar 2.2 Proses MFCC(Sumber : jtika.if.unram.ac.id)

a. Pre-Emphasis

Fungsi pra-penekanan menyaring sinyal audio untuk mengurangi nilai frekuensi sinyal audio input, sehingga hanya sinyal frekuensi tinggi yang dapat melewati proses penyaringan. Selain itu pada tahap ini dilakukan pengurangan dan peredaman noise, sehingga sistem hanya menerima sinyal wicara. Persamaannya dapat dilihat pada rumus

$$p(n) = s(n) - xs(n-1)$$

dimana x adalah konstanta filter pre-emphasis, biasanya bernilai antara $0.9 < x < 1.0$ (Helmiyah, S. et.al., 2018)

b. Frame Blocking

Sinyal suara yang telah melalui tahapan emphasis akan dibagi menjadi beberapa frame (bingkai). Setiap frame akan berisi sampel sinyal sebesar N . Pada proses ini frame yang berdekatan akan dipisahkan dengan jarak M sample. Setiap sample akan dibagi menjadi beberapa frame dengan panjang frame berdasarkan waktu berada pada rentang 20 - 40 ms . Jika frekuensi suara 20 kHz, maka jumlah sampel yang akan diekstrak adalah $0,025 \text{ s} * 20.000 \text{ Hz} = 500 \text{ sampel}$ (Chamidy, T., 2017)

c. Windowing

Tahapan framing mengakibatkan sinyal suara akan mengalami discontinue antar frame dengan frame setelahnya. Untuk melakukan perbaikan

terhadap sinyal suara digunakan Hamming Window pada sinyal tersebut (Elkusnandi, F. et.al., 2018)

Persamaannya adalah sebagai berikut :

$$w(n) = \begin{cases} 0,54 - 0,46 \cos \frac{2\pi n}{L-1}, & 0 \leq n \leq L-1 \\ 0, & n \text{ lainnya} \end{cases}$$

Dimana n merupakan sample sinyal suara ke- n dan L adalah panjangnya frame atau jumlah sampel dalam satu frame.

d. Fast Fourier Transform (FFT)

Fungsi FFT adalah mengubah sinyal dari domain waktu ke domain frekuensi. Oleh karena itu, pada tahap ini, perubahan domain frekuensi waktu terjadi pada setiap N kerangka sampel.. (Chamidy, T., 2017)

e. Mel-Frequency Wrapping

Mel Frequency Wrapping umumnya dilakukan dengan menggunakan Filterbank. Filterbank merupakan filter yang digunakan untuk mengetahui ukuran energi dari sebuah frekuensi band pada sinyal suara. Filter bank dapat diterapkan baik pada domain waktu maupun domain frekuensi, tetapi untuk keperluan MFCC, filter harus diterapkan dalam domain frekuensi. (Nasution, T., 2012). Pada sistem ini, wrapping dilakukan pada spektrum yang dihasilkan dari FFT sehingga akan menghasilkan *Mel-Scale* untuk menyesuaikan resolusi frekuensi terhadap pendengaran manusia. *Mel-Scale* dikelompokkan menjadi sejumlah bank dengan menggunakan filter bank. Dalam sebuah spektrum,

jangkauan frekuensi sangat luas dan sinyal suara tidak linear. Sehingga setelah spektrum mengalami proses komputasi, data sinyal yang dipetakan dalam skala Mel dengan filter segitiga akan saling tumpang tindih (Chamidy, T., 2017)

$$mel(f) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right)$$

$mel(f)$: fungsi mel scale

f : frekuensi

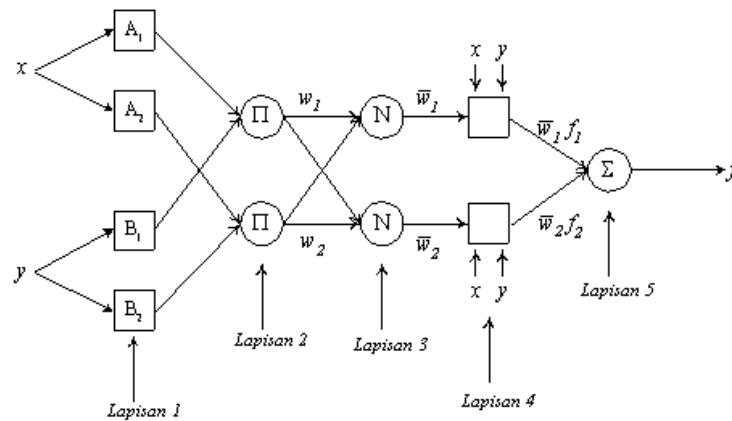
f. Discrete Cosine Transform (Cepstrum)

Pada tahapan Discrete Cosine Transform (DCT) dilakukan perubahan dari domain frekuensi ke domain waktu. Tahapan ini untuk menghasilkan nilai koefisien yang didapatkan dari hasil perkalian mel-filterbank yang sudah dibuah ke domain waktu. Pada tahapan ini akan menghasilkan log dari hasil perkalian di atas. Hasil log perkalian domain waktu ini menghasilkan mel-frequency cepstrum coefficient (MFCC) (Helmiah, S. et.al., 2018)

4. Adaptive Neuro-Fuzzy Inference System (ANFIS)

Neuro-fuzzy merupakan gabungan dari dua buah sistem yaitu sistem logika fuzzy dan jaringan syaraf tiruan. Secara umum, neuro-fuzzy bekerja berdasarkan pada sistem inferensi fuzzy yang dilatih dengan menggunakan algoritma pembelajaran. Algoritma ini diturunkan dari algoritma pada sistem jaringan syaraf tiruan. Sistem neuro-fuzzy mempunyai semua kelebihan yang dimiliki oleh sistem inferensi fuzzy dan sistem jaringan syaraf tiruan. Kemampuan neuro-fuzzy pada algoritma pembelajaran sehingga sistem ini

dikenal dengan ANFIS (Adaptive Neuro-Fuzzy Inference Systems). (Sanjaya, W.S.M. et.al., 2014)



Gambar 2.3 Struktur ANFIS

Struktur ANFIS yang terlihat pada Gambar 2.3 menggunakan inferensi fuzzy dengan model Takagi-Sugeno-Kang. Pada gambar tersebut terlihat bahwa sistem terdiri dari 5 buah lapisan, setiap lapisan yang memiliki simbol kotak merupakan lapisan yang bersifat adaptif, sedangkan yang berbentuk lingkaran bersifat tetap. (Sutisna, U. et.al., 2013)

Lapisan 1: Proses Fuzzyfication

Output atau keluran dari node i pada layer 1 dinotasikan sebagai $O1,i$.

Setiap node bersifat adaptif terhadap output:

$$O1,i = \mu A_i(x), i=1,2$$

$$O1,i = \mu B_i(y), i=1,2$$

x dan y adalah nilai-nilai input atau masukan untuk node tersebut dan A_i dan B_i merupakan himpunan fuzzy. Jadi, setiap node pada layer 1 memiliki fungsi sebagai pembangkit derajat keanggotaan.

Lapisan 2: Lapisan Product

Lapisan ini disimbolkan dengan π . Setiap node berfungsi untuk melakukan perhitungan kekuatan aktivasi (firing strength) pada setiap rule. Rule merupakan product atau hasil perhitungan dari semua input. Lapisan ini akan menghasilkan keluaran berupa hasil kali masukan w_i . Ini merupakan derajat aktivasi dari setiap rule

Lapisan 3: Lapisan Normalisasi

Lapisan ini disimbolkan dengan N . Node-node yang ada pada lapisan ini memiliki sifat non-adaptif. Node memiliki fungsi untuk menghitung rasio antara firing strength pada rule ke- i terhadap total firing strength dari semua rule. Lapisan ketiga akan melakukan perhitungan nilai hasil normalisasi dari derajat keanggotaan untuk setiap rule.

Lapisan 4: Lapisan Defuzzyfication

Lapisan keempat berfungsi untuk membuat koneksi yang memenuhi aturan bagian konsekuensi pada setiap aturan dengan output nilai fungsi $w_i.f_i$ ($x_1, x_2, \dots, x_N$)

Lapisan 5: Lapisan Total Output

Lapisan kelima terdiri dari satu node tunggal yang disimbolkan dengan Σ . Node pada layer ini akan mengagregasikan seluruh output pada layer 4 (hasil penjumlahan dari semua sinyal input).

5. MATLAB

MATLAB (Matrix Laboratory) adalah perangkat lunak yang digunakan untuk komputasi teknis, visualisasi, dan pemrograman. MATLAB memiliki

banyak toolbox yang memudahkan dalam melakukan pengolahan sinyal, analisis data, dan pemodelan sistem. MATLAB memiliki beberapa bagian utama, seperti:

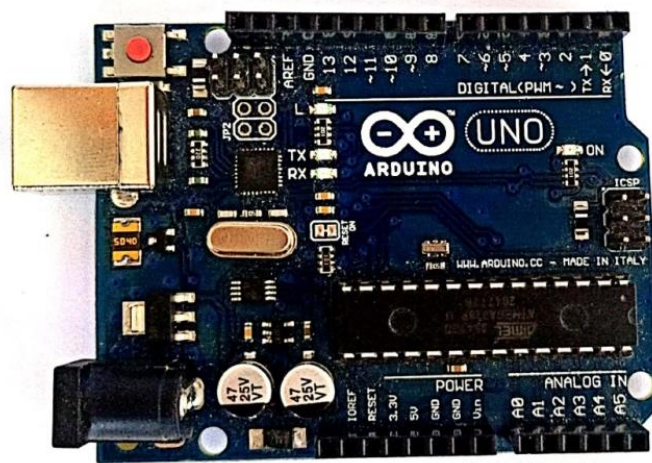
- a. Command Window: Merupakan area interaktif dimana pengguna dapat memasukkan perintah dan melihat hasilnya secara langsung.
- b. Workspace: Merupakan area dimana variabel dan data yang dibuat dan diproses dalam MATLAB disimpan.
- c. Editor/Debugger: Merupakan area dimana pengguna dapat membuat, mengedit, dan men-debug skrip MATLAB.
- d. Toolboxes: Merupakan kumpulan fungsi dan alat khusus yang dikembangkan untuk aplikasi tertentu, seperti pengolahan sinyal, analisis data, pemodelan sistem, dan sebagainya.

MATLAB memiliki banyak toolbox yang memudahkan dalam melakukan pengolahan sinyal, analisis data, dan pemodelan sistem. Dalam penelitian ini, MATLAB digunakan untuk memprogram dan menjalankan sistem pengendalian lengan robot dengan perintah suara menggunakan metode MFCC dan ANFIS.

6. Arduino

Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang didalamnya terdapat komponen utama yaitu chip mikrokontroler dengan jenis AVR yang dibuat dari perusahaan Atmel. Mikrokontroler merupakan chip atau *integrated circuit* (IC) yang dapat deprogram menggunakan komputer (Afdali dkk, 2017), ada berbagai macam jenis arduino

mulai dari Arduino USB, Arduino Serial, Arduino Mega, Arduino FIO, Arduino LILYPAD, Arduino BT, Arduino UNO R3, Arduino Nano dan sebagainya (Hutagalung, 2018). Untuk jenis arduino yang digunakan pada penelitian ini yaitu menggunakan Arduino Uno pada lengan robot, yang dapat dilihat pada Gambar 2.4.



Gambar 2.4 Arduino Uno (Sumber : Dokumentasi Pribadi)

Untuk spesifikasi Arduino Uno dapat dilihat pada tabel dibawah ini :

Tabel 2.1 Spesifikasi Arduino Uno

Mikrokontroler	ATmega328
Operasi Tegangan	5 Volt
Input Tegangan	7 – 12 Volt
Pin I/O Digital	14 (6 PWM)
Pin Analog	6

Arus DC tiap pin I/O	50 mA
Arus DC ketika 3.3V	50 mA
Memori Flash	32 Kb
SRAM	2 Kb
EEPROM	1 Kb
Kecepatan Clock	16 MHz

7. Motor Servo

Motor servo adalah salah satu jenis aktuator yang sering digunakan dalam bidang industri ataupun dalam sistem robotika (Prasetyawan dkk, 2018). Motor servo juga merupakan sebuah motor berarus DC yang dilengkapi dengan rangkaian kendali yang memiliki sistem *closed feedback* yang saling berkaitan antara satu sama lain dalam motor servo tersebut. Pada motor servo posisi putaran sumbu (*axis*) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada pada bagian dalam motor servo (Saefullah dkk, 2015).



Gambar 2.5 Motor Servo (Sumber : Dokumentasi Pribadi)

BAB III

METODE PENELITIAN

A. Waktu dan Tempat Penelitian

1. Waktu Penelitian

Waktu penelitian dilaksanakan sejak tanggal dikeluarkannya izin penelitian dalam kurun waktu kurang lebih 4 (empat) bulan dengan uraian kegiatan yaitu studi pustaka, perancangan alat, pengadaan alat dan bahan, pemasangan dan instalasi alat, Pengujian dan dilanjut pembuatan laporan akhir yang dapat dilihat pada tabel dibawah ini.

Tabel 3.1 Jadwal Pelaksanaan Penelitian

No.	Uraian	Bulan			
		1	2	3	4
1	Studi Pustaka				
2	Perancangan Alat				
3	Pengadaan Alat dan Bahan				
4	Pemasangan dan Instalasi Alat				
5	Pengujian dan Analisis				
6	Penyusunan Laporan Akhir				

2. Tempat Penelitian

Tempat pelaksanaan penelitian akan dilaksanakan di kampus II Universitas Muhammadiyah Parepare, gedung F area laboratorium Teknik Elektro dikarenakan adanya alat serta bahan yang dibutuhkan untuk pemasangan dan instalasi serta pengujian alat.

B. Jenis Penelitian

Untuk jenis penelitiannya sendiri menggunakan metode eksperimen dengan menguji fungsi rangkaian alat apakah sesuai yang diharapkan serta mengacu pada studi pustaka terdahulu yang telah dikumpulkan.

C. Alat dan Bahan

Alat dan bahan yang dibutuhkan dalam pembuatan alat dapat dilihat pada Tabel 3.2.

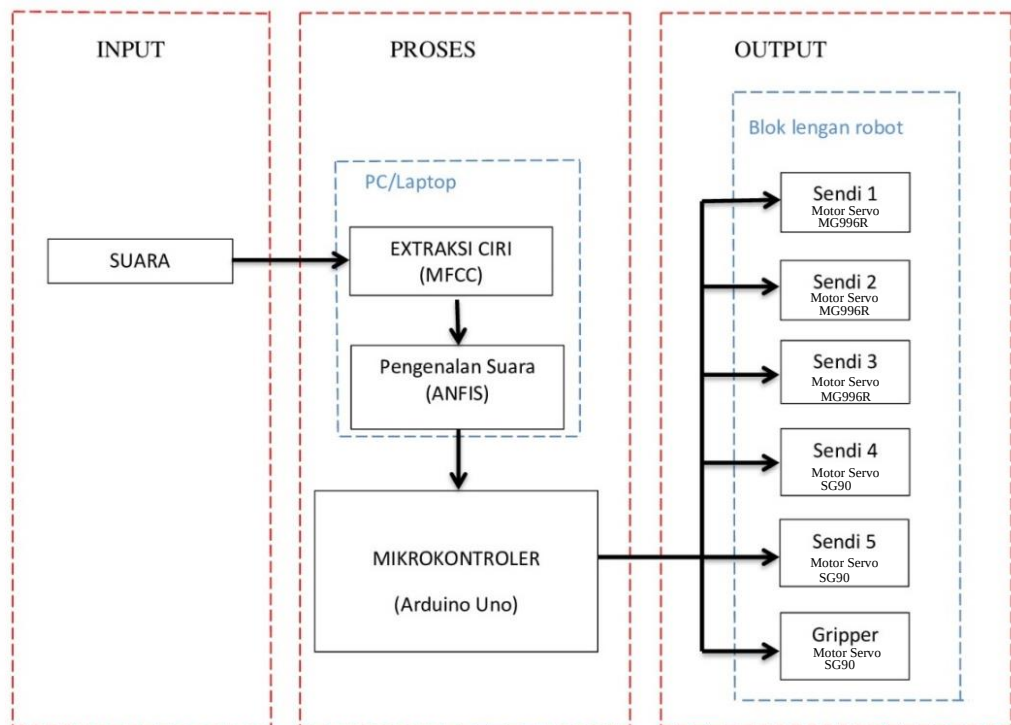
Tabel 3.2 Alat dan Bahan

No.	Uraian	Jumlah
1	Arduino Uno R3	1
2	Motor Servo SG90	3
3	Motor Servo MG996R	3
4	Modul Step Down LM2596	1
5	Baterai Li Ion 12 Volt	3
6	Kabel Jumper Male/Female	1set

D. Rancangan Penelitian

1. Blok Diagram

Untuk rancangan penelitian disajikan dalam bentuk blok diagram yang dapat dilihat pada gambar 3.1.



Gambar 3.1 Blok Diagram Rancangan Penelitian

Dari blok diagram rancangan penelitian diatas terdapat dua bagian yaitu bagian kontrol dan bagian lengan robot yang dapat dijelaskan sebagai berikut:

a. Bagian kontrol

Bagian Kontrol: Bagian ini terdiri dari beberapa blok utama seperti sensor suara, komputer, dan algoritma ANFIS. Sensor suara akan merekam suara pengguna dan mengirimkan sinyal audio ke komputer. Komputer akan menjalankan algoritma ANFIS untuk menganalisis fitur suara dan menentukan tindakan yang akan dilakukan oleh lengan robot.

b. Lengan Robot

Bagian lengan robot pada rancangan penelitian ini terdiri dari lengan robot, 5 DOF sendi, 1 gripper, 3 motor servo SG90 dan 3 motor servo MG996R, dan kontrol elektronik. Lengan robot berfungsi sebagai bagian fisik yang akan memindahkan objek, sedangkan 5 DOF sendi berfungsi untuk memindahkan lengan robot ke posisi yang diinginkan. Gripper berfungsi untuk memegang objek. Motor servo MG996R akan menggerakkan 3 sendi dan motor servo SG90 akan menggerakkan 3 sendi lainnya. Kontrol elektronik Arduino Uno akan menerima sinyal dari bagian kontrol dan memastikan bahwa semua bagian lain bekerja dengan benar sehingga lengan robot dapat bergerak dan memindahkan objek sesuai perintah suara yang diterima.



Gambar 3.2 Design Lengan robot

E. Teknik Pengumpulan Data

Pengumpulan data yang dilakukan pada penelitian ini meliputi beberapa tahap sebagai berikut:

1. Rekaman Suara: Salah satu cara untuk mengumpulkan data suara adalah dengan merekam perintah suara dari pengguna menggunakan sensor suara. Dalam hal ini, sensor suara akan merekam suara dan mengirimkan sinyal audio ke komputer untuk diproses lebih lanjut.
2. Ekstraksi Fitur: Fitur suara seperti MFCC akan diekstrak dari sinyal audio dan dipreproses sebelum digunakan dalam tahap analisis.
3. Analisis Fitur: Fitur suara akan dianalisis menggunakan metode ANFIS untuk menentukan tindakan yang akan dilakukan oleh lengan robot.
4. Identifikasi Tindakan: ANFIS akan mengidentifikasi tindakan yang akan dilakukan oleh lengan robot berdasarkan fitur suara yang dianalisis.
5. Feedback: Sistem akan memberikan feedback berupa visual atau audio untuk mengonfirmasi bahwa tindakan yang dilakukan oleh lengan robot sudah sesuai dengan perintah suara.
6. Evaluasi: Dalam tahap evaluasi, data hasil dari tindakan yang dilakukan oleh lengan robot akan dikumpulkan dan dianalisis untuk menentukan keefektifan sistem dalam memproses perintah suara dan mengontrol lengan robot.
7. Data ini akan dikumpulkan dan dianalisis secara sistematis dan tepat waktu untuk memastikan hasil yang akurat dan berkualitas. Data ini juga akan digunakan untuk memvalidasi hipotesis dan menentukan apakah sistem sudah berfungsi sebagaimana mestinya.

BAB IV

HASIL DAN PEMBAHASAN

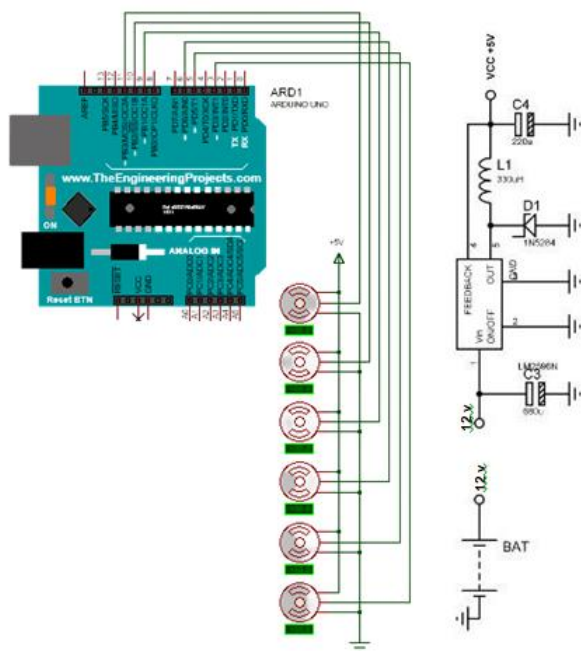
Lengan robot dengan perintah suara menggunakan metode mel frequency cepstral coefficient (mfcc) dan adaptive neuro-fuzzy inference system (anfis). pada dasarnya terdiri dari beberapa bagian yaitu perancangan perangkat keras (hardwere), perancangan perangkat lunak (softwere) dan pengujian Lengan robot dengan perintah suara.

A. Perancangan perangkat keras

Untuk komponen yang digunakan pada perancangan lengan robot 6-DOF yaitu :

1. Arduino yang digunakan pada peneliatian kali ini yaitu menggunakan Arduino Uno yang berfungsi sebagai Mikrokontroler pada sistem kendali lengan robot 6-DOF ini.
2. Servo SG90, Penggunaan servo pada penelitian ini menggunakan servo jenis SG90 dengan sudut pergerakan 180 derajat yang berfungsi sebagai sendi pergerakan pada lengan: AXIS 4 (servo4): Wrist roll: Sumbu ini memungkinkan pergelangan tangan lengan robot untuk berputar ke kiri dan kanan, AXIS 5 (servo5): Wrist pitch: Sumbu ini memungkinkan pergelangan tangan lengan robot untuk bergerak ke atas dan ke bawah, AXIS 6 (servo6): End Effector / Pointer: Sumbu ini memungkinkan ujung lengan robot untuk menjepit dan memegang objek.

3. Servo MG996R, Penggunaan servo pada penelitian ini menggunakan servo jenis MG996R dengan sudut pergerakan 180 derajat yang berfungsi sebagai sendi pergerakan pada lengan: AXIS 1 (servo1): Base: Sumbu ini memungkinkan lengan robot untuk berputar ke kiri dan kanan, AXIS 2 (servo2): Shoulder: Sumbu ini memungkinkan lengan robot untuk bergerak maju dan mundur, AXIS 3 (servo3): Elbow: Sumbu ini memungkinkan lengan robot untuk menekuk dan meluruskan.
4. Step Down DC-DC yang digunakan pada penelitian ini yaitu Step Down LM2596 DC-DC yang berfungsi untuk menurunkan tegangan dari Power Supply dengan tegangan 12V menjadi 5V yang akan digunakan pada pin vcc motor servo.



Gambar 4.1 Rancangan Rangkaian Lengan Robot Dengan Perintah Suara

Adapun komponen dan perangkat yang digunakan pada perancangan perangkat keras pada penelitian ini, yaitu dapat dilihat pada gambar dibawah ini.

Pada perancangan alat lengan robot 6-DOF ada 6 titik koordinat tempat mengambil tabung yang mana tabung akan di angkat kemudian dituangkan kedalam wadah yang telah disediakan dan wadah untuk meletakkan tabung kosong. Ada 6 (enam) servo yang berfungsi menggerakkan lengan robot yaitu servo 1 (satu) untuk berputar ke kiri dan kanan, servo 2 (dua) untuk pergerakan maju mundur, servo 3 (tiga) untuk menekuk dan meluruskan, servo 4 (empat) untuk berputar ke kiri dan kanan, servo 5 (lima) untuk pergerakan ke atas dan ke bawah dan yang terakhir yaitu servo 6 (enam) untuk menjepit dan memegang objek.

AXIS 1 (servo1): Base: Sumbu ini memungkinkan lengan robot untuk berputar ke kiri dan kanan.

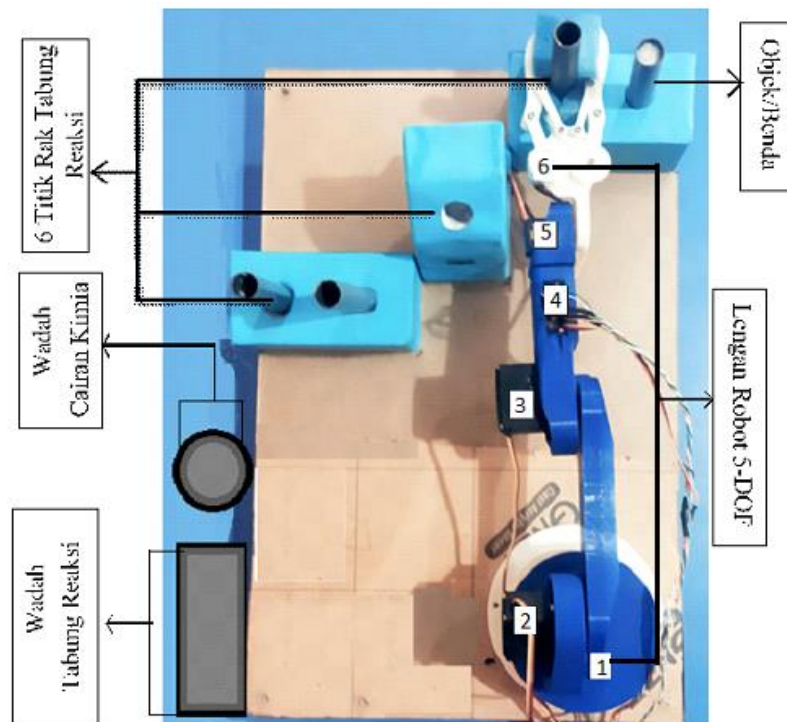
AXIS 2 (servo2): Shoulder: Sumbu ini memungkinkan lengan robot untuk bergerak maju dan mundur.

AXIS 3 (servo3): Elbow: Sumbu ini memungkinkan lengan robot untuk menekuk dan meluruskan.

AXIS 4 (servo4): Wrist roll: Sumbu ini memungkinkan pergelangan tangan lengan robot untuk berputar ke kiri dan kanan.

AXIS 5 (servo5): Wrist pitch: Sumbu ini memungkinkan pergelangan tangan lengan robot untuk bergerak ke atas dan ke bawah.

AXIS 6 (servo6): End Effector / Pointer: Sumbu ini memungkinkan ujung lengan robot untuk menjepit dan memegang objek.



Gambar 4.2 Rancangan Lengan Robot Dengan Perintah Suara

Untuk komponen yang digunakan pada perancangan lengan robot 5-DOF+1Gripper yaitu :

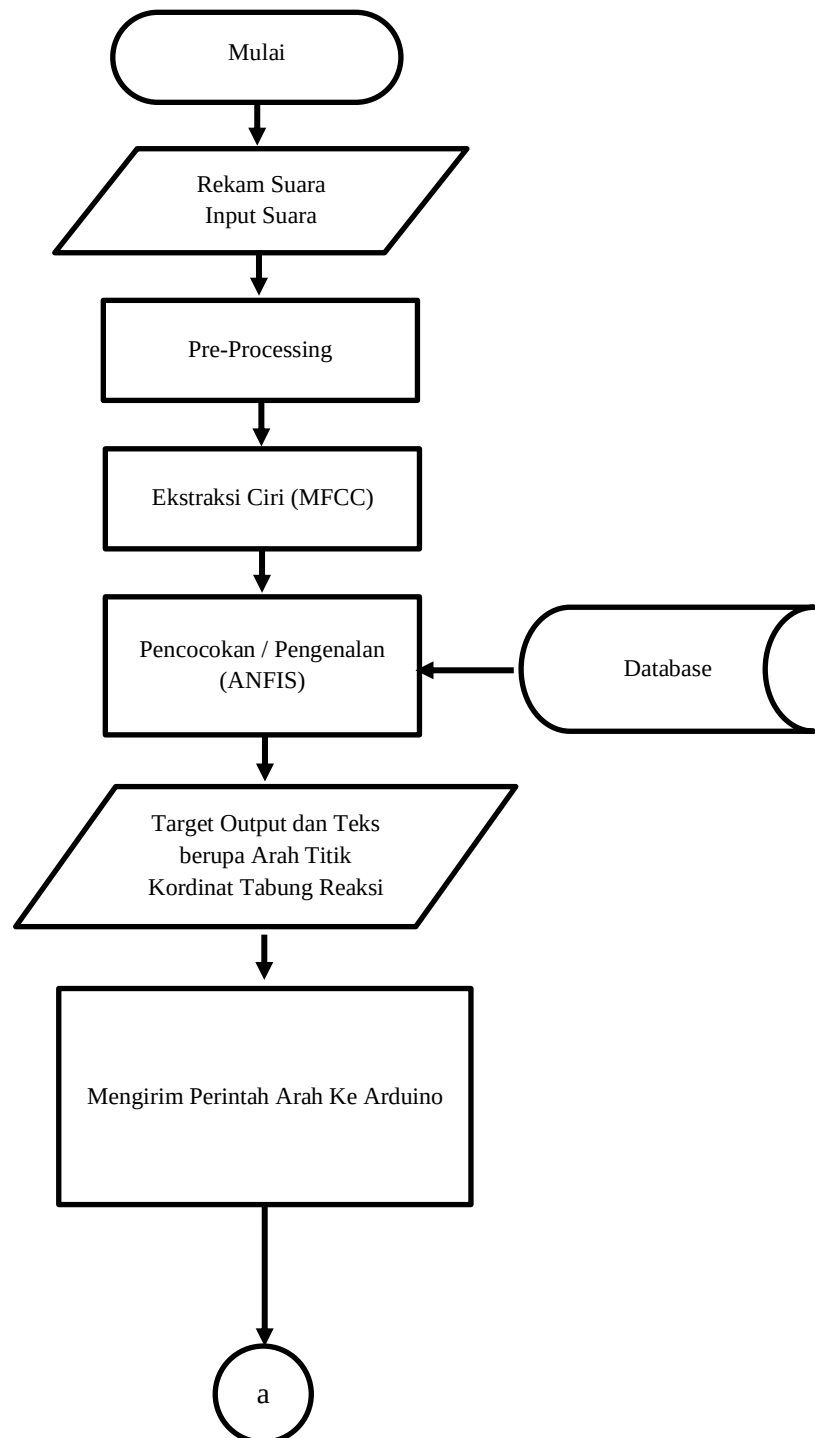
1. Arduino yang digunakan pada penelitian kali ini yaitu menggunakan Arduino Uno yang berfungsi sebagai Mikrokontroler pada sistem kendali lengan robot 6-DOF ini.
2. Servo SG90, Penggunaan servo pada penelitian ini menggunakan servo jenis SG90 dengan sudut pergerakan 180 derajat yang berfungsi sebagai sendi pergerakan pada lengan: AXIS 4 (servo4): Wrist roll: Sumbu ini memungkinkan pergelangan tangan lengan robot untuk berputar ke kiri dan kanan, AXIS 5 (servo5): Wrist pitch: Sumbu ini memungkinkan

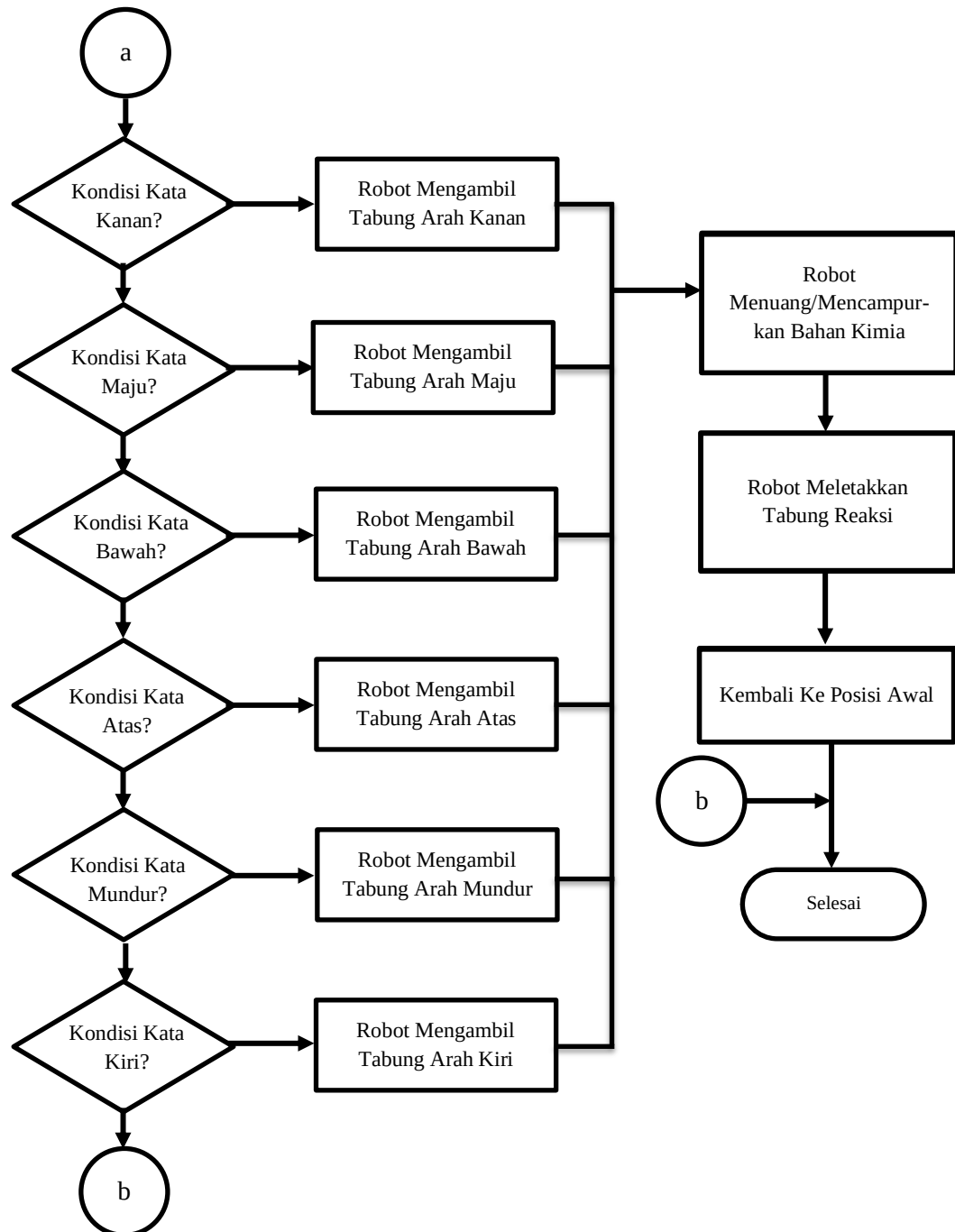
pergelangan tangan lengan robot untuk bergerak ke atas dan ke bawah, AXIS 6 (servo6): End Effector / Pointer: Sumbu ini memungkinkan ujung lengan robot untuk menjepit dan memegang objek.

3. Servo MG996R, Penggunaan servo pada penelitian ini menggunakan servo jenis MG996R dengan sudut pergerakan 180 derajat yang berfungsi sebagai sendi pergerakan pada lengan: AXIS 1 (servo1): Base: Sumbu ini memungkinkan lengan robot untuk berputar ke kiri dan kanan, AXIS 2 (servo2): Shoulder: Sumbu ini memungkinkan lengan robot untuk bergerak maju dan mundur, AXIS 3 (servo3): Elbow: Sumbu ini memungkinkan lengan robot untuk menekuk dan meluruskan.
4. Step Down DC-DC yang digunakan pada penelitian ini yaitu Step Down LM2596 DC-DC yang berfungsi untuk menurunkan tegangan dari Power Supply dengan tegangan 12V menjadi 5V yang akan digunakan pada pin vcc motor servo.

B. Perancangan Software

Perancangan perangkat lunak atau software dilakukan dengan menggunakan aplikasi MATLAB. Baris kode program yang telah dibuat pada MATLAB akan di kirim masuk ke perangkat Arduino Uno. Perancangan perangkat lunak atau perancangan software ini bertujuan untuk mengatur kinerja pada input dan output dari perangkat keras dengan adanya instruksi-instruksi yang dimasukkan ke Mikrokontroler. Flowchart prinsip kerja sistem dapat dilihat pada gambar dibawah ini.





Gambar 4.3 *Flowchart*

Pada Gambar 4.2 ditampilkan diagram alir lengan robot yang menggunakan perintah suara dengan metode MFCC dan ANFIS. Diagram ini menunjukkan tahap-

tahap yang dilakukan oleh lengan robot dalam mengeksekusi instruksi suara dari pengguna.

Seluruh langkah dari sistem yang dirancang dibuat dalam satu aplikasi yang dilakukan dengan memanfaatkan software Matlab. Pada Gambar 3. 4 ditampilkan Graphic User Interface (GUI) Matlab dari sistem pengenalan suara yang telah dirancang.



Gambar 4.4 Tampilan *GUI* Lengan Robot

Proses dimulai dengan menerima perintah dari antarmuka pengguna (GUI). Perintah suara tersebut kemudian diproses dengan algoritma MFCC untuk mengekstraksi fitur suara. Fitur-fitur suara ini selanjutnya diproses oleh algoritma ANFIS untuk menghasilkan output berupa instruksi gerakan bagi lengan robot.

Jumlah data ciri yang ditampilkan dalam sistem ini adalah 6 ciri, yaitu Ciri 1 – Ciri 6. Data ciri yang berbeda ditampilkan untuk setiap file audio. Setiap data ciri mendapatkan output target secara manual. Nilai keluaran target ditetapkan

untuk mempermudah proses pengklasifikasian suara. Suara yang memiliki pengucapan serupa diberi nilai keluaran target yang identik.

Dalam studi ini digunakan 6 kata, yaitu Kanan, Maju, Atas, Bawah, Mundur, dan Kiri. Setiap kata diberikan target output dari 1 – 6 seperti yang terlihat pada tabel berikut ini :

Tabel 4.1 Target Output

Kata	Target Output
Kanan	1
Maju	2
Atas	3
Bawah	4
Mundur	5
Kiri	6

Setelah input suara dilabeli sesuai Target Output dan Teks berupa Kata telah dikenali sesuai dengan kata yang benar ,setiap kata akan dikirim ke arduino berdasarkan intruksi masing masing setiap titik kordinat tabung reaksi.

Output perintah gerakan lengan robot kemudian dikirim ke mikrokontroler untuk dieksekusi. Mikrokontroler akan mengaktifkan aktuator-aktuator pada lengan robot untuk menggerakkan lengan robot sesuai dengan perintah yang diterima.

Berikut adalah penjelasan lebih rinci dari setiap langkah pada flowchart lengan robot :

- Mulai

Sistem dimulai dengan proses inisialisasi, termasuk mengatur perangkat

keras seperti mikrofon untuk merekam suara dan Arduino untuk kendali robot.

- Rekam dan Input Suara
 - Pengguna memberikan perintah suara yang direkam menggunakan mikrofon. Data suara ini kemudian diolah sebagai input untuk langkah selanjutnya.
 - Tahap ini penting untuk memastikan kualitas input suara yang cukup baik, bebas noise, agar sistem dapat mengenali pola dengan akurasi tinggi.
- Pre-Processing
 - Tahap pra-pemrosesan mencakup normalisasi sinyal suara, penghapusan noise, dan framing.
 - Hasilnya adalah data suara yang bersih dan siap untuk tahap ekstraksi fitur.
- Ekstraksi Ciri (MFCC)
 - MFCC digunakan untuk mengekstrak fitur penting dari sinyal suara, seperti informasi frekuensi yang relevan dengan karakteristik vokal.
 - Output dari tahap ini adalah vektor fitur yang mewakili suara input dalam bentuk data numerik.
- Pencocokan / Pengenalan (ANFIS)
 - Fitur hasil MFCC dimasukkan ke model ANFIS yang telah dilatih sebelumnya.
- Output Target dan Pengiriman Perintah

- Berdasarkan hasil pengenalan ANFIS, sistem menentukan koordinat atau arah yang harus dituju robot.
- Sistem mengirimkan perintah ke Arduino untuk menggerakkan robot sesuai arah yang dikenali.
- Eksekusi Perintah Robot
 - Robot melaksanakan tugas spesifik berdasarkan perintah suara, seperti mengambil tabung reaksi, mencampur bahan kimia, atau kembali ke posisi awal.
 - Setiap perintah seperti "kanan", "kiri", "atas", "bawah", "maju", atau "mundur" memiliki subproses tersendiri untuk mengontrol servo robot.
- Selesai
 - Sistem kembali ke kondisi awal atau menunggu input suara berikutnya jika perintah telah selesai dilaksanakan.

Flowchart ini menggambarkan langkah-langkah yang dilakukan lengan robot untuk menjalankan perintah suara dari pengguna dengan jelas dan rinci.

Untuk sketch program mell-frekuensi cepstral coefficients yaitu sebagai berikut :

```

Fs = 48000;
recObj = audiorecorder(Fs, 16, 1); % Membuat objek perekam audio
disp('Mulai merekam...');
recordblocking(recObj, 2); % Merekam selama 2 detik
disp('Rekaman selesai.');
```

```

suara = getaudiodata(recObj); % Mengambil data audio

% Normalisasi amplitudo
x1 = abs(suara);
xmax = max(x1);
xnorm = suara / xmax;
```



```

s = potong(xnorm);

% Menyimpan file audio ke format WAV
audiowrite('hasil(1).wav', s, Fs);

% Membaca file audio yang disimpan
s = audioread('hasil(1).wav');

% Memutar audio (opsional, jika dibutuhkan)
% sound(s, Fs);
% Plot sinyal audio
plot(s);
num = 6;

n = 512;          % Number of FFT points
Tf = 0.025;       % Frame duration in seconds
N = Fs * Tf;      % Number of samples per frame
fn = 24;          % Number of mel filters
l = length(s);    % Total number of samples in speech
Ts = 0.01;        % Frame step in seconds
FrameStep = Fs * Ts; % Frame step in samples
a = 1;
b = [1, -0.97];   % a and b are high pass filter coefficients

noFrames = floor(l / FrameStep); % Maximum no of frames in speech sample
FMatrix = zeros(noFrames - 2, num); % Matrix to hold cepstral coefficients
lifter = 1:num; % Lifter vector index
lifter = 1 + floor((num) / 2) * (sin(lifter * pi / num)); % Raised sine lifter version

if mean(abs(s)) > 0.01
    s = s / max(s); % Normalises to compensate for mic vol differences
end

% Segment the signal into overlapping frames and compute MFCC coefficients
for i = 1:noFrames - 2
    frame = s((i - 1) * FrameStep + 1:(i - 1) * FrameStep + N); % Holds individual frames
    Ce1 = sum(frame.^2); % Frame energy
    Ce2 = max(Ce1, 2e-22); % Floors to 2 X 10 raised to power -22
    Ce = log(Ce2);
    framef = filter(b, a, frame); % High pass pre-emphasis filter
    F = framef .* hamming(N); % Multiplies each frame with hamming window
    FFTo = fft(F, N); % Computes the FFT
    melf = melbankm(fn, n, Fs); % Creates 24 filter, mel filter bank
    halfn = 1 + floor(n / 2);
    spectr1 = log10(melf * abs(FTTo(1:halfn)).^2); % Result is mel-scale filtered
    spectr = max(spectr1(:), 1e-22);
    c = dct(spectr); % Obtains DCT, changes to cepstral domain
    c(1) = Ce; % Replaces first coefficient
    coeffs = c(1:num); % Retains first num coefficients
    ncoeffs = coeffs .* lifter; % Multiplies coefficients by lifter value
    FMatrix(i, :) = ncoeffs;
    koef = ncoeffs;
end
K1 = koef(1);
K2 = koef(2);
K3 = koef(3);
K4 = koef(4);
K5 = koef(5);
K6 = koef(6);

```

Untuk sketch program pengujian data suara dengan anfis yaitu sebagai berikut :

```
% Memastikan file 'suaraberhasil.fis' dapat diakses
fisFilePath = 'suaraberhasil.fis';
if exist(fisFilePath, 'file')% Membaca file ANFIS
    fis = readfis(fisFilePath);
    % Mengevaluasi input menggunakan FIS
    out = evalfis([K1, K2, K3, K4, K5, K6], fis);
    set(handles.outANFIS, 'String', out);
    % Menentukan hasil berdasarkan output dari FIS
myform=guidata(gcbo);
    if out >= 0.5 && out < 1.5
        set(myform.deteksiKata, 'String', 'KNAN');
    elseif out >= 1.5 && out < 2.5
        set(myform.deteksiKata, 'String', 'MAJU');
    elseif out >= 2.5 && out <= 3.5 %3
        set(myform.deteksiKata, 'String', 'ATAS');
    elseif out >= 3.5 && out <= 4.5
        set(myform.deteksiKata, 'String', 'BWAH');
    elseif out >= 4.5 && out <= 5.5
        set(myform.deteksiKata, 'String', 'MNDNR');
    elseif out >= 5.5 && out <= 6.5
        set(myform.deteksiKata, 'String', 'KIRI');
    else
        set(myform.deteksiKata, 'String', 'KATA TIDAK TERDETEKSI');
    end
else% Menampilkan pesan error jika file tidak ditemukan
    errorldg('File 'suaraberhasil.fis tidak ditemukan', 'Error');
end %akhir perbaikan program nable to open file 'suaraberhasil.fis'
```

Sketch program untuk kendali lengan robot untuk mengatur titik koordinat dari posisi 1 (satu) hingga posisi 6 (Enam) dibagi atas 6 sketch, Untuk sketch program arah titik koordinat 'KNAN'(kanan) yaitu sebagai berikut :

```
writePosition(servoMotor1, 135/180);
writePosition(servoMotor2, 70/180);
writePosition(servoMotor3, 180/180);
writePosition(servoMotor4, 0/180);
writePosition(servoMotor5, 170/180);
writePosition(servoMotor6, 90/180);
pause(0.1); % Tunggu beberapa detik agar servo mencapai sudut awal dengan stabil % KANAN JEPIT
targetAngle3 = 180; %60
targetAngle2 = 70; %0
targetAngle5 = 170; %100
targetAngle4 = 0; %0
targetAngle = 135;% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
writePosition(servoMotor3, targetAngle3/180);
writePosition(servoMotor2, targetAngle2/180);
writePosition(servoMotor5, targetAngle5/180);% Tunggu beberapa detik
pause(0.4);
numSteps = 50; % Jumlah langkah untuk pergerakan halus%kanan
```

```

targetAngles = linspace(135, 45, numSteps);
for i = 1:numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end % maju,bukagripper,jepit
targetAngle3 = 180; % Sudut target servoMotor3 saat maju
targetAngle2 = 70; % Sudut target servoMotor2 saat maju
targetAngle5 = 170; %170
targetAngle4 = 0;
targetAngle6 = 90; % Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180); % Tunggu beberapa detik
pause(0.4); % Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    currentAngle3 = targetAngle3 - (targetAngle3 - 40) * (i / numSteps);
    currentAngle2 = targetAngle2 - (targetAngle2 - 5) * (i / numSteps);
    %currentAngle5 = targetAngle5 - (targetAngle5 - 100) * (i / numSteps);
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    writePosition(servoMotor3, currentAngle3 / 180);
    writePosition(servoMotor2, currentAngle2 / 180);
    %writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
    currentAngle5 = targetAngle5 - (targetAngle5 - 100) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02);
for i = 1:numSteps % targetAngle6 = 180;
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
    pause(0.0002); %ATAS
    targetAngle4 = 0;
    targetAngle3 = 40; % Menggerakkan servo ke sudut target
    writePosition(servoMotor4, targetAngle4/180);
    currentAngle3 = targetAngle3 - (targetAngle3 - 20) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    targetAngle2 = 0; % Sudut target servoMotor2 saat maju
    targetAngle5 = 90;
    numSteps = 50; % Jumlah langkah untuk pergerakan halus
    for i = 1:numSteps
        currentAngle2 = targetAngle2 - (targetAngle2 - 90) * (i / numSteps);
        currentAngle5 = targetAngle5 - (targetAngle5 - 0) * (i / numSteps);
        writePosition(servoMotor2, currentAngle2 / 180);
        writePosition(servoMotor5, currentAngle5 / 180);
        pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
    end%KIRI
    targetAngles = linspace(45, 135, numSteps);
    for i = 1:numel(targetAngles)
        targetAngle = targetAngles(i);
        writePosition(servoMotor1, targetAngle/180);
        pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
    end
    pause(0.02); %BAWAH KIRI
    targetAngle3 = 20;
    targetAngle5 = 0;
    targetAngle2 = 90;
    numSteps = 50;
    for i = 1:numSteps% Menggerakkan servo ke sudut target
        currentAngle2 = targetAngle2 - (targetAngle2 - 40) * (i / numSteps);

```

```

currentAngle3 = targetAngle3 + (50 - targetAngle3) * (i / numSteps);
writePosition(servoMotor2, currentAngle2/180);
writePosition(servoMotor3, currentAngle3/180);
currentAngle5 = targetAngle5 - (targetAngle5 - 90) * (i / numSteps);
writePosition(servoMotor5, currentAngle5 / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle4 = 0;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (180 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02);
end
pause(0.1); % Tunggu sebentar sebelum melanjutkan loop
targetAngle4 = 180;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (90 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end %LEPAS
targetAngle = 135;
targetAngle3 = 50;
numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 160) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps % Menggerakkan servo ke sudut target
    currentAngle3 = targetAngle3 + (105 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle6 = 90;
for i = 1:numSteps
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end % kembali ke posisi awal
targetAngle = 160;
targetAngle2 = 40;
targetAngle3 = 105;
targetAngle4 = 90;
targetAngle5 = 90;
targetAngle6 = 180;
numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 135) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    currentAngle2 = targetAngle2 - (targetAngle2 - 70) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    currentAngle3 = targetAngle3 + (180 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle4 = targetAngle4 + (0 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.00002); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps % Menggerakkan servo ke sudut target
    currentAngle5 = targetAngle5 - (targetAngle5 - 170) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);

```

```

    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end

```

Sketch program untuk titik koordinat posisi pertama dengan arah kanan mencakup servo 1 (satu) yang memiliki pembacaan sudut sebesar 45° , servo 2 (dua) dengan sudut 5° , servo 3 (tiga) pada sudut 40° , servo 4 (empat) dengan sudut 0° , servo 5 (lima) pada sudut 100° , dan servo 6 (enam) dengan sudut 20° .

Untuk sketch program arah titik koordinat 'MAJU'(maju) yaitu sebagai berikut :

```

writePosition(servoMotor1, 135/180);
writePosition(servoMotor2, 70/180);
writePosition(servoMotor3, 180/180);
writePosition(servoMotor4, 0/180);
writePosition(servoMotor5, 170/180);
writePosition(servoMotor6, 90/180);
pause(0.1); % Tunggu beberapa detik agar servo mencapai sudut awal dengan stabil% MAJU JEPIT
targetAngle3 = 180; %60
targetAngle2 = 70; %0
targetAngle5 = 170; %100
targetAngle4 = 0; %0
targetAngle = 135;% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
writePosition(servoMotor3, targetAngle3/180);
writePosition(servoMotor2, targetAngle2/180);
writePosition(servoMotor5, targetAngle5/180);% Tunggu beberapa detik
pause(0.4); % % Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus%kanan
targetAngles = linspace(135, 55, numSteps);
for i = 1:numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end% maju,bukagripper,jepit
targetAngle3 = 180; % Sudut target servoMotor3 saat maju
targetAngle2 = 70; % Sudut target servoMotor2 saat maju
targetAngle5 = 170; %170
targetAngle4 = 0;
targetAngle6 = 90;% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);% Tunggu beberapa detik
pause(0.4);% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    currentAngle3 = targetAngle3 - (targetAngle3 - 40) * (i / numSteps);
    currentAngle2 = targetAngle2 - (targetAngle2 - 5) * (i / numSteps);
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    writePosition(servoMotor3, currentAngle3 / 180);
    writePosition(servoMotor2, currentAngle2 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end

```

```

    currentAngle5 = targetAngle5 - (targetAngle5 - 100) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
pause(0.02);
for i = 1:numSteps
% targetAngle6 = 180;
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
pause(0.0002);%ATAS
targetAngle4 = 0;
targetAngle3 = 40;% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
currentAngle3 = targetAngle3 - (targetAngle3 - 20) * (i / numSteps);
writePosition(servoMotor3, currentAngle3/180);% Tunggu beberapa detik
%pause(0.1);
% Sudut target servoMotor3 saat maju
targetAngle2 = 0; % Sudut target servoMotor2 saat maju
targetAngle5 = 90;% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    currentAngle2 = targetAngle2 - (targetAngle2 - 90) * (i / numSteps);
    currentAngle5 = targetAngle5 - (targetAngle5 - 0) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2 / 180);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus%KIRI
targetAngles = linspace(55, 135, numSteps);
for i = 1: numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
    pause(0.02);%BAWAH KIRI
targetAngle3 = 20;
targetAngle5 = 0;
targetAngle2 = 90;
numSteps = 50;
for i = 1:numSteps% Menggerakkan servo ke sudut target
    currentAngle2 = targetAngle2 - (targetAngle2 - 40) * (i / numSteps);
    currentAngle3 = targetAngle3 + (50 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle5 = targetAngle5 - (targetAngle5 - 90) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle4 = 0;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (180 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02);
end
pause(0.1); % Tunggu sebentar sebelum melanjutkan loop
targetAngle4 = 180;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (90 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end%LEPAS
targetAngle = 135;
targetAngle3 = 50;

```

```

numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 160) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps% Menggerakkan servo ke sudut target
    currentAngle3 = targetAngle3 + (105 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle6 = 90;
for i = 1:numSteps
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end% kembali ke posisi awal
targetAngle = 160;
targetAngle2 = 40;
targetAngle3 = 105;
targetAngle4 = 90;
targetAngle5 = 90;
targetAngle6 = 180;
numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 135) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    currentAngle2 = targetAngle2 - (targetAngle2 - 70) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    currentAngle3 = targetAngle3 + (180 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle4 = targetAngle4 + (0 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.00002); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps% Menggerakkan servo ke sudut target
    currentAngle5 = targetAngle5 - (targetAngle5 - 170) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180); pause(0.02); % Tunggu sebelum melanjutkan loo
end

```

Sketch program untuk titik koordinat posisi kedua dengan arah maju mencakup servo 1 (satu) yang memiliki pembacaan sudut 55°, servo 2 (dua) pada sudut 5°, servo 3 (tiga) dengan sudut 40°, servo 4 (empat) pada sudut 0°, servo 5 (lima) dengan sudut 100°, dan servo 6 (enam) pada sudut 20°.

Untuk sketch program arah titik koordinat 'ATAS'(atas) yaitu sebagai berikut:

```

writePosition(servoMotor1, 135/180);
writePosition(servoMotor2, 70/180);
writePosition(servoMotor3, 180/180);

```

```

writePosition(servoMotor4, 0/180);
writePosition(servoMotor5, 170/180);
writePosition(servoMotor6, 180/180);
pause(0.1); % Tunggu beberapa detik agar servo mencapai sudut awal dengan stabil

% MAJU JEPIT
targetAngle3 = 180; %60
targetAngle2 = 70; %0
targetAngle5 = 170; %100
targetAngle4 = 0; %0
targetAngle = 135;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
writePosition(servoMotor3, targetAngle3/180);
writePosition(servoMotor2, targetAngle2/180);
writePosition(servoMotor5, targetAngle5/180);
% Tunggu beberapa detik
pause(0.4);

% % Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
%kanan
targetAngles = linspace(135, 65, numSteps);
for i = 1:numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
% maju,bukagripper,jepit
targetAngle3 = 180; % Sudut target servoMotor3 saat maju
targetAngle2 = 70; % Sudut target servoMotor2 saat maju
targetAngle5 = 170; %170
targetAngle4 = 0;
% targetAngle6 = 90;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
% Tunggu beberapa detik
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    currentAngle3 = targetAngle3 - (targetAngle3 - 50) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
    currentAngle2 = targetAngle2 - (targetAngle2 - 30) * (i / numSteps);
    currentAngle5 = targetAngle5 - (targetAngle5 - 55) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2 / 180);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
pause(0.02);
targetAngle6 = 180;
for i = 1:numSteps
    currentAngle6 = targetAngle6 + (70 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
pause(0.02);
%ATAS
targetAngle4 = 0;
targetAngle3 = 50;

```



```

% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
currentAngle3 = targetAngle3 - (targetAngle3 - 20) * (i / numSteps);
writePosition(servoMotor3, currentAngle3/180);
% Tunggu beberapa detik
%pause(0.1);
% Sudut target servoMotor3 saat maju
targetAngle2 = 30; % Sudut target servoMotor2 saat maju
targetAngle5 = 60;
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    currentAngle2 = targetAngle2 - (targetAngle2 - 90) * (i / numSteps);
    currentAngle5 = targetAngle5 - (targetAngle5 - 0) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2 / 180);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end

% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
%KIRI
targetAngles = linspace(65, 135, numSteps);
for i = 1: numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
pause(0.02);
%BAWAH KIRI
targetAngle3 = 20;
targetAngle5 = 0;
targetAngle2 = 90;
numSteps = 50;
for i = 1:numSteps
% Menggerakkan servo ke sudut target
    currentAngle2 = targetAngle2 - (targetAngle2 - 40) * (i / numSteps);
    currentAngle3 = targetAngle3 + (50 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle5 = targetAngle5 - (targetAngle5 - 90) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle4 = 0;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (180 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02);
end
pause(0.1); % Tunggu sebentar sebelum melanjutkan loop
targetAngle4 = 180;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (90 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
%LEPAS
targetAngle = 135;
targetAngle3 = 50;
numSteps = 50;

```

```

for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 160) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
    % Menggerakkan servo ke sudut target
    currentAngle3 = targetAngle3 + (105 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle6 = 90;
for i = 1:numSteps
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
% kembali ke posisi awal
targetAngle = 160;
targetAngle2 = 40;
targetAngle3 = 105;
targetAngle4 = 90;
targetAngle5 = 90;
targetAngle6 = 180;
numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 135) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    currentAngle2 = targetAngle2 - (targetAngle2 - 70) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    currentAngle3 = targetAngle3 + (180 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle4 = targetAngle4 + (0 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.00002); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
    % Menggerakkan servo ke sudut target
    currentAngle5 = targetAngle5 - (targetAngle5 - 170) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end

```

Sketch program untuk titik koordinat posisi keempat dengan arah bawah mencakup servo 1 (satu) yang memiliki pembacaan sudut 70°, servo 2 (dua) pada sudut 40°, servo 3 (tiga) dengan sudut 20°, servo 4 (empat) pada sudut 0°, servo 5 (lima) dengan sudut 140°, dan servo 6 (enam) pada sudut 20°.

Untuk sketch program arah titik koordinat 'BWAH'(bawah) yaitu sebagai berikut :

```

writePosition(servoMotor1, 135/180);
writePosition(servoMotor2, 70/180);
writePosition(servoMotor3, 180/180);
writePosition(servoMotor4, 0/180);
writePosition(servoMotor5, 170/180);
targetAngle6 = 90;
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    % targetAngle6 = 180;
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
end
pause(0.001); % Tunggu beberapa detik agar servo mencapai sudut awal dengan stabil

% ATAS JEPIT
targetAngle3 = 180; %60
targetAngle2 = 70; %0
targetAngle5 = 170; %100
targetAngle4 = 0; %0
targetAngle = 135;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
writePosition(servoMotor3, targetAngle3/180);
writePosition(servoMotor2, targetAngle2/180);
writePosition(servoMotor5, targetAngle5/180);
% Tunggu beberapa detik
pause(0.4);

% % Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
%kanan
targetAngles = linspace(135, 70, numSteps);
for i = 1:numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
% maju,bukagripper,jepit
targetAngle3 = 180; % Sudut target servoMotor3 saat maju
targetAngle2 = 70; % Sudut target servoMotor2 saat maju
targetAngle5 = 170; %170
targetAngle4 = 0;
% targetAngle6 = 90;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
% Tunggu beberapa detik
pause(0.4);
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    % targetAngle6 = 180;
    currentAngle3 = targetAngle3 - (targetAngle3 - 120) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3 / 180);
    pause(0.002); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
    currentAngle2 = targetAngle2 - (targetAngle2 - 40) * (i / numSteps);
    currentAngle5 = targetAngle5 - (targetAngle5 - 140) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2 / 180);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.000002); % Tunggu sebentar sebelum melanjutkan loop
end

```

```

pause(0.000002);
for i = 1:numSteps
% targetAngle6 = 180;
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.002); % Tunggu sebentar sebelum melanjutkan loop
end
pause(0.000002);
%ATAS
targetAngle4 = 0;
targetAngle3 = 120;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
% Tunggu beberapa detik
% Sudut target servoMotor3 saat maju
targetAngle2 = 40; % Sudut target servoMotor2 saat maju
targetAngle5 = 140;
for i = 1:numSteps
    currentAngle2 = targetAngle2 - (targetAngle2 - 120) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2 / 180);
    currentAngle5 = targetAngle5 - (targetAngle5 - 60) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.4/numSteps);
end
targetAngle2 = 120;
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
targetAngle5 = 60;
for i = 1:numSteps
    currentAngle3 = targetAngle3 - (targetAngle3 - 20) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3 / 180);
    currentAngle2 = targetAngle2 - (targetAngle2 - 90) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2 / 180);
    currentAngle5 = targetAngle5 - (targetAngle5 - 0) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.4/numSteps);
end
pause(0.000002);
%KIRI
targetAngles = linspace(72, 135, numSteps);
for i = 1:numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
pause(0.000002);
%BAWAH KIRI
targetAngle3 = 20;
targetAngle5 = 0;
targetAngle2 = 90;
numSteps = 50;
for i = 1:numSteps
% Menggerakkan servo ke sudut target
    currentAngle2 = targetAngle2 - (targetAngle2 - 40) * (i / numSteps);
    currentAngle3 = targetAngle3 + (50 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle5 = targetAngle5 - (targetAngle5 - 90) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle4 = 0;

```

```

for i = 1:numSteps
    currentAngle4 = targetAngle4 + (180 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02);
end
pause(0.1); % Tunggu sebentar sebelum melanjutkan loop
targetAngle4 = 180;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (90 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
%LEPAS
targetAngle = 135;
targetAngle3 = 50;
numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 160) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
    % Menggerakkan servo ke sudut target
    currentAngle3 = targetAngle3 + (105 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle6 = 90;
for i = 1:numSteps
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
% kembali ke posisi awal
targetAngle = 160;
targetAngle2 = 40;
targetAngle3 = 105;
targetAngle4 = 90;
targetAngle5 = 90;
targetAngle6 = 180;
numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 135) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    currentAngle2 = targetAngle2 - (targetAngle2 - 70) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    currentAngle3 = targetAngle3 + (180 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle4 = targetAngle4 + (0 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.00002); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
    % Menggerakkan servo ke sudut target
    currentAngle5 = targetAngle5 - (targetAngle5 - 170) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end

```

Sketch program untuk titik koordinat posisi keempat dengan arah bawah mencakup servo 1 (satu) yang memiliki pembacaan sudut 70°, servo 2 (dua) pada sudut 40°, servo 3 (tiga) dengan sudut 20°, servo 4 (empat) pada sudut 0°, servo 5 (lima) dengan sudut 140°, dan servo 6 (enam) pada sudut 20°.

Untuk sketch program arah titik koordinat 'MNDR'(mundur) yaitu sebagai berikut :

```
writePosition(servoMotor1, 135/180);
writePosition(servoMotor2, 70/180);
writePosition(servoMotor3, 180/180);
writePosition(servoMotor4, 0/180);
writePosition(servoMotor5, 170/180);
writePosition(servoMotor6, 90/180);
pause(0.1); % Tunggu beberapa detik agar servo mencapai sudut awal dengan stabil
% mundur jepit
targetAngle3 = 180; %60
targetAngle2 = 70; %0
targetAngle5 = 170; %100
targetAngle4 = 0; %0
targetAngle = 135;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
writePosition(servoMotor3, targetAngle3/180);
writePosition(servoMotor2, targetAngle2/180);
writePosition(servoMotor5, targetAngle5/180);
% Tunggu beberapa detik
pause(0.4);

% % Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
%kanan
targetAngles = linspace(135, 88, numSteps);
for i = 1:numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
% maju, bukagripper, jepit
targetAngle3 = 180; % Sudut target servoMotor3 saat maju
targetAngle2 = 70; % Sudut target servoMotor2 saat maju
targetAngle5 = 170; %170
targetAngle4 = 0;
targetAngle6 = 90;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
% Tunggu beberapa detik
pause(0.4);
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    currentAngle3 = targetAngle3 - (targetAngle3 - 120) * (i / numSteps);
```

```

currentAngle2 = targetAngle2 - (targetAngle2 - 55) * (i / numSteps);
%currentAngle5 = targetAngle5 - (targetAngle5 - 110) * (i / numSteps);
currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
writePosition(servoMotor6, currentAngle6 / 180);
writePosition(servoMotor3, currentAngle3 / 180);
writePosition(servoMotor2, currentAngle2 / 180);
%writePosition(servoMotor5, currentAngle5 / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
currentAngle5 = targetAngle5 - (targetAngle5 - 110) * (i / numSteps);
writePosition(servoMotor5, currentAngle5 / 180);
pause(0.02);
for i = 1:numSteps
% targetAngle6 = 180;
currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
writePosition(servoMotor6, currentAngle6 / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
pause(0.0002);%0.02
%ATAS
targetAngle4 = 0;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
% Tunggu beberapa detik
%pause(0.1);
% Sudut target servoMotor3 saat maju
targetAngle2 = 45; % Sudut target servoMotor2 saat maju
targetAngle5 = 110;
targetAngle3 = 130;
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
currentAngle2 = targetAngle2 - (targetAngle2 - 90) * (i / numSteps);
writePosition(servoMotor2, currentAngle2 / 180);
currentAngle5 = targetAngle5 - (targetAngle5 - 55) * (i / numSteps);
writePosition(servoMotor5, currentAngle5 / 180);
currentAngle3 = targetAngle3 - (targetAngle3 - 70) * (i / numSteps);
writePosition(servoMotor3, currentAngle3 / 180);
pause(0.0004);
end
targetAngle3 = 70;
targetAngle5 = 55;
for i = 1:numSteps
currentAngle5 = targetAngle5 - (targetAngle5 - 0) * (i / numSteps);
writePosition(servoMotor5, currentAngle5 / 180);
currentAngle3 = targetAngle3 - (targetAngle3 - 20) * (i / numSteps);
writePosition(servoMotor3, currentAngle3 / 180);
pause(0.04);
end
%kIRI
targetAngles = linspace(88, 135, numSteps);
for i = 1: numel(targetAngles)
targetAngle = targetAngles(i);
writePosition(servoMotor1, targetAngle/180);
pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
pause(0.02);
%BAWAH KIRI
targetAngle3 = 20;
targetAngle5 = 0;
targetAngle2 = 90;
numSteps = 50;

```

```

for i = 1:numSteps
% Menggerakkan servo ke sudut target
currentAngle2 = targetAngle2 - (targetAngle2 - 40) * (i / numSteps);
currentAngle3 = targetAngle3 + (50 - targetAngle3) * (i / numSteps);
writePosition(servoMotor2, currentAngle2/180);
writePosition(servoMotor3, currentAngle3/180);
currentAngle5 = targetAngle5 - (targetAngle5 - 70) * (i / numSteps);
writePosition(servoMotor5, currentAngle5 / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle4 = 0;
for i = 1:numSteps
currentAngle4 = targetAngle4 + (180 - targetAngle4) * (i / numSteps);
writePosition(servoMotor4, currentAngle4 / 180);
pause(0.02);
end
pause(0.1); % Tunggu sebentar sebelum melanjutkan loop
targetAngle4 = 180;
for i = 1:numSteps
currentAngle4 = targetAngle4 + (90 - targetAngle4) * (i / numSteps);
writePosition(servoMotor4, currentAngle4 / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
%LEPAS
targetAngle = 135;
targetAngle3 = 50;
numSteps = 50;
for i = 1:numSteps
currentAngle = targetAngle - (targetAngle - 160) * (i / numSteps);
writePosition(servoMotor1, currentAngle / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
% Menggerakkan servo ke sudut target
currentAngle3 = targetAngle3 + (105 - targetAngle3) * (i / numSteps);
writePosition(servoMotor3, currentAngle3/180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle6 = 90;
for i = 1:numSteps
currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
writePosition(servoMotor6, currentAngle6 / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
% kembali ke posisi awal
targetAngle = 160;
targetAngle2 = 40;
targetAngle3 = 105;
targetAngle4 = 90;
targetAngle5 = 70;
targetAngle6 = 180;
numSteps = 50;
for i = 1:numSteps
currentAngle = targetAngle - (targetAngle - 135) * (i / numSteps);
writePosition(servoMotor1, currentAngle / 180);
currentAngle2 = targetAngle2 - (targetAngle2 - 70) * (i / numSteps);
writePosition(servoMotor2, currentAngle2/180);
currentAngle3 = targetAngle3 + (180 - targetAngle3) * (i / numSteps);
writePosition(servoMotor3, currentAngle3/180);
currentAngle4 = targetAngle4 + (0 - targetAngle4) * (i / numSteps);
writePosition(servoMotor4, currentAngle4 / 180);
pause(0.00002); % Tunggu sebentar sebelum melanjutkan loop

```



```

end
for i = 1:numSteps
% Menggerakkan servo ke sudut target
currentAngle5 = targetAngle5 - (targetAngle5 - 170) * (i / numSteps);
writePosition(servoMotor5, currentAngle5 / 180);
currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
writePosition(servoMotor6, currentAngle6 / 180);
pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end

```

Sketch program untuk titik koordinat posisi kelima arah mundur mencakup servo 1 (satu) dengan pembacaan sudut pada program yaitu 88° , servo 2 (dua) dengan pembacaan sudut pada program yaitu 55° , servo 3 (tiga) dengan pembacaan sudut pada program yaitu 120° , servo 4 (empat) dengan pembacaan sudut pada program yaitu 0° , servo 5 (lima) dengan pembacaan sudut pada program yaitu 110° dan untuk servo 6 (enam) dengan pembacaan sudut pada program yaitu 20° .

Untuk sketch program arah titik koordinat ‘KIRI’(kiri) yaitu sebagai berikut:

```

writePosition(servoMotor1, 135/180);
writePosition(servoMotor2, 70/180);
writePosition(servoMotor3, 180/180);
writePosition(servoMotor4, 0/180);
writePosition(servoMotor5, 170/180);
writePosition(servoMotor6, 90/180);
pause(0.1); % Tunggu beberapa detik agar servo mencapai sudut awal dengan stabil
% KIRI JEPIT
targetAngle3 = 180; %60
targetAngle2 = 70; %0
targetAngle5 = 170; %100
targetAngle4 = 0; %0
targetAngle = 135;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
writePosition(servoMotor3, targetAngle3/180);
writePosition(servoMotor2, targetAngle2/180);
writePosition(servoMotor5, targetAngle5/180);
% Tunggu beberapa detik
pause(0.4); % Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
%kanan
targetAngles = linspace(135, 99, numSteps);
for i = 1:numel(targetAngles)
targetAngle = targetAngles(i);
writePosition(servoMotor1, targetAngle/180);
pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
% maju, bukagripper, jepit
targetAngle3 = 180; % Sudut target servoMotor3 saat maju

```

```

targetAngle2 = 70; % Sudut target servoMotor2 saat maju
targetAngle5 = 170; %170
targetAngle4 = 0;
targetAngle6 = 90;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
% Tunggu beberapa detik
pause(0.4);
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus

for i = 1:numSteps
    currentAngle3 = targetAngle3 - (targetAngle3 - 110) * (i / numSteps);
    currentAngle2 = targetAngle2 - (targetAngle2 - 45) * (i / numSteps);
    %currentAngle5 = targetAngle5 - (targetAngle5 - 110) * (i / numSteps);
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    writePosition(servoMotor3, currentAngle3 / 180);
    writePosition(servoMotor2, currentAngle2 / 180);
    %writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
currentAngle5 = targetAngle5 - (targetAngle5 - 110) * (i / numSteps);
writePosition(servoMotor5, currentAngle5 / 180);
pause(0.02);
for i = 1:numSteps
    % targetAngle6 = 180;
    currentAngle6 = targetAngle6 + (90 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
pause(0.0002);
%ATAS
targetAngle4 = 0;
% Menggerakkan servo ke sudut target
writePosition(servoMotor4, targetAngle4/180);
% Tunggu beberapa detik
%pause(0.1);
% Sudut target servoMotor3 saat maju
targetAngle2 = 45; % Sudut target servoMotor2 saat maju
targetAngle5 = 110;
targetAngle3 = 130;
% Menggerakkan servo ke sudut target dengan perubahan sudut yang halus
numSteps = 50; % Jumlah langkah untuk pergerakan halus
for i = 1:numSteps
    currentAngle2 = targetAngle2 - (targetAngle2 - 90) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2 / 180);
    currentAngle5 = targetAngle5 - (targetAngle5 - 55) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    currentAngle3 = targetAngle3 - (targetAngle3 - 70) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3 / 180);
    pause(0.0004);
end
targetAngle3 = 70;
targetAngle5 = 55;
for i = 1:numSteps
    currentAngle5 = targetAngle5 - (targetAngle5 - 0) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    currentAngle3 = targetAngle3 - (targetAngle3 - 20) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3 / 180);
    pause(0.0004);
end
end

```

```

%KIRI
targetAngles = linspace(99, 135, numSteps);
for i = 1:numel(targetAngles)
    targetAngle = targetAngles(i);
    writePosition(servoMotor1, targetAngle/180);
    pause(0.4/numSteps); % Dibagi dengan numSteps untuk mengurangi waktu pause
end
pause(0.02);
%BAWAH KIRI
targetAngle3 = 20;
targetAngle5 = 0;
targetAngle2 = 90;
numSteps = 50;
for i = 1:numSteps
    % Menggerakkan servo ke sudut target
    currentAngle2 = targetAngle2 - (targetAngle2 - 40) * (i / numSteps);
    currentAngle3 = targetAngle3 + (50 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor2, currentAngle2/180);
    writePosition(servoMotor3, currentAngle3/180);
    currentAngle5 = targetAngle5 - (targetAngle5 - 70) * (i / numSteps);
    writePosition(servoMotor5, currentAngle5 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle4 = 0;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (180 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02);
end
pause(0.1); % Tunggu sebentar sebelum melanjutkan loop
targetAngle4 = 180;
for i = 1:numSteps
    currentAngle4 = targetAngle4 + (90 - targetAngle4) * (i / numSteps);
    writePosition(servoMotor4, currentAngle4 / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
%LEPAS
targetAngle = 135;
targetAngle3 = 50;
numSteps = 50;
for i = 1:numSteps
    currentAngle = targetAngle - (targetAngle - 160) * (i / numSteps);
    writePosition(servoMotor1, currentAngle / 180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
for i = 1:numSteps
    % Menggerakkan servo ke sudut target
    currentAngle3 = targetAngle3 + (105 - targetAngle3) * (i / numSteps);
    writePosition(servoMotor3, currentAngle3/180);
    pause(0.02); % Tunggu sebentar sebelum melanjutkan loop
end
targetAngle6 = 90;
for i = 1:numSteps
    currentAngle6 = targetAngle6 + (180 - targetAngle6) * (i / numSteps);
    writePosition(servoMotor6, currentAngle6 / 180);
    pause(0.002); % Tunggu sebentar sebelum melanjutkan loop
end
% kembali ke posisi awal
targetAngle = 160;
targetAngle2 = 40;
targetAngle3 = 105;
targetAngle4 = 90;

```

yaitu pengujian Hasil pengujian model anfis dua kata menggunakan database. Data

hasil pengujian disajikan dalam bentuk tabel.

1. Hasil Pengujian Input Suara Langsung (model anfis 6 kata)

Tabel 4.2. Pengujian 1 Input Suara Langsung

No	Kata	Pengujian 1		Berhasil																				
		Jumlah Benar	(%)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	Kanan	15	75	x	√	x	√	x	x	x	√	√	√	√	√	√	√	√	√	√	√	√	√	√
2	Maju	16	80	√	√	√	√	√	√	√	√	√	√	x	x	√	√	√	x	x	√	√	√	√
3	Atas	9	45	x	x	√	x	x	√	√	x	x	√	√	√	√	x	x	x	x	√	x	√	√
4	Bawah	13	65	√	x	√	x	x	x	√	√	x	√	√	x	√	√	√	√	√	√	√	x	√
5	Mundur	5	25	x	x	x	x	√	√	x	√	√	x	x	x	x	x	x	x	x	x	√	x	x
6	Kiri	12	60	√	√	x	√	√	x	√	x	√	√	√	√	x	√	x	√	√	x	x	x	x

$$\text{Kanan} : \frac{15}{20} \times 100 \% = 75 \%$$

$$\text{Maju} : \frac{16}{20} \times 100 \% = 80 \%$$

$$\text{Atas} : \frac{9}{20} \times 100 \% = 45 \%$$

$$\text{Bawah} : \frac{13}{20} \times 100 \% = 65 \%$$

$$\text{Mundur} : \frac{5}{20} \times 100 \% = 25 \%$$

$$\text{Kiri} : \frac{12}{20} \times 100 \% = 60 \%$$

Hasil pengujian input suara secara langsung menunjukkan variasi tingkat keberhasilan yang berbeda pada setiap perintah. Kata seperti "Kanan" memiliki tingkat keberhasilan yang cukup baik, yaitu 75%, di mana sistem berhasil mengenali sebagian besar input suara meskipun masih ada beberapa kegagalan pada

pengujian tertentu. Hal serupa juga terlihat pada kata "Bawah" dengan tingkat keberhasilan 65% dan "Kiri" sebesar 60%, yang menunjukkan bahwa sistem dapat mengenali sebagian besar pola suara dengan baik, meskipun masih ada kendala akibat noise, perbedaan intonasi, atau inkonsistensi aksen. Namun, kata seperti "Maju" dan "Atas" memiliki akurasi yang lebih rendah dengan tingkat keberhasilan masing-masing 45% dan 25%. Penurunan ini terjadi karena variasi pola fitur MFCC yang lebih kompleks dan cenderung tumpang tindih dengan kata lainnya, sehingga menyulitkan model ANFIS dalam mengenali perintah secara akurat.

Sementara itu, kata "Mundur" memiliki tingkat keberhasilan paling rendah, yaitu 25%, yang menandakan bahwa sistem masih mengalami kesulitan dalam mengenali perintah tersebut akibat minimnya konsistensi fitur yang dapat diidentifikasi atau jumlah data latih yang belum memadai. Secara umum, hasil pengujian ini menunjukkan adanya penurunan akurasi seiring dengan bertambahnya jumlah opsi kata yang diuji. Semakin banyak opsi kata dalam sistem, semakin tinggi peluang terjadi tumpang tindih pola fitur MFCC antar perintah, yang pada akhirnya menyebabkan penurunan performa pengenalan suara. Hal ini menjadi tantangan penting dalam pengembangan sistem kendali suara, di mana optimasi model ANFIS dan peningkatan kualitas database menjadi kunci untuk meningkatkan akurasi pengenalan pada opsi kata yang lebih banyak.

2. Hasil Pengujian Input Suara dengan Database (model anfis 6 kata)

Tabel 4.3. Pengujian 2 dengan Database

No	Kata	Pengujian 2		Berhasil																				
		Jumlah Benar	(%)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	Kanan	20	100	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
2	Maju	13	65	√	√	√	x	√	√	x	√	√	x	x	√	√	x	x	x	√	√	√	√	√
3	Atas	14	70	√	x	√	√	x	√	x	x	√	√	√	x	√	√	√	√	√	√	√	x	
4	Bawah	20	100	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
5	Mundur	6	30	x	x	x	x	x	x	x	x	x	x	√	√	√	x	x	√	√	x	x	√	
6	Kiri	15	75	√	√	√	√	x	√	√	√	x	√	√	√	√	x	√	√	x	x	√	√	

$$\text{Kanan} : \frac{20}{20} \times 100 \% = 100 \%$$

$$\text{Maju} : \frac{13}{20} \times 100 \% = 65 \%$$

$$\text{Atas} : \frac{14}{20} \times 100 \% = 70 \%$$

$$\text{Bawah} : \frac{20}{20} \times 100 \% = 100 \%$$

$$\text{Mundur} : \frac{6}{20} \times 100 \% = 30 \%$$

$$\text{Kiri} : \frac{15}{20} \times 100 \% = 75 \%$$

Hasil pengujian input suara menggunakan database menunjukkan performa pengenalan yang lebih unggul dibandingkan pengujian rekam langsung, terutama pada kata "Kanan" dan "Bawah" yang mencapai tingkat keberhasilan 100%. Hal ini mengindikasikan bahwa sistem dapat mengenali kedua perintah tersebut dengan akurasi yang sangat baik, berkat pola fitur MFCC yang konsisten dan mudah dipelajari oleh model ANFIS. Sementara itu, kata "Kiri" memiliki tingkat keberhasilan 75%, diikuti oleh kata "Atas" dengan 70% dan "Maju" sebesar 65%, yang cukup baik, namun masih menunjukkan adanya beberapa kegagalan akibat

variasi intonasi, durasi suara, atau noise dalam input yang diuji. Di sisi lain, kata "Mundur" memiliki performa terendah dengan tingkat keberhasilan hanya 30%, yang menunjukkan adanya kesulitan dalam mengenali pola fitur MFCC dari perintah ini, terutama ketika data latihnya belum mencukupi atau fitur yang dihasilkan kurang konsisten.

3. Hasil Pengujian Input Suara Langsung (model anfis 2 kata)

Tabel 4.4. Pengujian 3 Input Suara Langsung

No	Kata	Pengujian 3		Berhasil																									
		Jumlah Benar	(%)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	Kanan	17	85	√	√	√	x	√	√	√	√	√	x	√	x	√	√	√	√	√	√	√	√	√	√	√	√	√	√
2	Maju	17	85	√	√	√	√	√	√	x	√	x	√	√	x	√	√	√	√	√	√	√	√	√	√	√	√	√	√
3	Atas	17	85	√	√	x	√	√	√	√	x	√	√	x	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
4	Bawah	16	80	√	x	√	√	√	√	√	√	√	√	√	x	√	√	√	x	√	x	√	√	√	√	√	√	√	√
5	Mundur	17	85	√	√	√	√	√	√	√	x	x	√	√	√	√	√	√	√	√	√	√	x	√	√	√	√	√	√
6	Kiri	16	80	√	√	√	x	x	√	√	√	√	√	√	√	x	√	√	√	√	√	√	√	√	√	√	√	√	x

$$\text{Kanan} : \frac{17}{20} \times 100 \% = 85 \%$$

$$\text{Maju} : \frac{17}{20} \times 100 \% = 85 \%$$

$$\text{Atas} : \frac{17}{20} \times 100 \% = 85 \%$$

$$\text{Bawah} : \frac{16}{20} \times 100 \% = 80 \%$$

$$\text{Mundur} : \frac{17}{20} \times 100 \% = 85 \%$$

$$\text{Kiri} : \frac{16}{20} \times 100 \% = 80 \%$$

Dari hasil pengujian, akurasi tertinggi diperoleh pada kata-kata Kanan, Maju, Atas, dan Mundur dengan tingkat keberhasilan sebesar 85%. Keberhasilan

ini menunjukkan bahwa pola fitur MFCC dari kata-kata tersebut dapat dikenali dengan baik oleh model ANFIS meskipun terdapat beberapa kegagalan. Faktor kegagalan kemungkinan besar disebabkan oleh noise pada lingkungan pengujian atau variasi intonasi yang signifikan antara pengguna.

Kata Bawah dan Kiri memiliki tingkat keberhasilan sebesar 80%, sedikit lebih rendah dibandingkan kata lainnya. Penurunan akurasi ini menunjukkan bahwa fitur MFCC dari kata-kata tersebut cenderung memiliki karakteristik yang lebih sulit dibedakan oleh model, terutama pada kondisi input suara langsung.

Secara keseluruhan, hasil pengujian ini menunjukkan bahwa model ANFIS dengan dua kata cukup andal dalam mengenali perintah suara langsung, tetapi akurasi masih dapat dipengaruhi oleh kondisi lingkungan dan jumlah opsi kata dalam sistem.

4. Hasil Pengujian Model ANFIS Dua Kata Menggunakan Database (model anfis 2 kata)

Tabel 4.5. Pengujian 4 dengan Database

No	Kata	Pengujian 4		Berhasil																				
		Jumlah Benar	(%)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	Kanan	20	100	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
2	Maju	19	95	x	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
3	Atas	19	95	√	√	√	√	√	√	√	√	√	√	√	x	√	√	√	√	√	√	√	√	√
4	Bawah	20	100	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
5	Mundur	20	100	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
6	Kiri	20	100	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

$$\text{Kanan} : \frac{20}{20} \times 100 \% = 100 \%$$

$$\text{Maju} : \frac{19}{20} \times 100 \% = 95 \%$$

$$\text{Atas} : \frac{19}{20} \times 100 \% = 95 \%$$

$$\text{Bawah} : \frac{20}{20} \times 100 \% = 100 \%$$

$$\text{Mundur} : \frac{20}{20} \times 100 \% = 100 \%$$

$$\text{Kiri} : \frac{20}{20} \times 100 \% = 100 \%$$

Pengujian menggunakan model ANFIS dengan dua kata menunjukkan peningkatan kinerja yang signifikan pada setiap perintah. Kata-kata seperti "Kanan", "Bawah", "Mundur", dan "Kiri" berhasil diidentifikasi dengan tingkat keberhasilan 100%, yang menandakan pola fitur MFCC dari kata-kata ini sangat konsisten dan dapat dipelajari dengan baik oleh model. Hasil ini mencerminkan kemampuan sistem untuk mengenali perintah-perintah dengan pola fitur yang jelas dan terpisah dari kata lainnya.

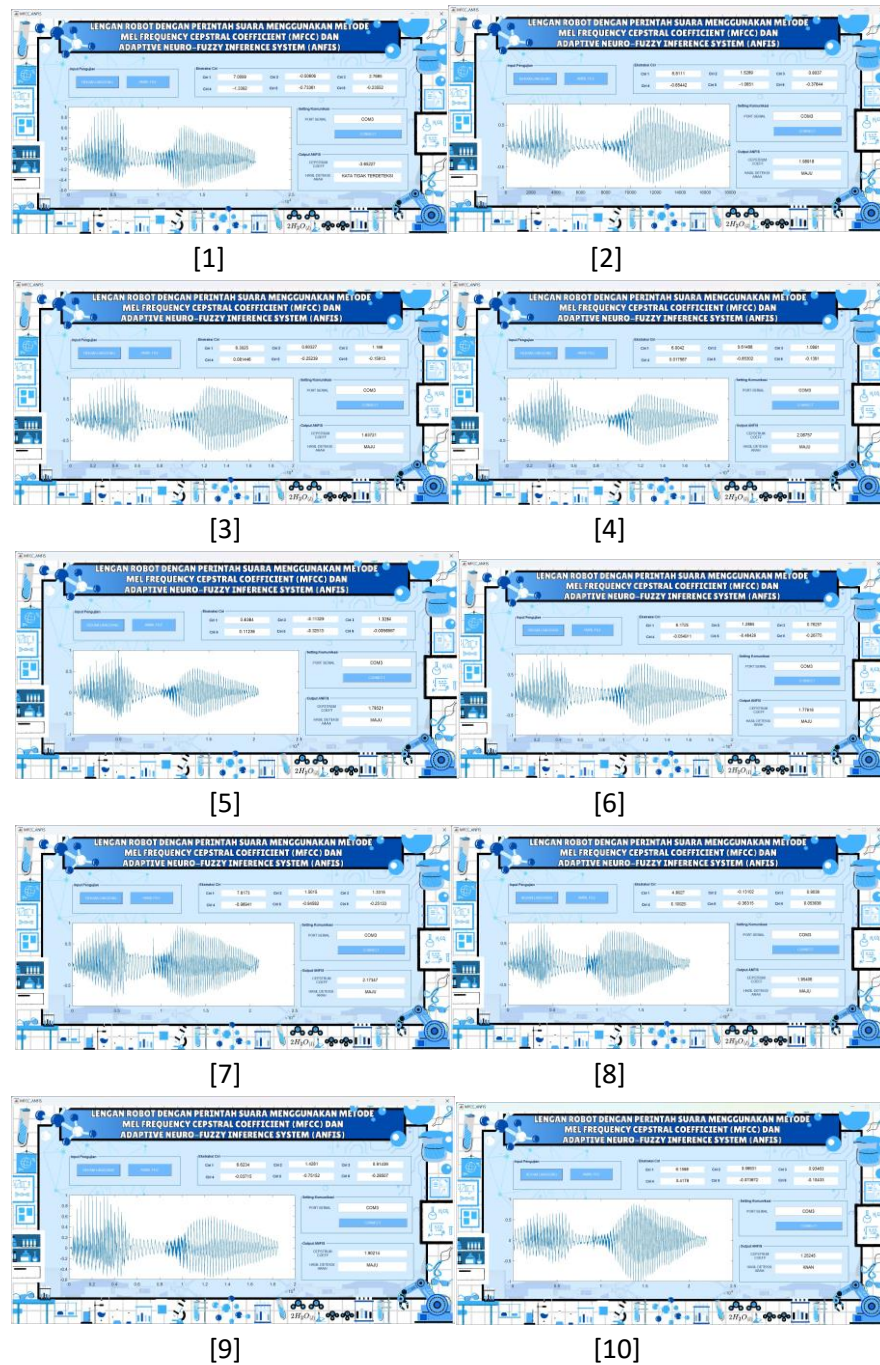
Kata "Maju" dan "Atas" menunjukkan tingkat keberhasilan masing-masing 95%, dengan hanya satu pengujian yang gagal. Kegagalan ini kemungkinan disebabkan oleh kebisingan atau variasi kecil pada intonasi suara, tetapi secara keseluruhan hasilnya sangat baik. Hasil pengujian ini menunjukkan bahwa dengan jumlah opsi kata yang lebih sedikit, akurasi sistem meningkat secara signifikan. Hal ini terjadi karena sistem hanya perlu mengenali pola fitur MFCC dari beberapa kata, sehingga potensi tumpang tindih antar kata menjadi sangat kecil.

Hasil penelitian ini menunjukkan bahwa performa sistem cenderung menurun seiring dengan meningkatnya jumlah opsi kata yang diuji. Semakin banyak opsi kata yang digunakan, semakin rumit pola fitur MFCC yang harus dipelajari oleh model, dan adanya beberapa kegagalan akibat variasi intonasi, durasi suara, atau noise dalam input yang diuji sehingga meningkatkan potensi tumpang tindih antar kata dan menyebabkan penurunan akurasi pengenalan suara. Hal ini menunjukkan perlunya optimasi lebih lanjut pada model ANFIS dan penambahan variasi database agar sistem dapat mengenali lebih banyak perintah dengan akurasi yang stabil.

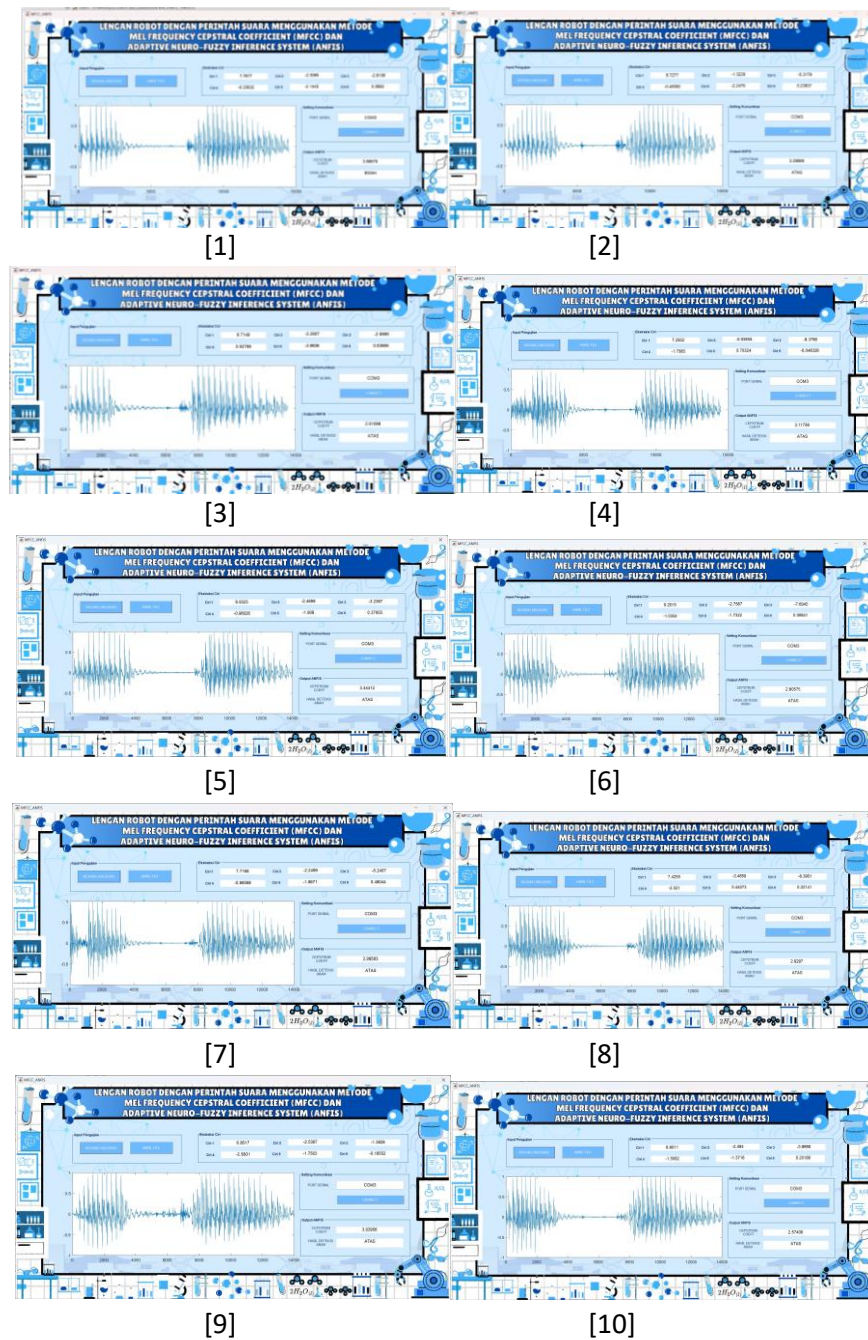
C. Analisis Sinyal Suara Lengan Robot



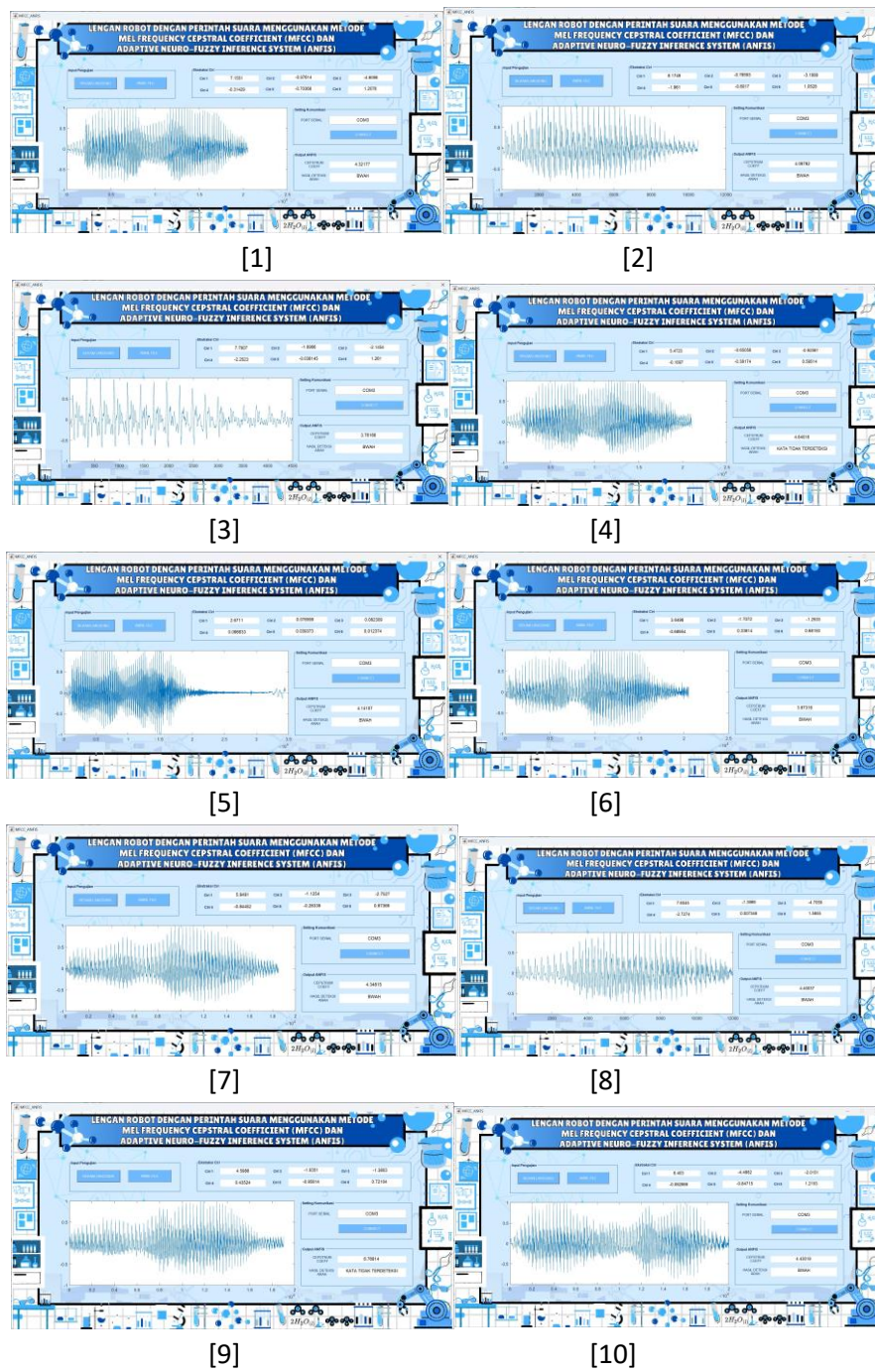
Gambar 4.5 Sinyal suara kanan(Sumber : Dokumentasi Pribadi)



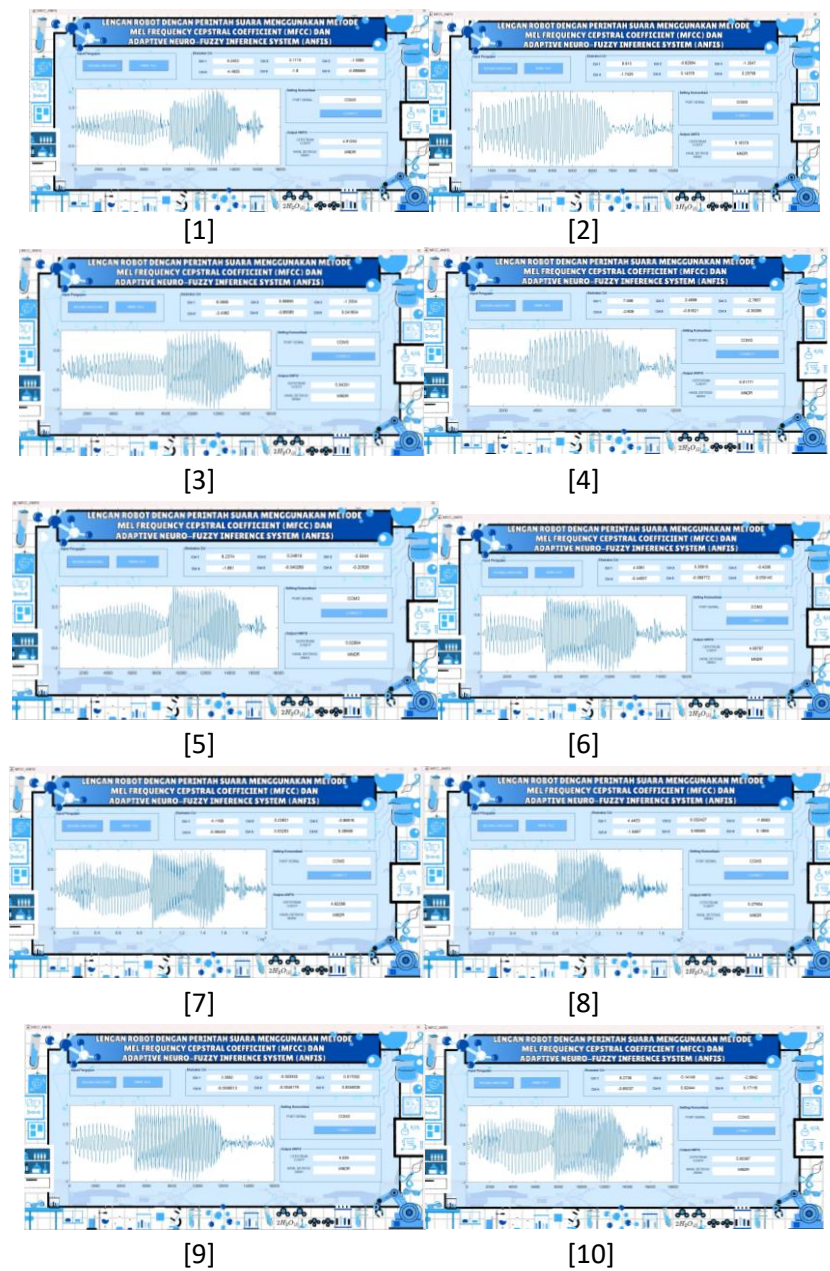
Gambar 4.6 Sinyal suara maju(Sumber : Dokumentasi Pribadi)



Gambar 4.7 Sinyal suara atas(Sumber : Dokumentasi Pribadi)



Gambar 4.8 Sinyal suara bawah(Sumber : Dokumentasi Pribadi)





Gambar 4.10 Sinyal suara kiri(Sumber : Dokumentasi Pribadi)

Tabel 4.6. Pengujian bentuk sinyal suara

No	Gambar	No urut sinyal suara	Output ANFIS		Aksi lengan robot/arah kordinat tabung reaksi
			Cepstrum coeff	Hasil deteksi arah	
1	Gambar 4.5 Sinyal suara kanan	[1]	0,462242	Kata Tidak Terdeteksi	Diam
		[2]	1,44346	KNAN	Kanan
		[3]	1,38443	KNAN	Kanan
		[4]	1,45642	KNAN	Kanan
		[5]	1,44813	KNAN	Kanan
		[6]	1,37631	KNAN	Kanan
		[7]	1,33456	KNAN	Kanan
		[8]	10,3681	Kata Tidak Terdeteksi	Diam
		[9]	1,34981	KNAN	Kanan
		[10]	1,35437	KNAN	Kanan
2	Gambar 4.6 Sinyal suara maju	[1]	-3,65227	Kata Tidak Terdeteksi	Diam
		[2]	1,98918	MAJU	Maju
		[3]	1,63721	MAJU	Maju
		[4]	2,06757	MAJU	Maju
		[5]	1,78521	MAJU	Maju
		[6]	1,77916	MAJU	Maju
		[7]	2,17347	MAJU	Maju
		[8]	1,95406	MAJU	Maju
		[9]	1,90214	MAJU	Maju
		[10]	1,25245	KNAN	Kanan

Lanjutan Tabel 4.6

No	Gambar	No urut sinyal suara	Output ANFIS		Aksi lengan robot/arah kordinat tabung reaksi
			Cepstrum coeff	Hasil deteksi arah	
3	Gambar 4.7 Sinyal suara atas	[1]	3,68979	BWAH	Bawah
		[2]	3,29069	ATAS	Atas
		[3]	2,51088	ATAS	Atas
		[4]	3,11756	ATAS	Atas
		[5]	3,44412	ATAS	Atas
		[6]	2,80575	ATAS	Atas
		[7]	2,98583	ATAS	Atas
		[8]	2,9297	ATAS	Atas
		[9]	3,03956	ATAS	Atas
		[10]	2,57409	ATAS	Atas
4	Gambar 4.8 Sinyal suara bawah	[1]	4,32177	BWAH	Bawah
		[2]	4,08782	BWAH	Bawah
		[3]	3,76166	BWAH	Bawah
		[4]	4,64016	Kata Tidak Terdeteksi	Diam
		[5]	4,14107	BWAH	Bawah
		[6]	3,67316	BWAH	Bawah
		[7]	4,34815	BWAH	Bawah
		[8]	4,40037	BWAH	Bawah
		[9]	6,76814	Kata Tidak Terdeteksi	Diam
		[10]	4,43019	BWAH	Bawah

Lanjutan Tabel 4.6

No	Gambar	No urut sinyal suara	Output ANFIS		Aksi lengan robot/arah koordinat tabung reaksi
			Cepstrum coeff	Hasil deteksi arah	
5	Gambar 4.9 Sinyal suara mundur	[1]	4,91252	MNDR	Mundr
		[2]	5,16379	MNDR	Mundr
		[3]	5,04291	MNDR	Mundr
		[4]	5,01171	MNDR	Mundr
		[5]	5,02994	MNDR	Mundr
		[6]	4,98787	MNDR	Mundr
		[7]	4,92298	MNDR	Mundr
		[8]	5,27904	MNDR	Mundr
		[9]	5,029	MNDR	Mundr
		[10]	5,00367	MNDR	Mundr
6	Gambar 4.10 Sinyal suara kiri	[1]	6,14421	KIRI	Kiri
		[2]	6,26463	KIRI	Kiri
		[3]	5,80426	KIRI	Kiri
		[4]	8,70568	Kata Tidak Terdeteksi	Diam
		[5]	6,38893	KIRI	Kiri
		[6]	5,92362	KIRI	Kiri
		[7]	6,39762	KIRI	Kiri
		[8]	6,32346	KIRI	Kiri
		[9]	5,76701	KIRI	Kiri
		[10]	6,09387	KIRI	Kiri

Berdasarkan analisis yang dilakukan pada sistem lengan robot dengan perintah suara dengan memanfaatkan metode MFCC dan ANFIS, ditemukan bahwa tingkat akurasi sistem berada di angka 80-85%. Angka ini menunjukkan kinerja yang cukup memuaskan, namun masih ada beberapa kelemahan dalam mendeteksi perintah suara. Salah satu masalah yang paling mencolok adalah adanya perintah

yang tidak teridentifikasi, yang biasanya terjadi karena nilai Cepstrum Coeff berada di luar batas klasifikasi yang sudah ditetapkan. Misalnya, untuk perintah "Kanan", nilai Cepstrum Coeff sebesar 0,462242 tidak memenuhi ambang batas klasifikasi, sehingga sistem tidak mampu mengenali perintah tersebut. Hal yang sama juga berlaku untuk perintah "Maju", di mana salah satu sampel menunjukkan nilai -3,65227, yang terlalu jauh dari batas klasifikasi yang bisa diproses oleh ANFIS, sehingga perintah tersebut dinyatakan sebagai "KATA TIDAK TERDETEKSI". Selain itu, ada juga situasi di mana nilai Cepstrum Coeff yang terlalu tinggi, seperti 8,70568 pada perintah "Kiri", menyebabkan sistem gagal dalam mengenali perintah itu. Ini menunjukkan bahwa sistem masih memiliki kelemahan terhadap variasi ekstrem dalam karakteristik suara, yang dapat dipengaruhi oleh faktor seperti intensitas suara, gangguan dari lingkungan sekitar, serta distribusi data latih yang tidak mencakup beragam variasi pelafalan.

Selain perintah yang tidak terdeteksi, analisis juga menyoroti adanya kesalahan dalam klasifikasi atau misprediction, di mana sistem keliru dalam menentukan perintah yang dimaksud karena terdapat tumpang tindih dalam rentang Cepstrum Coeff antar kategori. Sebagai contoh, untuk sinyal suara "Atas", salah satu sampelnya memiliki nilai Cepstrum Coeff = 3,68979, yang jatuh dalam rentang klasifikasi "Bawah" (3,5 - 4,5), sehingga sistem salah mengenali perintah tersebut. Hal serupa juga muncul pada perintah "Maju", di mana salah satu sampelnya memiliki Cepstrum Coeff = 1,25245, yang lebih mendekati rentang "Kanan" (0,5 - 1,5) dibandingkan rentang yang seharusnya (1,5 - 2,5), sehingga sistem menganggapnya sebagai perintah yang salah. Kesalahan tersebut menunjukkan

adanya beberapa perintah suara dengan spektrum yang mirip, yang membuat sistem sulit untuk membedakannya. Ini juga mengisyaratkan bahwa batas-batas klasifikasi yang diterapkan dalam sistem saat ini masih terlalu ketat, sehingga sedikit variasi dalam sinyal suara bisa mengakibatkan kesalahan klasifikasi.

1) Kanan

Sinyal suara "Kanan" memiliki karakteristik spektrum dengan fluktuasi amplitudo yang relatif teratur, meskipun terdapat sedikit lonjakan pada bagian awal dan akhir sinyal. Puncak frekuensi dominan berada pada rentang menengah hingga tinggi, menunjukkan kestabilan pola gelombang suara yang cukup baik. Dalam beberapa sampel, spektrum "Kanan" menunjukkan kemiripan dengan spektrum sinyal suara lain terutama pada bagian transisi suara, yang dapat menyebabkan ketidakterdeteksian oleh ANFIS. Oleh karena itu, perbedaan pengucapan yang jelas dengan intonasi stabil sangat diperlukan untuk memastikan akurasi deteksi. Ketika dikenali dengan benar, lengan robot bergerak ke arah kanan sesuai dengan koordinat yang telah ditentukan dalam sistem.

2) Maju

Sinyal suara "Maju" memiliki pola amplitudo yang naik secara bertahap tanpa lonjakan mendadak. Hal ini mencerminkan kestabilan spektral yang tinggi dengan distribusi energi yang lebih merata dibandingkan sinyal lainnya. Frekuensi dominan berada pada rentang menengah hingga rendah, dengan beberapa gelombang suara yang memperkuat kestabilan deteksi. ANFIS dapat mengenali sinyal "Maju" dengan akurasi yang lebih tinggi dibandingkan

"KNAN," karena pola suaranya lebih konsisten dan memiliki perbedaan yang cukup jelas dibandingkan kata perintah lain. Jika perintah ini dikenali dengan benar, lengan robot akan bergerak maju sesuai dengan arah koordinat yang telah ditentukan.

3) Atas

Pola spektral sinyal "Atas" menunjukkan kenaikan amplitudo yang bertahap, diikuti oleh fluktuasi kecil yang tetap dalam rentang kestabilan. Sinyal ini memiliki dominasi frekuensi menengah hingga tinggi, dengan distribusi spektral yang relatif stabil. Namun, terdapat kemiripan dengan sinyal "Bawah" dalam beberapa kasus, terutama pada bagian awal dan akhir sinyal, yang dapat menyebabkan kesalahan deteksi oleh ANFIS. Ketika dikenali dengan benar, lengan robot akan bergerak ke arah atas untuk mengambil tabung posisi koordinat atas.

4) Bawah

Sinyal suara "Bawah" memiliki amplitudo yang lebih tinggi dibandingkan sinyal lainnya, dengan fluktuasi yang lebih stabil di tengah sinyal tetapi memiliki sedikit ketidakaturan pada bagian awal dan akhir. Rentang frekuensi utama berada pada spektrum rendah hingga menengah, dengan beberapa gelombang suara yang dapat mengganggu deteksi. Kesamaan pola spektral dengan sinyal "Atas" dapat menyebabkan kesalahan klasifikasi oleh ANFIS, sehingga diperlukan optimasi lebih lanjut dalam algoritma pemrosesan suara. Jika sinyal ini dikenali dengan benar, lengan robot akan bergerak ke bawah sesuai dengan koordinat yang telah ditetapkan.

5) Mundur

Dibandingkan dengan semua sinyal lainnya, sinyal suara "Mundur" memiliki fluktuasi amplitudo yang paling stabil, dengan pola spektral yang menunjukkan gelombang suara yang kuat dan tidak mengalami lonjakan tajam. Rentang frekuensi utama berada pada spektrum menengah dengan distribusi energi yang sangat konsisten. Stabilitas ini membuat sinyal "Mundur" lebih mudah dikenali oleh ANFIS dengan akurasi tinggi, sehingga hampir tidak ditemukan kesalahan deteksi. Saat dikenali dengan benar, lengan robot akan bergerak mundur sesuai dengan koordinat yang telah ditetapkan dalam sistem.

6) Kiri

Sinyal suara "Kiri" memiliki pola spektral yang stabil dengan cepstrum coefficients yang tinggi, menunjukkan fluktuasi amplitudo yang merata dan hanya sedikit variasi pada bagian awal dan akhir sinyal. Frekuensi dominan berada pada rentang menengah hingga tinggi, dengan beberapa komponen gelombang suara pada frekuensi rendah. Meskipun cukup stabil, terdapat beberapa kasus di mana satu sampel tidak terdeteksi oleh ANFIS, kemungkinan akibat intensitas yang terlalu tinggi atau adanya noise yang mengganggu ekstraksi fitur MFCC. Jika perintah ini dikenali dengan benar, lengan robot akan bergerak ke arah kiri sesuai dengan koordinat yang telah ditentukan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada empat set data menggunakan metode Mel Frequency Cepstral Coefficient (MFCC) dan Adaptive Neuro-Fuzzy Inference System (ANFIS), dapat disimpulkan bahwa tingkat akurasi sistem lengan robot dengan perintah suara bervariasi tergantung pada model yang digunakan dan kondisi pengujian. Pengujian suara langsung dengan model ANFIS enam kata, pengujian database dengan model ANFIS enam kata, pengujian suara langsung dengan model ANFIS dua kata, dan pengujian database dengan model ANFIS dua kata. Hasil pengujian menunjukkan bahwa akurasi rata-rata sistem bervariasi secara signifikan di setiap skenario. Pengujian suara langsung dengan model enam kata menghasilkan rata-rata akurasi sebesar 58,33%, sedangkan pengujian database dengan model enam kata mencapai 70,83%. Pada pengujian suara langsung dengan model dua kata, akurasi rata-rata tercatat sebesar 83,33%, dan pengujian database model dua kata dengan suara database menghasilkan akurasi rata-rata tertinggi, yaitu 98,33%.

Lengan robot dengan perintah suara menggunakan metode MFCC dan ANFIS berhasil dirancang. Namun yang menjadi kendala terletak pada perintah suara model anfis 6 kata yang kadang pembacaan kata titik koordinat tabung reaksi kurang akurat karena pengaruh dari kata yang tumpang tindih banyaknya opsi kata yang dilatih, semakin banyak opsi kata yang digunakan maka pembacaan perintah robot akan semakin kurang baik.

B. Saran

1. Disarankan untuk menggunakan data latih suara yang lebih beragam, baik dari segi karakteristik suara (intonasi, aksen, dan frekuensi suara) maupun kondisi lingkungan (tingkat kebisingan yang berbeda). Selain itu, penting untuk memperbarui data latih secara berkala dengan menambahkan data baru dari hasil pengujian sistem. Dengan cara ini, metode ANFIS dapat terus dioptimalkan, sehingga lebih adaptif dalam mengenali setiap kata input suara dan mampu mempertahankan performa akurasi dalam berbagai kondisi operasional.
2. Penggunaan motor servo pada lengan robot disarankan menggunakan servo tipe mg90 dan mg996 atau yang menggunakan bahan metal gir.

DAFTAR PUSTAKA

- Abdullah, S. (2016). Analisis dan Perbandingan Metode Mel-Frequency Cepstral Coefficient (MFCC) dan Dynamic Time Warping (DTW) pada Identifikasi Suara. *Jurnal Elektronika dan Instrumentasi*, 5(1), 1-8.
- Chandra, R. (2015). Implementasi Algoritma Adaptive Neuro-Fuzzy Inference System (ANFIS) pada Kendali Lengan Robot Berdasarkan Perintah Suara. *Jurnal Elektronika dan Instrumentasi*, 4(3), 189-197.
- G. N. Saripan and M. N. Ismail, "Design and Implementation of Voice-Controlled Robotic Arm Using Mel-Frequency Cepstral Coefficient and Adaptive Neuro-Fuzzy Inference System," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2, pp. 53-62, 2018.
- J. J. Craig, "Introduction to Robotics: Mechanics and Control", Pearson Education, 2005.
- MathWorks, "MATLAB: The Language of Technical Computing," [Online]. Available: <https://www.mathworks.com/products/matlab.html>. [Accessed: 11-Feb-2023]
- R. A. H. Abdullah and M. N. Ismail, "Speech Recognition for Robotic Arm Control Using Mel-Frequency Cepstral Coefficient and Adaptive Neuro-Fuzzy Inference System," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 2-7, pp. 45-50, 2017.
- R. Jang, C. T. Sun, and E. Mizutani, "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence," Prentice Hall, 1997.
- R. P. Paul, "Robot Manipulators: Mathematics, Programming, and Control", MIT Press, 1981.
- Riza, I., & Suyanto, A. (2017). Penerapan Algoritma Adaptive Neuro-Fuzzy Inference System (ANFIS) untuk Prediksi Harga Saham. *Jurnal Teknik Elektro dan Komputer*, 6(2), 90-97.
- S. K. Singh, "Mechatronics: Principles and Applications", CRC Press, 2013.
- Saefullah, A., & Suyanto, A. (2015). Penerapan Algoritma Adaptive Neuro Fuzzy Inference System (ANFIS) untuk Prediksi Nilai Tukar Rupiah Terhadap Dollar Amerika Serikat. *Jurnal Teknik Elektro dan Komputer*, 4(1), 1-8.
- Waluyo, A. (2015). Sistem Kendali Lengan Robot Berdasarkan Perintah Suara Menggunakan Metode Mel-Frequency Cepstral Coefficient (MFCC) dan Algoritma Adaptive Neuro-Fuzzy Inference System (ANFIS). *Jurnal Teknik Elektro dan Informatika*, 4(2), 105-112.