IMPLEMENTASI SISTEM *VOICE RECOGNITION* PADA ROBOT PEMINDAH OBJEK SEBAGAI SISTEM NAVIGASI

SKRIPSI

Diajukan untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Teknik

Disusun Oleh:

JATRA KURNIA ARDI

NIM. 0910633055 - 63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2014

LEMBAR PERSETUJUAN

IMPLEMENTASI SISTEM *VOICE RECOGNITION* PADA ROBOT PEMINDAH OBJEK SEBAGAI SISTEM NAVIGASI

SKRIPSI

Diajukan untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Teknik

Disusun Oleh:

JATRA KURNIA ARDI

NIM. 0910633055 - 63

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Ir.Nurussa'adah,MT.	**Mochammad Rif'an, ST., MT.**
NIP. 19680706 199203 2 001	**NIP. 19710301 200012 1 001**

PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Implementasi Sistem *Voice Recognition* pada Robot Pemindah Objek sebagai Sistem Navigasi" dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Teknik dari jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Penulis menyadari bahwa tanpa bantuan, bimbingan serta dorongan dari semua pihak, penyelesaian skripsi ini tidak mungkin bisa terwujud. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

- Bapak Soehartono dan Ibu Suyati selaku orang tuadan seluruh keluarga besar penulis atas segala nasehat, kasih sayang, perhatian,dan doasehingga skripsi ini dapat terselesaikan,
- Bapak Aziz Muslim, ST., MT.,Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Hadi Suyono, ST., MT., Ph.Dselaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Mochammad Rif'an,ST.,MT. selaku Ketua Prodi Strata Satu Jurusan Teknik Elektro Universitas Brawijaya dan juga Dosen Pembimbing II penulis atas segala pengarahan ide, saran,dan bimbingan yang telah diberikan,
- Bapak Ir. Hery Purnomo selaku Dosen Pembimbing akademik atas segala bimbingan, motivasi dan masukan yang telah diberikan,
- Ibu Ir. Nurussa'adah, MT. sebagai Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijayadan juga Dosen Pembimbing I atas segala bimbingan, pengarahan, ide, nasehat, saran dan kritik yang telah diberikan,
- Bapak dan Ibu Dosen Jurusan Teknik Elektro,
- Staff Recording Jurusan Teknik Elektro,

- Sevril Renishanti atas segala kesabaran, perhatian dan *support* sehingga penulis lebih termotivasi menyelesaikan skripsi ini,
- Rekan seperjuangan dalam skripsi,Juang, Sam Budi, Eky "Sinyo", Raditya "Mbah", Wahyu "Wito", Rizal "Somad", Rafi, Risma, dan Dwisnita "Bona" terima kasih atas segala bantuan yang telah diberikan,
- Seluruh Keluarga Besar Anggota Tim Robot Kontes Robot Pemadam Api Indonesiaterima kasih atas segala semangat dan bantuan yang telah diberikan,
- Seluruh Keluarga Besar Tim Robot UB Jurusan Teknik Elektro atas segala bantuan alat, bahan dan masukan-masukannya yang telah di berikan,
- Teman-teman Ampere angkatan 2009,
- Seluruh teman-teman serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu-persatu, terima kasih banyak atas segala bentuk bantuan dan dukungannya.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Januari 2014

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Jatra Kurnia Ardi, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Januari 2014, Implementasi Sistem *Voice Recognition* pada Robot Pemindah Objek sebagai Sistem Navigasi, Dosen Pembimbing: Ir. Nurussa'adah, MT. dan Mochammad Rif'an, ST., MT.

Perkembangan ilmu pengetahuan dan teknologi khususnya di bidang robotika membawa dampak positif dalam kehidupan manusia yang pada saat ini telah sampai pada zaman *autonomous robot*. Perkembangan di bidang robotika ini juga mampu memberikan dampak positif di bidang industri. Salah satu contohnya adalah proses memindahkan sebuah barang/objek dari suatu tempat ke tempat lainnya. Penggunaan *mobile robot* yang dapat memindahkan barang/objek dapat menekan biaya produksi dan meningkatkan efisiensi waktu. Dalam kenyataannya, sering dijumpai robot pemindah barang/objek yang salah satu sistem kerjanya adalah dengan cara memberi perintah pada *mobile robot* tersebut menggunakan lebih dari satu tombol atau dalam bentuk *joystick*.

Oleh karena itu, dalam skripsi ini akan dirancang Implementasi Sistem *Voice Recognition*pada Robot Pemindah Objek sebagai Sistem Navigasi.Metode pengambilan *sample* suara pada modul *EasyVR* dilakukan sebanyak dua kali pengambilan suara dengan variasi pengucapan yang relatif sama pada setiap kata. Hal ini dilakukan sesuai dengan kemampuan *EasyVR* yang tidak bisa menerima pengucapan variasi suara kedua jika berbeda dengan variasi pengucapan suara pertama. Selain itu, *EasyVR commander* memberi batas waktu selama 5 detik untuk setiap pengucapan suara yang akan dijadikan *sample*.

Penerapan sistem *voice recognition* pada robot pemindah objek dapat digunakan sebagai sistem navigasi robot meskipun belum mampu menggantikan penggunaan perangkat berupa *joystick*.

Kata Kunci:sistem *voice recognition*, *EasyVR*, robot pemindah objek.

BAB I
PENDAHULUAN

1.1. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi khususnya di bidang robotika membawa dampak positif dalam kehidupan manusia yang pada saat ini telah sampai pada zaman *autonomous robot*.*Autonomous robot* adalah suatu robot yang mampu berperilaku secara mandiri (hanya sesekali membutuhkan perintah atau otomatis).

Autonomous robot dapat dibagi menjadi dua yaitu *autonomous stationary robot* dan *autonomous mobile robot*.*Autonomous stationary robot*adalah robot mandiri yang dirancang untuk tidak berpindah posisi atau tetap (*station*). Salah satu aplikasi *autonomous stationary robot*yang sering dijumpai adalah robot lengan yang dilengkapi manipulator (capit, bor, dll.). Sedangkan *autonomous mobile robot*adalah robot mandiri yang dirancang untuk dapat berpindah posisi (*mobile*). Salah satu aplikasi *autonomous mobile robot* adalah robot yang dapat memindahkan sebuah objek dari suatu tempat ke tempat yang lainnya.

Perkembangan di bidang robotika ini juga mampu memberikan dampak positif di bidang industri. Salah satu contohnya adalah proses memindahkan sebuah barang/objek dari suatu tempat ke tempat lainnya. Jika zaman dahulu untuk memindahkan barang/objek dari satu tempat ke tempat lainnya membutuhkan tenaga manusia yang cukup banyak, maka untuk saat ini hal tersebut dirasa kurang efisien lagi. Penggunaan *mobile robot* yang dapat memindahkan barang/objek dapat menekan biaya produksi dan meningkatkan efisiensi waktu.

Dalam kenyataannya, sering dijumpai robot pemindah barang/objek yang salah satu sistem kerjanya adalah dengan cara memberi perintah pada *mobile robot* tersebut menggunakan lebih dari satu tombol atau dalam bentuk*joystick*. Hal tersebut masih memiliki kekurangan yaitu masih adanya bantuan fisik manusia yang dapat diartikan robot tersebut masih belum bisa mandiri. Oleh karena itu, dalam skripsi ini akan dirancang Implementasi Sistem *Voice recognition* pada Robot Pemindah Objek sebagai Sistem Navigasi. Fungsi dari sistem ini adalah

untuk memberi perintah kepada robot dengan menggunakan suara manusia, sehingga manusia tersebut tidak perlu bersentuhan langsung dengan robot.

Pada penelitian yang dilakukan oleh Mohamed Fezari (2009) telah dirancang suatu sistem*voice recognition* dengan judul *New Speech Processor and Ultrasonic Sensors Based Embedded System to Improve the Control of a Motorised Wheelchair*. Namun, masih ada kekurangan dari penelitian tersebut yaitu tingkat keberhasilan menerima perintah suara dengan benar sebesar 80%, jarak yang digunakan dalam pemberian perintah hanya 5 cm sampai 20 cm, dan pengujian hanya dilakukan dengan satu jenis peritah

Perancangan ini menggunakan *mobile robot* berukuran kecil yang berfungsi sebagai *prototype* untuk mobile *robot* berukuran besar yang mampu memindahkan barang berukuran besar, misal barang-barang industri. Modul yang digunakan dalam perancangan ini adalah modul *EasyVR*yang berfungsi sebagai telinga pada robot agar mampu bernavigasi sesuai perintah yang diberikan.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat disusun rumusan masalah sebagai berikut:

1) Bagaimana merancang sistem *voice recognition* menggunakan modul *EasyVR*.
2) Bagaimana tingkat keberhasilan metode sistem *voice recognition* jika diukur dari beberapa variasi metode yang dilakukan.
3) Apakah penerapan sistem *voice recognition* mampu menggantikan penggunaan perangkat berupa *joystick*.

1.3. Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan penelitian akan diberi batasan sebagai berikut:

1) Mikrokontroler yang dipakai adalah ATMega328 dengan board *ArduinoUNO* sebagai pengendali utama.
2) Modul *voice recognition* yang digunakan adalah *EasyVR*.

3) Perintah yang digunakan adalah untuk bernavigasi maju, mundur, berhenti, hadap kanan, hadap kiri, membuka capit, dan menutup capit.
4) Robot hanya dirancang untuk simulasi sederhana tanpa harus berjalan dalam sebuah arena dan memindahkan objek.

1.4. Tujuan

Tujuan penelitian ini adalah merancang dan membuatsistem *voice recognition*sebagai sistem penggerak robot yang mampu memberi perintah berupa suara agar sebuah *mobile robot*bergerak atau bernavigasi sesuai perintah yang diberikan.

1.5. Sistematika Penulisan

Sistematika penulisan dalam penelitian ini sebagai berikut:

BAB I Pendahuluan

Memuat tentang latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika pembahasan

BAB II Tinjauan Pustaka

Membahas mengenai teori-teori yang mendukung dalam perancangan dan pembuatan alat atau sistem.

BAB III Metodologi

Berisi tentang metode-metode yang dipakai dalam melakukan perancangan, pengujian, dan analisis data.

BAB IV Perancangan

Membahas mengenai penentuan spesifikasi alat beserta fungsi dan prinsip kerjanya, perancangan diagram blok, perancangan perangkat keras, perancangan perangkat lunak dan realisasi alat.

BAB V Pengujian dan Analisis

Memuat tentang aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-hasil pengujian. Pengujian dan analisis dilakukan pada seluruh sub sistem dan sistem secara keseluruhan.

BAB VI Kesimpulan dan Saran

Memuat tentang intisari dari hasil pengujian, menjawab rumusan masalah, serta memberikan saran ataupun rekomendasi untuk perbaikan kualitas penelitian pada masa yang akan datang.

BAB II
TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penelitian ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- *Voice recognition*
- Modul*EasyVR*
- *Mobile robot* sistem Diferensial
- Motor DC Servo
- Sensor *Microphone*
- *ArduinoUNO*
- Komunikasi serial

2.1. Sistem *Voice Recognition*

Sistem *voice recognition* atau sistem pengenalan suara menurut Russ Adam (1990) adalah suatu sistem teknologi berupasuara, kata, atau frasayang diucapkan olehmanusia yang diubah menjadi sinyal listrik dan sinyal-sinyal tersebut diubah menjadipolapengkodeanyangartinyatelah ditetapkan.

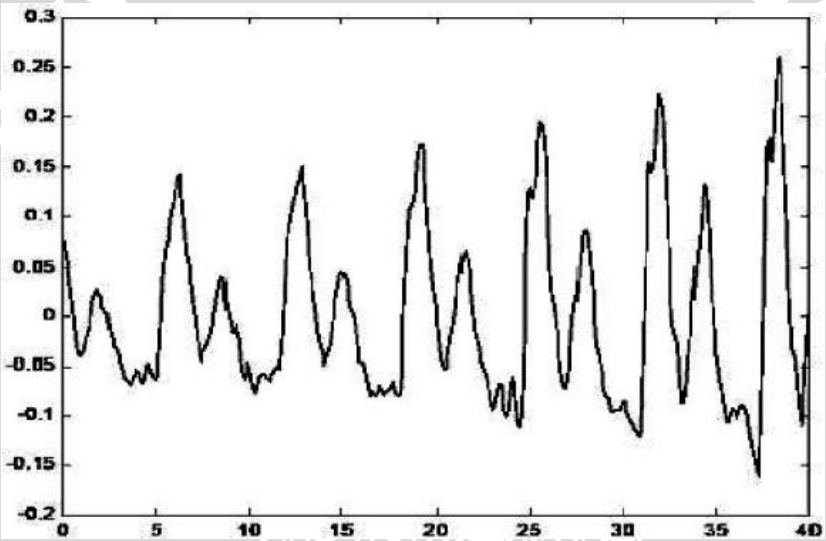
Kesulitan dalammenggunakan suarasebagaiinput untuksimulasi komputerterletak padaperbedaan mendasar antarasuara manusiadanbentuk-bentukyang lebih tradisional darimasukankomputer. Sementara programkomputerbiasanya dirancanguntuk menghasilkan responsyang tepatdanterdefinisi dengan baiksetelah menerimamasukan berupa suara manusiadan kata-katayang tepat. Setiapmanusia memiliki perbedaaan suara yang identik dan dapat memiliki arti yangberbeda jikaberbicara denganinfleksiyang berbeda.

2.2. Metode MFCC (*Mel Frequency Cepstrum Coefficient*)

MFCC (Mel Frequency Cepstrum Coefficients) merupakan salah satu metode yang banyak digunakan dalam bidang speech technology, baik speaker

recognition maupun speech recognition. Metode ini digunakan untuk melakukan feature extraction, sebuah proses yang mengkonversikan sinyal suara menjadibeberapa parameter. Beberapa keunggulan dari metode ini adalah sebagai berikut:

1) Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara, atau dengan kata lain dapat menangkap informasi-informasi penting yang terkandung dalam sinyal suara.
2) Menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang dikandungnya.
3) Mereplikasi organ pendengaran manusia dalam melakukan persepsi terhadap sinyal suara.

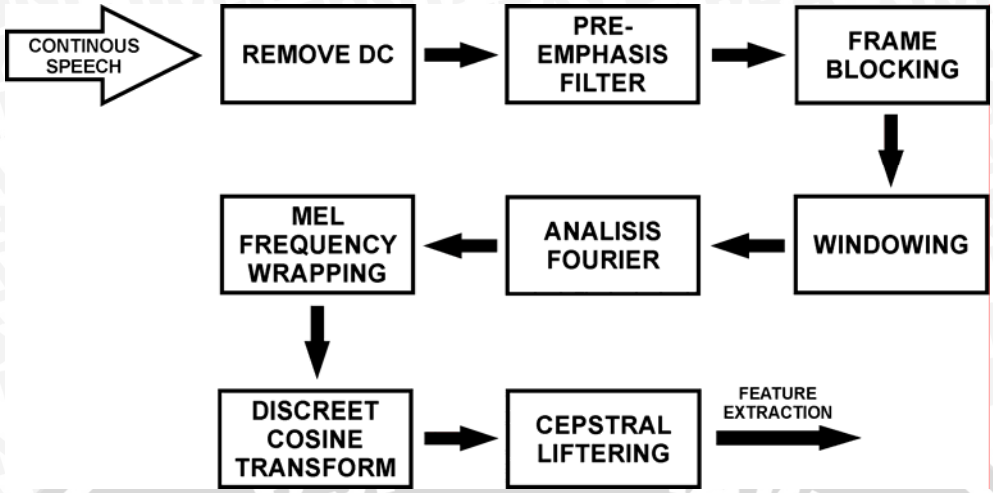


Gambar 2.1Contoh Sinyal Suara.

Sumber : Aditya, Reza, 2012

Contoh dari sinyal suara dapat dilihat pada gambar di atas. Pengujian yang dilakukan untuk periode waktu yang cukup pendek (sekitar 10 sampai 30 milidetik) akan menunjukkan karakteristik sinyal suara yang stationary.Tetapi bila dilakukan dalam periode waktu yang lebih panjang karakteristik signal suara akan terus berubah sesuai dengan kata yang diucapkan.

MFCC feature extraction sebenarnya merupakan adaptasi dari sistem pendengaran manusia, dimana signal suara akan di-filter secara linear untuk frekuensi rendah (dibawah 1000 Hz) dan secara logaritmik untuk frekuensi tinggi (diatas 1000 Hz). Gambar 2.2 menunjukkan blok diagram untuk MFCC.



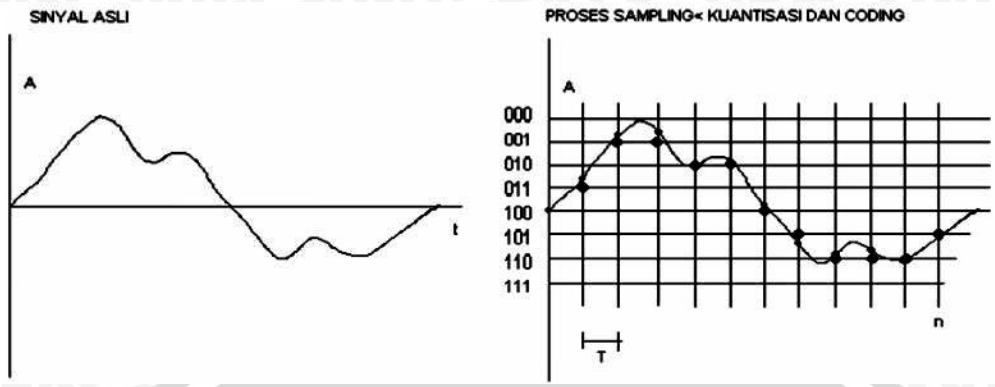
Gambar 2.2 Blok diagram MFCC.

2.2.1. Konversi Sinyal*Analog* menjadi *Digital*

Sinyal-sinyal yang natural pada umumnya seperti sinyal suara merupakan *signal continue* dimana memiliki nilai yang tidak terbatas. Sedangkan pada komputer, semua sinyal yang dapat diproses oleh komputer hanyalah *signal discrete* atau sering dikenal sebagai istilah *digital signal.* Agar sinyal natural dapat diproses oleh komputer, maka harus diubah terlebih dahulu dari data *signal continue* menjadi *discrete.* Hal itu dapat dilakukan melalui 3 proses, diantaranya adalah proses *sampling* data, proses kuantisasi dan proses pengkodean.

Proses *sampling* adalah suatu proses untuk mengambil data *signal continue* untuk setiap periode tertentu. Dalam melakukan proses *sampling* data, berlaku aturan Nyquist, yaitu bahwa frekuensi *sampling (sampling rate)* minimal harus 2 kali lebih tinggi dari frekuensi maksimum yang akan di-*sampling.* Jika *signal sampling* kurang dari 2 kali frekuensi maksimum sinyal yang akan di-*sampling,* maka akan timbul efek *aliasing. Aliasing* adalah suatu efek dimana sinyal yang dihasilkan memiliki frekuensi yang berbeda dengan sinyal aslinya.
Proses kuantisasi adalah proses untuk membulatkan nilai data ke dalam bilangan-bilangan tertentu yang telah ditentukan terlebih dahulu. Semakin banyak level yang dipakai maka semakin akurat pula data sinyal yang disimpan tetapi akan menghasilkan ukuran data besar dan proses yang lama.

Proses pengkodean adalah proses pemberian kode untuk tiap-tiap data sinyal yang telah terkuantisasi berdasarkan level yang ditempati.



Gambar 2.3Proses Pembentukan Sinyal *Digital.*

Sumber : Aditya, Reza, 2012

2.2.2. *Remove DC*

Remove DC bertujuan untuk menghitung rata-rata dari data sampel suara, dan mengurangkan nilai setiap sampel suara dengan nilai rata-rata tersebut. Tujuannya adalah mendapat normalisasi dari data suara input.

$$y[n] = x[n] - \bar{x}\,, 0 \leq n \leq N - 1 \quad \text{.....(2.1)}$$

dimana:

$y[n]$ = sampel sinyal hasil proses *remove DC.*

$x[n]$ = sampel sinyal asli.

$x[n]$ = nilai rata-rata sampel sinyal asli.

N = panjang sinyal.

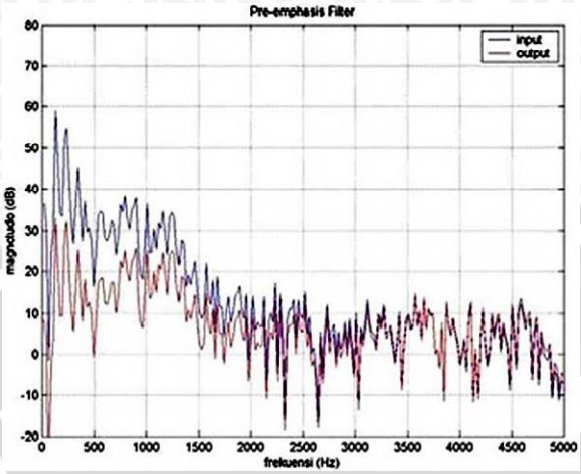
2.2.3. *Pre-Emphasis Filter*

Pre-emphasis filter merupakan salah satu jenis *filter* yang sering digunakan sebelum sebuah sinyal diproses lebih lanjut. *Filter* ini mempertahankan frekuensi-frekuensi tingi pada sebuah *spectrum*, yang umumnya tereliminasi pada saat proses produksi suara.

Tujuan dari *Pre-emphasis filter* ini adalah sebagai berikut:

1) Mengurangi *noise ratio* pada sinyal, sehingga dapat meningkatkan kualitas sinyal.
2) Menyeimbangkan spektrum dari *voice sound.* Pada saat memproduksi *voiced sound*, *glottis* manusia menghasilkan sekitar -12 dB *octave slope*. Namun ketika energi akustik tersebut dikeluarkan melalui bibir, terjadi peningkatan

sebesar +6 dB. Sehingga sinyal yang terekam oleh *microphone* adalah sekitar -6dB *octave slope*. Dampak dari ini dilihat pada gambar di bawah ini.



Gambar 2.4

Sumber : Aditya, Reza, 2012

Pada gambar diatas terlihat bahwa distribusi energi pada setiap frekuensi terlihat lebih seimbang setelah diimplementasikan *pre-emphasis filter*.Bentuk yang paling umum digunakan dalam *pre- emphasis filter* adalah sebagai berikut:

$$H(z) = 1 - \alpha z^1 \quad \text{.....(2.2)}$$

dimana $0.9 \leq \alpha \leq 1.0$, dan $\alpha \in R$. Formula diatas dapat dijadikan sebagian *first order differentiator,* sebagai berikut:

$$y[n] = s[n] - \alpha s[n-1] \quad \text{.....(2.3)}$$

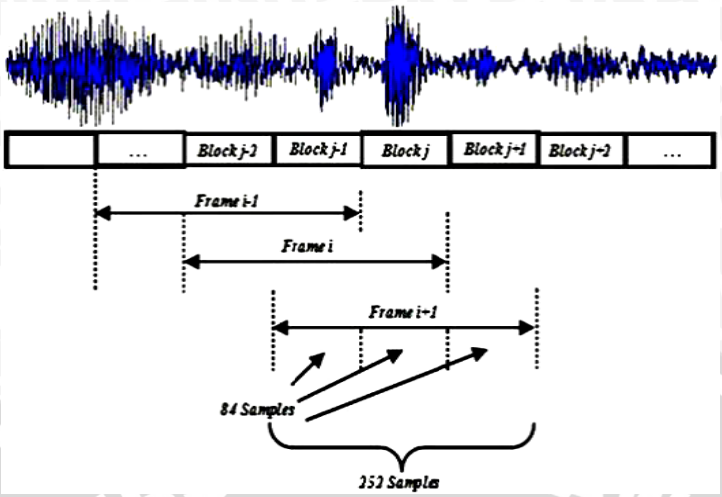
dimana:

$y[n]$ =sinyal hasil *pre-emphasis filter*.

$s[n]$ = sinyal sebelum *pre-emphasis filter*.

2.2.4. *Frame Blocking*

Karena sinyal suara terus mengalami perubahan akibat adanya pergeseran artikulasi dari organ produksi vokal , sinyal harus diproses secara *short segments (short frame).* Panjang *frame* yang biasa digunakan untuk pemrosesan sinyal adalah antara 10-30 milidetik. Panjang *frame* yang digunakan sangat mempengaruhi keberhasilan dalam analisa spektral. Di satu sisi, ukuran dari *frame* harus sepanjang mungkin untuk dapat menunjukkan frekuensi yang baik. Tetapi di lain sisi, ukuran *frame* juga harus cukup pendek untuk dapat menunjukkan resolusi waktu yang baik.



Gambar 2.5Bentuk Sinyal yang di *Frame Blocking*

Sumber : Aditya, Reza, 2012

Proses*frame* ini dilakukan terus sampai seluruh sinyal dapat diproses. Selain itu, proses ini umumnya dilakukan secara *overlapping* untuk setiap *frame*-nya. Panjang daerah *overlap* yang umum digunakan adalah kurang lebih 30% sampai 50% dari panjang *frame*. *Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame*.

2.2.5. *Windowing*

Proses *framing* dapat menyebabkan terjadinya kebocoran spektral (*spectral leakage*) atau *aliasing.Aliasing* adalah sinyal baru dimana memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate*, ataupun karena proses *frame blocking* dimana menyebabkan sinyal menjadi *discontinue*. Utuk mengurangi kemungkinan terjadinya kebocoran spektral, maka hasil dari proses *framing* harus melewati proses *window*.

Sebuah fungsi *window* yang baik harus menyempit pada bagian *main lobe* dan melebar pada bagian *side lobe*-nya.Berikut adalah representasi dari fungsi *window* terhadap sinyal suara yang diinputkan.

…….(2.4)

dimana:

= nilai sampel sinyal hasil *windowing.*

= nilai sampel dari *frame* sinyal ke i.

$w(n)$ = fungsi *window.*

N = *frame* size, merupakan kelipatan 2.

Terdapat banyak fungsi *window*, namun yang paling sering digunakan dalam aplikasi *speaker recognition* adalah *hamming window*. Fungsi *window* ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB), selain itu *noise* yang dihasilkan pun tidak terlalu besar .Fungsi *Hamming window* adalah sebagai berikut:

$$w(n) = 0{,}54 - 0{,}4 \in cos\frac{2\pi n}{M-1} \quad \text{.....(2.5)}$$

dimana:

n = 0, 1, ... , M – 1.

M = panjang *frame.*

2.2.6. Analisis *Fourier*

Analisis *fourier*adalah sebuah metode yang memungkinkan untuk melakukan analisa terhadap *spectral properties* dari sinyal yang diinputkan. Representasi dari *spectral properties* sering disebut sebagai *spectrogram.*

Dalam *spectrogram* terdapat hubungan yang sangat erat antara waktu dan frekuensi.Hubungan antara frekuensi dan waktu adalah hubungan berbanding terbalik. Bila resolusi waktu yang digunakan tinggi, maka resolusi frekuensi yang dihasilkan akan semakin rendah.

2.2.6.1. Discrete Fourier Transform (DFT)

DFT merupakan perluasan dari transformasi *fourier*yang berlaku untuk sinyal-sinyal diskrit dengan panjang yang terhingga. Semua sinyal periodik terbentuk dari gabungan sinyal-sinyal sinusoidal yang menjadi satu yang dapat dirumuskan sebagai berikut:

$$S[k] = \sum_{n=0}^{N-1} s(n)e^{-\int 2\pi n\frac{k}{N}}, 0 \le k \le N-1 \quad \text{.....(2.6)}$$

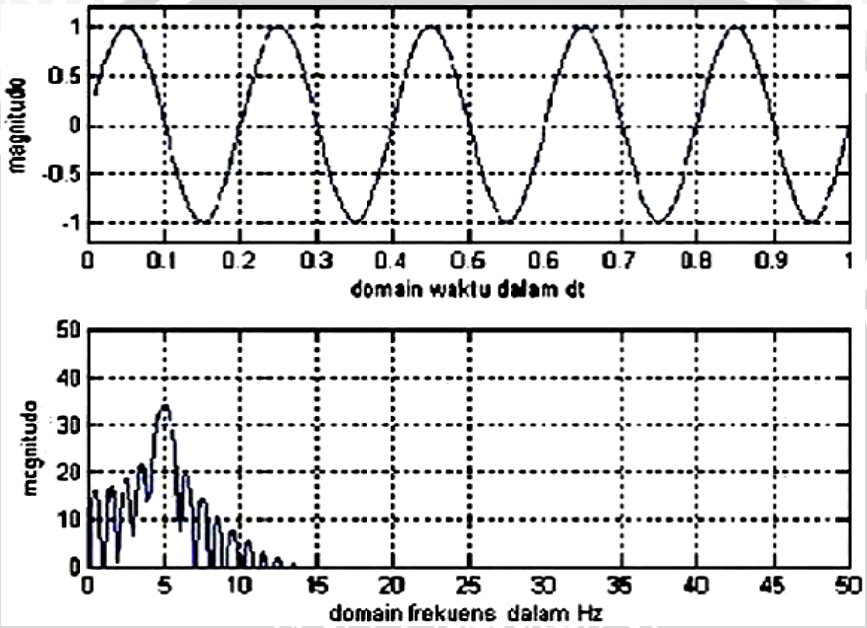
dimana:

N = jumlah sampel yang akan diproses.

$s(n)$ = nilai sampel sinyal.

k = variabel frekuensi diskrit, dimana akan bernilai$\left(k = \frac{N}{2}, k \in N\right)$.

Dengan rumus diatas, suatu sinyal suara dalam domain waktu dapat kita cari frekuensi pembentuknya. Hal inilah tujuan penggunaan analisa *fourier*pada data suara, yaitu untuk merubah data dari domain waktu menjadi data spektrum di domain frekuensi. Untuk pemrosesan sinyal suara, hal inilah sangat menguntungkan karena data pada domain frekuensi dapat diproses dengan lebih mudah dibandingkan data pada domain waktu, karena pada domain frekuensi, keras lemahnya suara tidak seberapa berpengaruh.



Gambar 2.6Domain Waktu Menjadi Domain Frekuensi

Sumber : Aditya, Reza, 2012

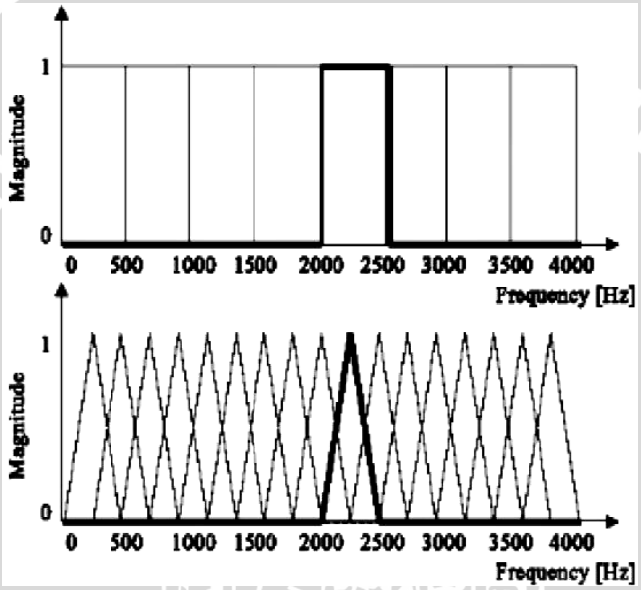
Untuk mendapatkan spektrum dari sebuah sinyal dengan DFT diperlukan N buah sampel data berurutan pada domain waktu, yaitu x[m] sampai x[m+N-1]. Data tersebut dimasukkan dalam fungsi DFT maka akan menghasilkan N buah data. Namun karena hasil dari DFT adalah simetris, maka hanya N/2 data yang diambil sebagai spektrum.

2.2.6.2. *Fast Fourier Transform* (FFT)

Perhitungan DFT secara langsung dalam komputerisasi dapat menyebabkan proses perhitungan yang sangat lama. Hal itu disebabkan karena dengan DFT, dibutuhkan perkalian bilangan kompleks. Hal itu dapat dilakukan dengan algoritma *fast fourier transform* (FFT) dimana FFT menghilangkan proses perhitungan yang kembar dalam DFT.

2.2.7. *Mel Frequency Wrapping*

Mel Freqency Wrapping umumnya dilakukan dengan menggunakan *Filterbank. Filterbank* adalah salah satu dari bentuk filter yang dilakukan dengan tujuan untuk mengetahui ukuran energi dari *frequency band* tertentu dalam sinyal suara. *Filterbank* dapat diterapkan baik dalam domain waktu maupun pada domain frekuensi, tetapi untuk keperluan MFCC, *filterbank* harus diterapkan dalam domain frekuensi.



Gambar 2.7 Magnitude dari Rectangular dan Triangul ar Filterbank

Sumber : Aditya, Reza, 2012

Filterbank menggunakan representasi konvolusi dalam melakukan *filter* terhadap sinyal. Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum sinyal dengan koefisien *filterbank.* Berikut ini adalah rumus yang digunakan dalam *filterbanks.*

……(2.7)

dimana:

N = jumlah *magnitude spectrum.*

S[*j*] = *magnitude spectrum* pada frekuensi j.

= koefisien *filterbank* pada frekuensi j ($1 \leq i \leq M$).

M = jumlah *channel* dalam *filterbank.*

Presepsi manusia terhadap frekuensi dari sinyal suara tidak mengikuti *linier scale*. Frekuensi yang sebenarnya (dalam Hz) dalam sebuah sinyal akan diukur manusia secara subyektif dengan menggunakan *mel scale*. *Mel frequency scale* adalah *linier* frekuensi *scale* pada frekuensi dibawah 1000 Hz dan merupakan *logarithmic scale* pada frekuensi diatas 1000 Hz.

2.2.8. *Discrete Cosine Transform* (DCT)

DCT merupakan langkah terakhir dari proses utama MFCC *feature extraction*. Konsep dasar dari DCT adalah mendekorelasikan *mel spectrum* sehingga menghasilkan representasi yang baik dari properti spektral lokal. Pada dasarnya konsep DCT sama dengan *inverse fourier transform.* Namun hasil dari DCT mendekati PCA (*principle component analysis*).PCA adalah metode statik klasik yang digunakan secara luas dalam analisa data dan kompresi. Hal inilah yang menyebabkan seringkali DCT menggantikan *inverse fourier transform* dalam proses MFCC *feature extraction*. Berikut adalah formula yang digunakan untuk menghitung DCT.

$$C_n = \sum_{k-1}^{K} (\log S_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right]; n = 1, 2, \ldots, K \ldots\ldots\ldots (2.8)$$

dimana:

S_k = keluar dari proses *filterbank* pada *index* k.

K = jumlah koefisien yang diharapkan.

Koefisien ke nol dari DCT pada umumnya akan dihilangkan, walaupun sebenarnya mengindikasikan energi dari *frame* sinyal tersebut. Hal dilakukan karena, berdasarkan penelitian-penelitian yang pernah dilakukan, koefisien ke nol ini tidak *reliable* terhadap *speaker recognition*.

2.2.9. Cepstral Liftering

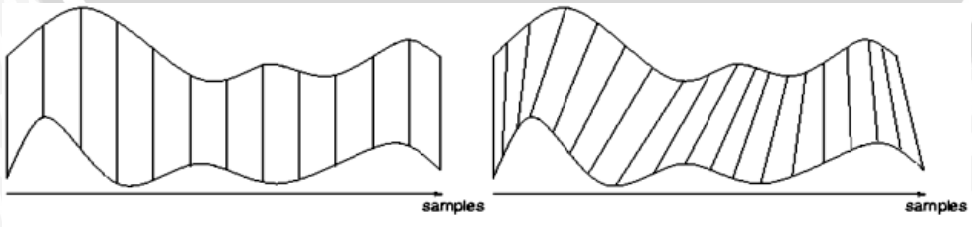
Hasil dari proses utama MFCC *feature extraction* memiliki beberapa kelemahan. *Low order* dari *cepstral coefficients* sangat sensitif terhadap *spectral slope*, sedangkan bagian *high order*-nya sangat sensitif terhadap *noise*. Oleh karena itu, *cepstral liftering* menjadi salah satu standar teknik yang diterapkan untuk meminimalisasi sensitifitas tersebut.*Cepstral liftering* dapat dilakukan dengan mengimplementasikan fungsi *window* terhadap *cepstral features.*

Cepstral liftering menghaluskan spektrum hasil dari *main processor* sehingga dapat digunakan lebih baik untuk *pattern matching*.

2.3. Metode DTW (Dynamic Time Warping)

Satu masalah yang cukup rumit dalam pengenalan wicara adalah poses perekaman yang terjadi seringkali berbeda durasinya, biarpun kata atau kalimat yang diucapkan sama. Bahkan untuk satu suku kata yang sama atau vokal yang sama seringkali proses perekaman terjadi dalam durasi yang berbeda. Sebagai akibatnya proses *matching* antara sinyal uji dengan sinyal referensi (*template*) seringkali tidak menghasilkan nilai yang optimal.

Sebuah teknik yang cukup populer di awal perkembangan teknologi pengolahan sinyal wicara adalah dengan memanfaatkan sebuah teknik *dynamic-programming* yang juga lebih dikenal sebagai *Dynamic Time Warping* (DTW). DTW (*Dynamic Time Warping*) adalah metode untuk menghitung jarak antara dua data *time series*. Keunggulan DTW dari metode jarak yang lainnya adalah mampu menghitung jarak dari dua vektor data dengan panjang berbeda.Jarak DTW diantara dua vektor dihitung dari jalur pembengkokkan optimal (*optimal warping path*) dari dua vektor tersebut.Prinsip dasarnya adalah dengan memberikan sebuah rentang "*steps*□ dalam ruang (dalam hal ini sebuah *frame-frame* waktu dalam *sample, frame-frame* waktu dalam *template*) dan digunakan untuk mempertemukan lintasan yang menunjukkan *local match* terbesar (kemiripan) antara *time frame* yang lurus. Total *similarity cost* yang diperoleh dengan *algorithm* ini merupakan sebuah indikasi seberapa bagus *sample* dan *template* ini memiliki kesamaan, yang selanjutnya akan dipilih *best-matching template*.Ilustrasi pencocokan dengan metode DTW ditunjukkan pada gambar dibawah ini.



Gambar 2.10 Pencocokan *Sequence* (a) *Alignment* Asli dari 2 *Sequence* (b) Alignment dengan DTW

Sumber : Aditya, Reza, 2012

Dari beberapa teknik yang digunakan untuk menghitung DTW, salah satu yang paling handal adalah dengan metode pemrograman dinamis. Jarak DTW dapat dihitung dengan rumus:

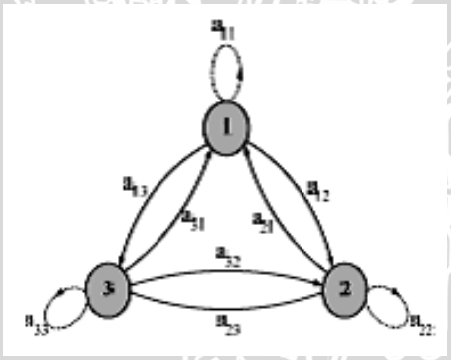
……(2.9)

2.4. Metode HMM (*Hidden Markov Model*)

HMM adalah analisis statistika yang memodelkan sinyal suara dan mencari bentuk kata yang paling sesuai.HMM berkembang dengan sangat cepat karena pemodelan ini sangat kaya dalam struktur matematika dan mengacu pada fungsi probabilitas rantai markov.

2.4.1. Rantai Markov

Algoritma HMM didasari oleh model matematik yang dikenal dengan rantai markov.Rantai markov secara umum ditunjukkan pada Gambar 2.



Gambar 2.8

Sumber : Hidayatno, Ahmad

Beberapa hal yang dapat dijelaskan tentang rantai markov yaitu:

- Transisi keadaan dari suatu keadaan tergantung pada keadaan sebelumnya.

  $P[q_t = j|q_{t-1} = i, q_{t-2} = k.....] = P[q_t = j \mid q_{t-1} = i]$……(2.10)

- Transisi keadaan bebas terhadap waktu.

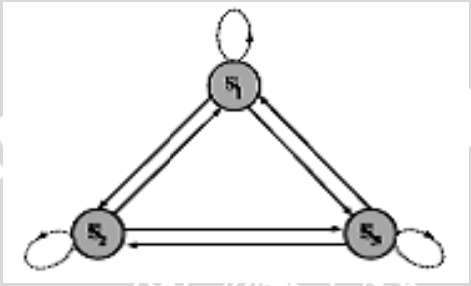
  $a_{ij} = P[q_t = j \mid q_{t-1} = i\ ]$……..…(2.11)

2.4.2. Tipe HMM

HMM dibagi menjadi dua tipe dasar yaitu HMM ergodic dan HMM Kiri-Kanan.

1) HMM *ergodic*.

Pada HMM model *ergodic* perpindahan keaadaan satu ke keadaan yang lainsemuanya memungkinkan, hal ini ditunjukkan dalam Gambar 3.

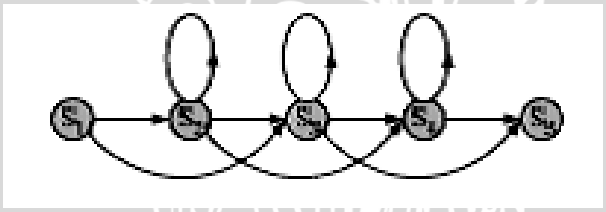


Gambar 2.9 HMM model *ergodic*.

Sumber : Hidayatno, Ahmad

2) HMM Kiri-kanan

Pada HMM kiri-kanan perpindahan keadaan hanya dapat berpindah dari kirikekanan, perpindahan keadaan tidak dapat mundur ke belakang, hal iniditunjukkan dalam Gambar 4.



Gambar 2.10 HMM model kiri-kanan.

Sumber : Hidayatno, Ahmad.

2.4.3. Elemen HMM

Elemen yang terdapat pada HMM yaitu :

a) N, jumlah keadaan (state) dalam model.

b) M, jumlah simbol observasi yang berbeda tiap keadaan.

c) Distribusi keadaan transisi A={aij} dengan:

d) Distribusi probabilitas simbol observasi.

…..(2.12)

e) Distribusi keadaan awal.

$$\pi_i = P[q_t = i]\,; \quad 1 \leq i \leq N \text{..................................(2.13)}$$

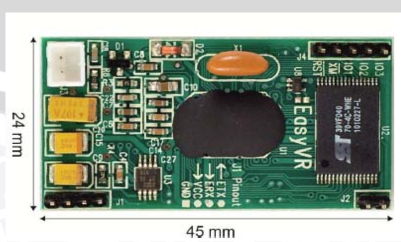
2.5. Modul *EasyVR*

EasyVR merupakan modul*voice recognition* multi-fungsi yang dapat digunakan padabanyak aplikasi pengontrolan yang membutuhkan pendeteksian suara dan percakapan.Modul ini dapat digunakan/dihubungkan dengan board mikrokontroler *Arduino*. Cocok digunakan untuk beragam aplikasi, seperti home automation (dimana kita dapat mengontrol nyala lampu, kunci pintu, televisi, atau perangkat lainnya) atau sebagai modul pelengkap sensor pendengaran robot yang dibuat sebagaimana robot-robot canggih yang dijual di pasaran yang harganya luar biasa mahal.

Secara umum, fitur dari *EasyVR* adalah sebagai berikut:

- Mendukung beberapa bahasa, yaitu:English(US), Italian, German, French, Spanish, Japanese.
- Mendukung hingga 32 *custom Speaker Dependet* (SD) trigger atau perintah, bahkan dapat digunakan pada bahasa apapun.
- GUI yang mudah digunakan.
- x GPIO (IO1, IO2, IO3) dapat dikontrol dengan perintah protokol baru.
- PWM audio output mendukung speaker 8 ohm.

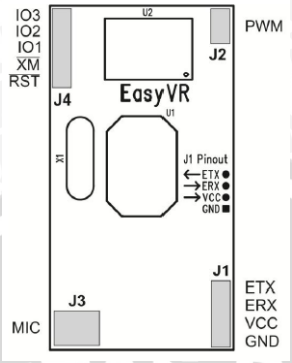
Modul*EasyVR*dapat digunakan denganantarmukaUART yang didukungpada rentang tegangan 3.3V-5V, sepertiPIC dan board *Arduino*. Beberapacontoh aplikasitermasukotomatisasi rumah, seperti suara yang mengendalikanswitch pada lampu, kunci, atau penambahan"pendengaran" untuk robot yang saat ini sedang berkembang. Bentuk fisik *EasyVR* ditunjukkan dalam Gambar 2.1.



Gambar 2.11 Bentuk fisik *EasyVR*.

Sumber : Datasheet *EasyVR*

2.5.1. Fungsi Pin pada *EasyVR*



Gambar 2.12 Konfigurasi pin pada *EasyVR.*
Sumber : Datasheet *EasyVR*

Konfigurasi pin pada *EasyVR* yang ditunjukkan dalam Gambar 6.10 dan fungsi dari masing-masing pin ditunjukkan dalam Tabel 2.1.

Tabel 2.1 Fungsi pin *EasyVR.*

KONEKTOR	NOMOR	NAMA	TIPE	DESKRIPSI
J1	1	GND	-	Ground
	2	VCC	I	Sumber tegangan DC input
	3	ERX	I	Serial Data Receive
	4	ETX	O	Serial Data Tranceive
J2	1 – 2	PWM	O	Diferensial audio output (mampu speaker 8 ohm)
J3	1	MIC_RET	-	Ground pada microphone
	2	MIC_IN	I	Sinyal input pada microphone
J4	1	RST	I	Reset (logika LOW)
	2	XM	I	Boot select
	3	IO1	I/O	General purpose I/O (3.0 VDC TTL level)
	4	IO2	I/O	General purpose I/O (3.0 VDC TTL level)
	5	IO3	I/O	General purpose I/O (3.0 VDC TTL level)

Sumber: Datasheet *EasyVR*

2.5.2. Karakteristik Elektrik *EasyVR*

Karakteristik elektrik *EasyVR* ditunjukkan dalam Tabel 2.2.

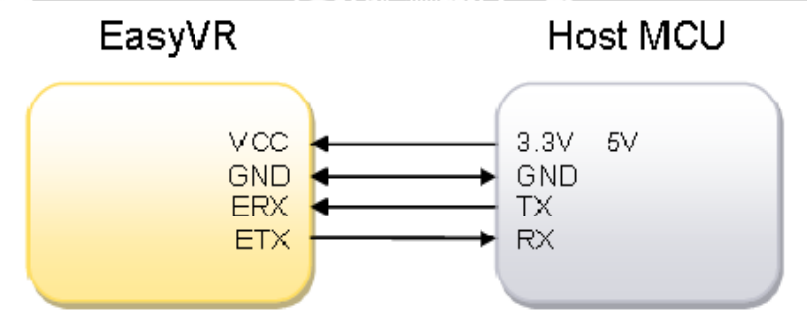
Tabel 2.2 Karakteristik elektrik *EasyVR*

Symbol	Parameter		Min	Typ	Max	Unit
V_{IH}	Input High Voltage		2.4	3.0	3.3	V
V_{IL}	Input Low Voltage		-0.1	0.0	0.75	V
I_{IL}	Input Leakage Current (0 < V_{IO} < 3V, Hi-Z Input)			<1	10	µA
R_{PU}	Pull-up Resistance	Strong		10		kΩ
		Weak		200		kΩ
V_{OH}	Output High Voltage (I_{OH} = -5 mA)		2.4			V
V_{OL}	Output Low Voltage (I_{OL} = 8 mA)				0.6	V

Sumber : Datasheet *EasyVR*

2.5.3. *Communication Protocol*

Sistem komunikasi padamodul*EasyVR*menggunakanantarmukaUART dengan standarkompatibel 3,3-5V TTL/CMOSsesuai denganteganganVCC.



Gambar 2.13 Protokol komunikasi pada modul *EasyVR*.

Sumber: Datasheet *EasyVR*

Konfigurasi awaldipower onadalah9600baud, 8 bit data, no parity, 1bit stop. Baud rate dapat diubah untuk beroperasidalam kisaran9600-115200baud.

Protokol komunikasihanya menggunakankarakterASCII yang dapat dicetak, yang dapat dibagidalam dua kelompok utama:

- *Command*dan statuskarakter, masing-masing padaTXdanRX, dipilih di antarahuruf kecil.
- Argumen perintahataurincianstatus, padaTXdanRX, mencakupberbagaihuruf kapital.

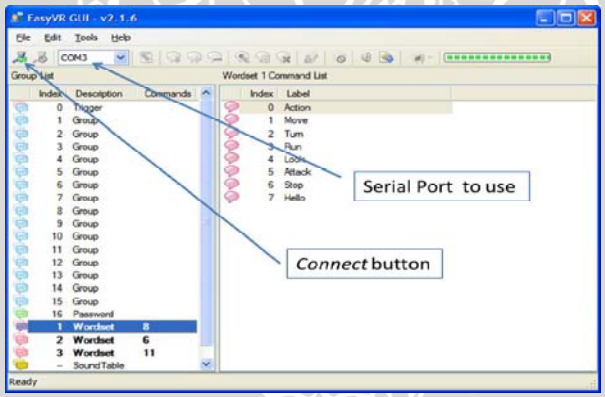
Setiapperintah yang dikirimpada TX, dengannol atau lebihargumentambahan, menerimajawabanpada RXdalambentukstatus byte yang diikuti olehnol atau lebihargumen. Terdapat penundaanminimumsebelum

setiapbytedikirim keluar darimodul*EasyVR*ke RX yang awalnyaditetapkan 20msdan dapat dipilihkemudian dalamrentang0 - 9ms, 10-90ms, 100ms-1s. Karena pada antarmuka serial*EasyVR*jugaberbasis software, penundaan yang sangatpendekmungkin diperlukansebelum mengirimkankarakter kemodul, terutama jikahost bekerja sangatcepat, sehingga memungkinkan*EasyVR*untuk kembalimendengarkankarakter baru.

2.5.4. *EasyVRCommander*

EasyVRCommander adalah software yang digunakan untukmengkonfigurasimodul*EasyVR*yang terhubung ke dengan menggunakanmikrokontroleryang menyediakan program "bridge".

Pengguna dapatmenentukan kelompokperintah ataupassword danmenghasilkankode templateuntuk menangani mereka. Hal ini diperlukanuntuk mengeditkode yang dihasilkanuntuk mengimplementasikan aplikasi logika. Templateberisi semua fungsiatausubrutinuntuk menanganitugas-tugaspengenalan suara. Tampilan software *EasyVRCommander* ditunjukkan dalam Gambar 2.4.



Gambar 2.14 Tampilan pada *EasyVRCommander*.

Sumber : Datasheet *EasyVR*

Terdapat empat jenis perintah dalam *EasyVRCommander*, yaitu:

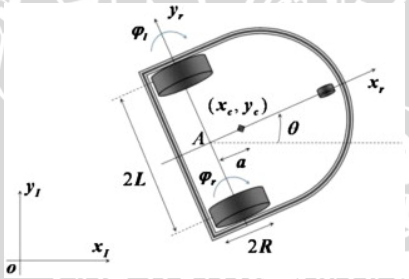
- Trigger : kelompokkhusus di manaAnda memilikibuilt-in memicukataSI "Robot" dan Andadapat menambahkan satuuser-definedmemicu kataSD. Kata-kata pemicudigunakan untukmemulaiproses pengenalan.
- Group : untuk menambahkankanperintahSDuser-defined.
- Password :kelompok khususuntuk"password vokal" (sampai lima),

menggunakanSpeaker Verifikasi(SV) teknologi.

- Wordset : built-insetperintahSI(misalnya pada Gambar1di atas,Wordset 1 dipilih).

2.6. Mobile Robot Sistem Diferensial

Robot mobil sistem diferensial merupakan robot mobil yang bergerak dengan menggunakan dua buah roda yang dapat digerakkan secara terpisah.Roda diletakkan pada kedua sisi robot.Robot dapat merubah arah pergerakannya dengan mengatur putaran pada kedua roda. Kedua roda ini berfungsi sebagai penggerak sekaligus sebagai kemudi robot mobil.Oleh karena itu pada robot ini tidak diperlukan pengendali arah gerakan tambahan. Sehingga dalam sistem *differential drive* dibutuhkan minimal dua buah motor sebagai penggerak roda-roda tersebut. Dengan mengkombinasikan kecepatan putaran motor-motornya, robot dapat dikemudikan lurus, membentuk lengkungan (*curve*), dan juga berputar (*rotation*) di tempat. *Mobile robot* sistem *differntial drive* dan koordinat referensi ditunjukkan dalam Gambar 2.5.



Gambar 2.15 *Mobile robot* Tipe *Differntial Drive* dan Koordinat Referensi.
Sumber : http://www.sciencedirect.com

Untuk menganalisa kinematika *mobile robot* dengan sistem *differential drive,* maka robot dimodelkan menjadi suatu rangka statis di atas roda, yang mana robot akan beroperasi pada bidang horizontal. Pada bidang tersebut rangka robot memiliki tiga dimensi yaitu dua posisi dalam bidang dan satu untuk orientasi sepanjang sumbu vertikal yang ortogonal terhadap bidang. Kenyataannya tentu akan ditemui banyak derajat kebebasan dan fleksibilitas karena as roda, *wheel steering joints*, serta *wheel castor joints*. Namun dalam hal ini rangka robot yang dimodelkan hanya mengacu pada rangka statis robot itu sendiri, sehingga dapat

dikatakan mengabaikan sendi (*joints*) dan derajat kebebasan (*degree of freedom*) internal terhadap robot dan roda-rodanya.

2.7. Motor DC *Brushed*

Prinsip kerja motor DC *brushed* sesuai dengan hukum kemagnetan Lorenz, yaitu membangkitkan fluksi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi.

Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

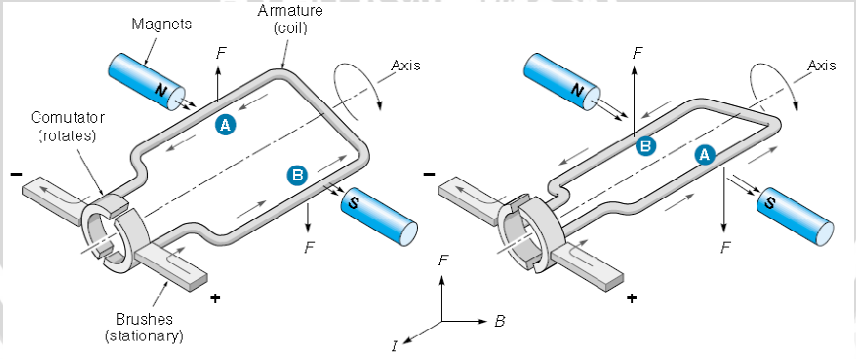
Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah gaya putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

$$F = B.I.L \text{ (Newton)} \qquad (2.1)$$

Dimana : B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

I = arus listrik (ampere)



Gambar 2.16Ilustrasi Motor DC Brushed

Sumber : Kilian, 2002

Ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a–b berada di dalam medan magnet ditunjukkan dalam Gambar 7.10. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*comutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

Bagian *comutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Arah fluk magnetik adalah dari kiri ke kanan ditunjukkan dalam Gambar 7.3. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah . Gaya-gaya yang bekerja sama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar a dan penghantar b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

2.8. Motor DC Servo

Servo berasal dari nama latin ”servus” yang berarti budak. Servo adalah sebuah sistem kendali otomatis yang dipakai untuk mengatur kecepatan sebuah motor DC. Sehingga kecepatan motor akan konstan sampai kapan pun karena adanya servo ini. Servo menggunakan sistem umpan balik. Sensor untuk mendeteksi putaran motor terhubung ke rangkaian umpan balik, kemudian sistem akan menghitung kesalahan dan memutuskan untuk mempercepat atau memperlambat putaran motor. Motor servo adalah gabungan antara motor DC yang telah dilengkapi dengan rangkaian umpan balik.

2.8.1. Teori Motor Servo

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan

memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Motor Servo ditunjukkan dalam Gambar 2.6.

Gambar 2.17 Motor Servo
Sumber : gwsus.com

Motor Servo merupakan sebuah motor DC yang memiliki rangkaian control elektronik dan internal gear untuk mengendalikan pergerakan dan sudut angularnya. Sistem Mekanik Motor Servo ditunjukkan dalam Gambar 2.7.

Gambar 2.18 Sistem Mekanik Motor Servo
Sumber : gwsus.com

Motor servo adalah motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena internal gearnya. Lebih dalam dapat digambarkan bahwa sebuah motor servo memiliki :

1) 3 jalur kabel : power, ground, dan control
2) Sinyal control mengendalikan posisi
3) Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum.

4) Konstruksi didalamnya meliputi internal gear, potensiometer, dan feedback control.

2.8.2. Jenis-Jenis dan Kegunaan Motor Servo :

1) Motor Servo Standar 180°

Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°.

2) Motor Servo *Continuous*

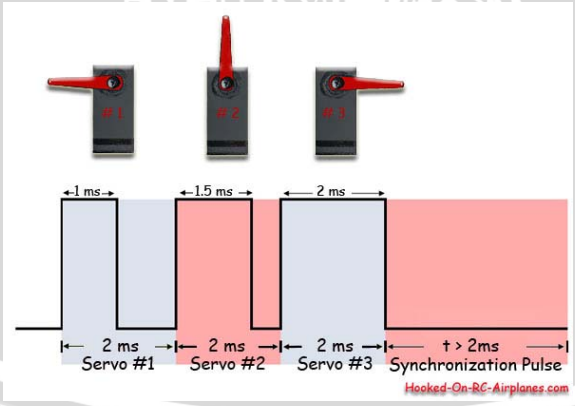
Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

Kebanyakan motor servo digunakan sebagai :

1) Manipulator.
2) *Moving camera's*.
3) *Robot arms*.

2.8.3. Mode Pemberian Pulsa pada Motor Servo

Sudut dari motor servo tergantung dari besarnya pulsa yang akan diterima oleh motor servo. Mode pemberian pulsa dapat ditunjukkan dalam Gambar 2.8.

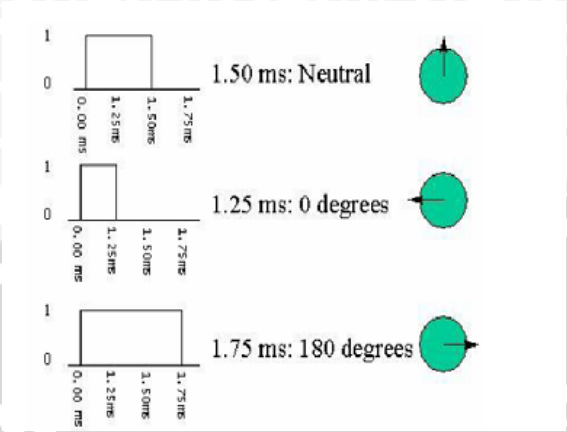


Gambar 2.19 Mode pemberian pulsa

Sumber : Hooked-On-RC-Airplanes.com

Contoh dimana bila diberikan pulsa dengan besar 1.5ms mencapai gerakan 90 derajat, maka bila kita berikan data kurang dari 1.5 ms maka posisi mendekati 0 derajat dan bila kita berikan data lebih dari 1.5 ms maka posisi

mendekati 180 derajat. Contoh posisi dan waktu pemberian pulsa ditunjukkan dalam Gambar 2.9.



Gambar 2.20 Contoh posisi dan waktu pemberian pulsa

Sumber : Riyanto Sigit, 2007

1) Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50Hz.
2) Dimana pada saat sinyal dengan frekuensi 50Hz tersebut dicapai pada kondisi Ton *duty cycle* 1.5ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0° / netral).
3) Pada saat Ton *duty cycle* dari sinyal yang diberikan kurang dari 1.5ms, maka rotor akan berputar ke arah kiri dengan membentuk sudut yang besarnya linier terhadap besarnya Ton *duty cycle*, dan akan bertahan diposisi tersebut.
4) Dan sebaliknya, jika Ton *duty cycle* dari sinyal yang diberikan lebih dari 1.5ms, maka rotor akan berputar ke arah kanan dengan membentuk sudut yang linier pula terhadap besarnya Ton *duty cycle*, dan bertahan diposisi tersebut.

2.9. Sensor *Microphone*

Microphone adalah sebuah transduser sinyal akustik ke sinyal listrik atau sensor yang mengubah sinyal suara menjadi sinyal listrik. Microphone dapat digunakan dalam berbagai aplikasi seperti telepon, tape recorder, karaoke, alat bantu dengar, teknik live audio dan rekaman, radio FRS, megafon, penyiaran

televisi dan komputer untuk merekam suara, pengenalan suara, dan untuk tujuan non-akustik seperti pemeriksaan ultrasonik atau sensor ketukan.

2.9.1. Jenis-jenis Microphone berdasarkan Kepekaannya

Microphone memiliki berbagai jenis. Berikut ini adalah jenis-jenis microphone berdasarkan kepekaannya:

1) *Omnidirectional microphone* : jenis microphone yang dapat menerima suara dari segala arah.
2) *Bidirectional microphone* : jenis microphone yang mampu menerima suara dari depan dan belakang, sedangkan jika dari sambing tidak peka.
3) *Unidirectional microphone* : jenis microphone yang hanya mampu menerima suara dari satu arah saja.

2.9.2. Sifat-sifat *Microphone*

Microphone terbagi dua menurut sifatnya, yaitu:

1) Microphone Dynamic

Microphone dynamic adalah suatu transduser yang menghasilkan sinyal listrik, akibat perubahan garis gaya magnet. Garis gaya magnet berubah-ubah karena tekanan yang berubah-ubah pada membran. Membran bergerak karena perubahan getaran udara yang mengenai membran.

2) Microphone Condensor

Microphone Condensor adalah suatu transducer yang mampu menghasilkan signal listrik. Signal listrik yang dihasilkan adalah karena perubahan nilai kapasitansi pada microphone tersebut. Microphone condensar disupply dengan batu battery sebesar 1,5 volt atau lebih tergantung dari kontruksinya.

2.10. *ArduinoUNO*

*ArduinoUNO*adalah board berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital input / output pin (dimana 6 pin dapat digunakan sebagai *output PWM*), 6 input *analog*, 16 MHz *osilator* kristal, koneksi USB, jack

listrik tombol reset. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.

Gambar 2.21 Board*ArduinoUNO* dan kabel USB.

Sumber : Datasheet ArduinoUNO

Board *ArduinoUNO*memiliki karakteristik sebagai berikut :

Tabel 2.2 Karakteristik *ArduinoUNO*.

Mikrokontroler	ATMega328
Operasi Voltage	5 V
Input Voltage	7 – 12 V (rekomendasi)
Input Voltage	6 – 20 V (limit)
I/O	14 pin (6 pin untuk PWM)
Arus	50 mA
Flash Memory	32 KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 MHz

Sumber : Datasheet ArduinoUNO

2.10.1. Catu Daya

*ArduinoUNO*dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (non-USB) daya dapat datang baik dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya *plug* pusat-positif 2.1mm ke dalam

board colokan listrik. Lead dari baterai dapat dimasukkan ke dalam *header* pin Gnd dan Vin dari konektor *Power*. *Board* dapat beroperasi pada pasokan daya dari 6 - 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat menyuplai kurang dari 5 volt dan *board* mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Rentang yang dianjurkan adalah 7 - 12 volt.

Pin catu daya adalah sebagai berikut:

- VIN. Tegangan input ke *board Arduino* ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- 5V. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di *board*. Hal ini dapat terjadi baik dari VIN melalui regulator onboard,
- atau diberikan oleh USB .
- 3,3 volt pasokan yang dihasilkan oleh regulator on-board. Menarik arus maksimum adalah 50 mA.
- GND

2.10.2. Memory

ATmega328 ini memiliki 32 kB dengan 0,5 kB digunakan untuk *loading file*. Ia juga memiliki 2 kB dari SRAM dan 1 kB dari EEPROM.

2.10.3. Input & Output

Masing-masing dari 14 pin digital pada *ArduinoUNO* dapat digunakan sebagai input atau output, menggunakan fungsi pinMode(), digitalWrite(), dan digitalRead(). Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal dari 20-50 K. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat attachInterrupt () fungsi untuk rincian.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit output PWM dengan analogWrite () fungsi.
- SPI: 10 (SS), 11 (mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH, itu off.
- *UNO* memiliki 6 input analog, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara default sistem mengukur dari tanah sampai 5 volt.
- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- Aref. Referensi tegangan untuk input analog. Digunakan dengan *analogReference* ().
- Reset.

2.10.4. Komunikasi

ArduinoUNO memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, *Arduino* lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran *board* ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. *Firmware Arduino* menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Inf diperlukan. Perangkat lunak *Arduino* termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board Arduino*. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

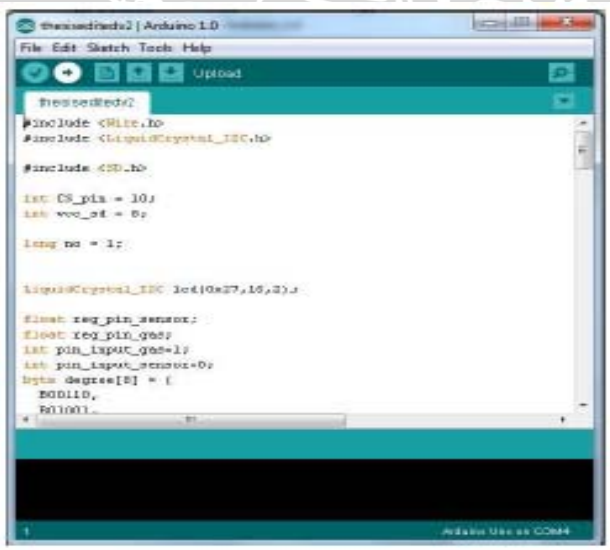
ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi inteface pada sistem.

2.10.5. Programming

*ArduinoUNO*dapat diprogram dengan perangkat lunak *Arduino*. Pilih *ArduinoUNO* dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. ATmega328 pada *ArduinoUNO*memiliki bootloader yang memungkinkan Anda untuk meng-upload program baru untuk itu tanpa menggunakan programmer hardware eksternal. Ini berkomunikasi menggunakan protokol dari bahas C. Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau programmer DFU (Mac OS X dan Linux) untuk memuat *firmware* baru. Atau Anda dapat menggunakan header ISP dengan programmer eksternal.

2.10.6. Perangkat Lunak (*Arduino* IDE)

Lingkungan *open-source Arduino*memudahkan untuk menulis kode dan meng-*upload* ke *board Arduino*. Ini berjalan pada Windows, Mac OS X, dan Linux. Berdasarkan Pengolahan, avr-gcc, dan perangkat lunak sumber terbuka lainnya.



Gambar 2.22 Tampilan *framework Arduino*.
Sumber : Datasheet Arduino

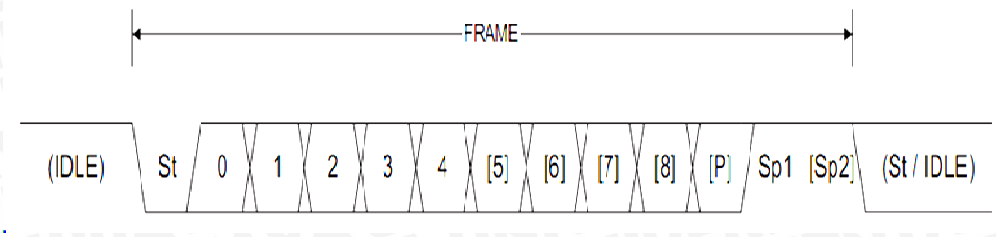
2.10.7. Otomatis Software Reset

Tombol reset*ArduinoUNO*dirancang untuk menjalankan program yang tersimpan didalam mikrokontroller dari awal. Tombol reset terhubung ke Atmega328 melalui kapasitor 100nf. Setelah tombol reset ditekan cukup lama untuk me-reset chip, *software IDE Arduino* dapat juga berfungsi untuk meng-*upload* program dengan hanya menekan tombol *upload* di *software IDE Arduino*.

2.11. Komunikasi Serial

Pada prinsipnya, komunikasi serial ialah komunikasi dimana pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan komunikasi paralel seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam sekali waktu. Devais pada komunikasi serial port dibagi menjadi 2 (dua) kelompok yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). Pengiriman data secara serial dapat berupa sinkron, yaitu pengiriman clock dilakukan bersamaan dengan data, atau berupa asinkron, yaitu pengiriman clock tidak bersamaan dengan data namun secara dua tahap, saat data dikirim dan data diterima. Untuk istilah yang sering digunakan untuk mengirim dan menerima data secara asinkron biasa disebut *Universal Asynchronous Receiver Transmitter* (UART). Komunikasi data serial menggunakan UART sangat umum dan mudah penggunaannya, misalnya pada port serial PC. Pada UART jalur pengiriman dan penerimaan data serial dipisahkan.

Setiap pengiriman data pada UART menggunakan bit tanda *start* bit dan *stop* bit. Jalur data yang digunakan hanya satu untuk setiap pengiriman data. Data-data serial dikirim melewati jalur data satu persatu setiap satuan waktu. Format pengiriman data serial secara asinkron ditunjukkan dalam Gambar 2.12.



Gambar 2.23 Format frame data serial USART

Sumber: Atmel, 2007:137

BAB III
METODOLOGI PENELITIAN

Penyusunan proposal ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasian sistem agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah-langkah yang perlu dilakukan untuk merealisasikan sistem yang dirancang adalah penentuan spesifikasi alat, studi literatur, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

3.1. Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut:

1) Robotmenggunakan sistem penggerak roda (*tracked*) yang terletak di sisi kiri dan kanan badan robot, serta digerakkan oleh dua motor DC dan menggunakan satu roda bebas yang terletak di bagian belakang robot.
2) Pencapit robot menggunakan dua motor servo.
3) Menggunakan modul *ArduinoUNO* sebagai pengendali utama *mobile robot*.
4) Menggunakan modul *EasyVR* sebagai pengolah suara.
5) Robot mampu mengenali suara maju, mundur, kanan, kiri, stop, buka dan tutup.

3.2. Studi Literatur

Studi literatur dilakukan untuk mempelajari teori penunjang sistem yang dibutuhkan dalam perencanaan dan pembuatan alat. Teori yang diperlukan antara lain berkaitan dengan *mobile robot*, sistem *voice recognition*, modul *EasyVR*,sensor *electret microphone*, dan *ArduinoUNO*.

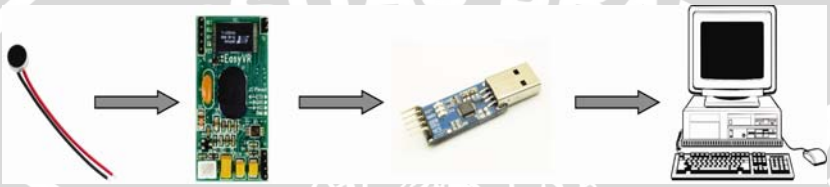
3.3. Perancangan dan Perealisasian Alat

3.3.1. Perancangan Perangkat Keras dan Realisasi Tiap Blok

a. Penentuan spesifikasi alat
b. Pembuatan blok diagram lengkap sistem
c. Pembuatan mekanik robot
d. Penentuan dan perhitungan komponen yang akan digunakan
e. Merakit perangkat keras masing-masing blok

3.3.2. Perancangan *Voice Recognition* Menggunakan *EasyVR*

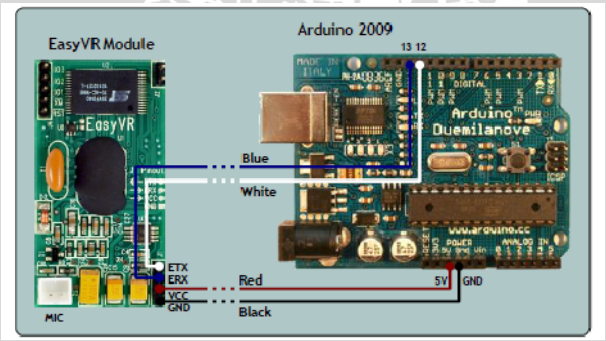
3.3.2.1. Perancangan Sistem Komunikasi *EasyVR* ke Komputer



Gambar 3.1. Proses komunikasi *EasyVR* dengan komputer.

Proses komunikasi antara *EasyVR* dengan computer berfungsi untuk sampling data berupa suara melalui software *EasyVRCommander*. Komunikasi antara *EasyVR* menggunakan komunikasi serial berupa USB to TTL.

3.3.2.2. Perancangan Sistem Komunikasi *EasyVR* ke *ArduinoUNO*

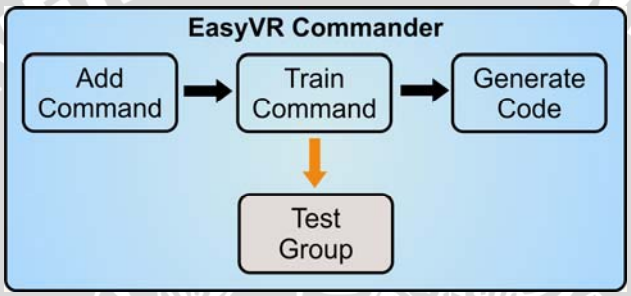


Gambar 3.2. Proses komunikasi *EasyVR* dengan *ArduinoUNO*.

Komunikasi antara *EasyVR* dengan *ArduinoUNO* menggunakan Bridge Mode atau secara serial. Pin 13 pada *ArduinoUNO* difungsikan sebagai Tx yang akan dihubungkan dengan Rx dari *EasyVR*. Sedangkan pin 12 pada *ArduinoUNO* difungsikan sebagai Rx yang akan dihubungkan dengan Tx pada *EasyVR*.

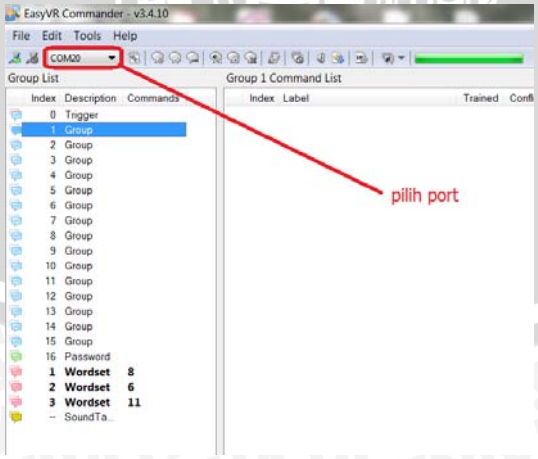
Hasil sampling data yang tersimpan dalam *EasyVR* tidak dikirimkan ke mikrokontroler yang terdapat pada *ArduinoUNO*, namun tetap tersimpan di dalam *EasyVR*. Prinsip kerjanya adalah pada saat input berupa suara sesuai degan hasil sampling, *EasyVR* akan memberi perintah kepada *ArduinoUNO* berupa data serial 8 bit dengan kecepatan 9600 bps. Jika *Arduino* menerima data karakter ASCII "A" (dalam bentuk desimal 65 atau hex 41) dari *EasyVR*, maka akan muncul data berupa 10000010.

3.3.3. Perancangan Pengambilan Sampel Suara Menggunakan*EasyVRCommander*



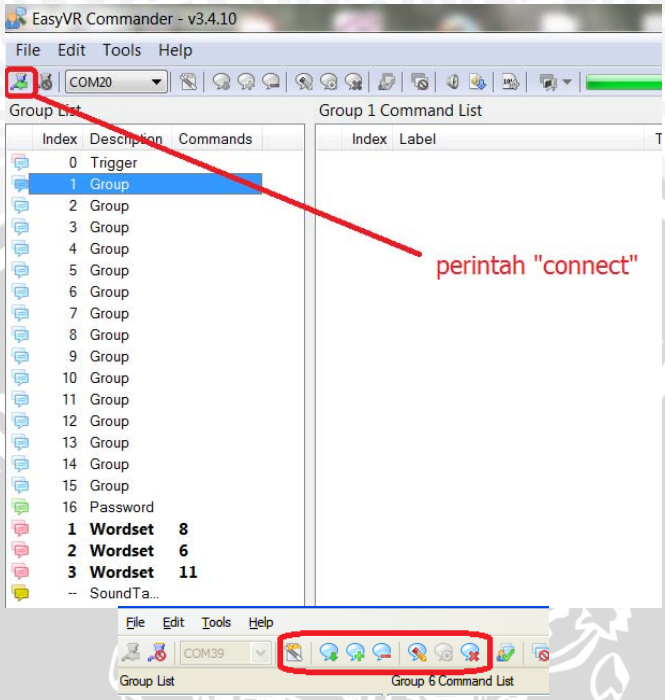
Gambar 3.3. Blok diagram sistem voice reognition melalui *EasyVR*.

Sebelum modul *EasyVR* dihubungkan ke PC dengan menggunakan antarmuka serial berupa USB to TTL, software *EasyVRCommander* harus diaktifkan terlebih dahulu. Kemudian terdapat pilihan port USB yang terhubung oleh modul *EasyVR* yang ditunjukkan dalam Gambar 3.4.



Gambar 3.4 Tampilan pada software *EasyVRCommander*

Kemudian tekan perintah *connect* agar *EasyVR* dapat terhubung dengan software tersebut dan untuk mengaktifkan berbagai perintah yang terdapat dalam software tersebut seperti yang ditunjukkan dalam Gambar 3.5.



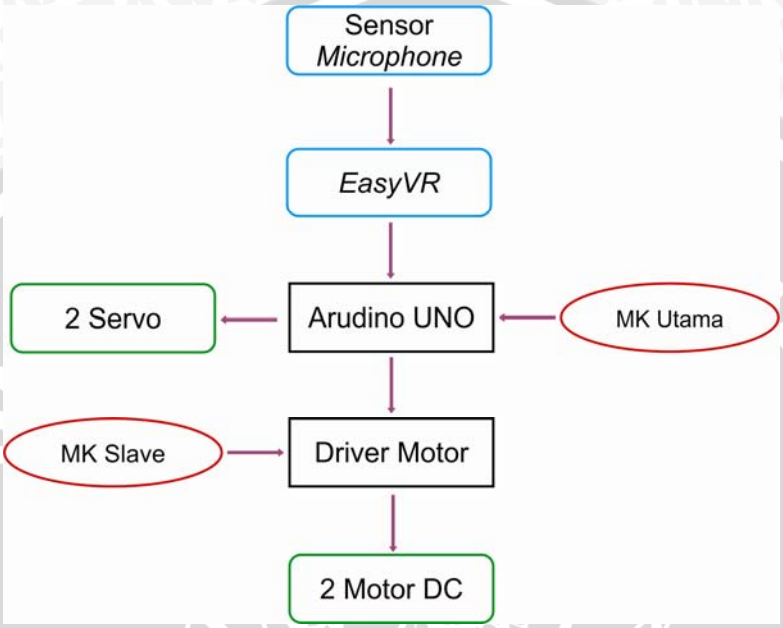
Gambar 3.5 Tampilan pada software *EasyVRCommander*

Terdapat beberapa command yang dapat digunakan untuk memperoleh sampling suara. Langkah pertama untuk mendapatkan sampling suara adalah dengan menambahkan command ke dalam group melalui perintah "add command" yang terdapat dalam software. Kemudian memilih perintah "train command" untuk memasukkan sampling suara yang akan digunakan. Command yang telah ditambahkan akan secara langsung tersimpan dalam database pada modul *EasyVR*. Perintah "test group" berguna untuk mengecek apakah hasil sampling suara mampu digunakan dengan cara memasukkan input suara melalui sensor microphone. Generate code berguna untuk menunjukkan code dalam bahasa C untuk mempermudah pemrograman pada perancangan software.

3.3.4. Perancangan dan Penyusunan Perangkat Lunak

Penyusunan perangkat lunak digunakan untuk mengendalikan dan mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang kemudian diterapkan ke dalam program *Arduino* dengan menggunakan bahasa C.

3.3.5. Perancangan Sistem Secara Keseluruhan



Gambar 3.6 Diagram blok sistem secara keseluruhan.

Fungsi masing-masing bagian dalam diagram blok yang ditunjukkan dalam gambar 3.6 adalah sebagai berikut:

1) Sensor *microphone* berfungsi untuk mendeteksi suara.
2) Modul*EasyVR* berfungsi sebagai antarmuka yang berisi database berupa sampling suara yang telah dikonfigurasi dengan *EasyVRCommander*.
3) *ArduinoUNO* yang berfungsi untuk memanggil dan mengolah data-data berupa suara dari *EasyVR*yang kemudian akan diolah untuk proses navigasi robot.
4) Driver motor berfungsi sebagai antarmuka *ArduinoUNO* dengan motor DC.

Prinsip kerja dari perancangan adalah robot aktif saat tombol ON diaktifkan. Setelah itu pemberian input dilakukan dengan menggunakan perintah suara. Data akan diakses oleh *EasyVR*yang akan melakukan pemilihan jenis suara

dan akan diolah oleh mikrokontroler yang melakukan pemrosesan data untuk menggerakkan motor servo sesuai dengan perintah yang diberikan.

3.4. Pengujian Alat

Pengujian alat dilakukan untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan. Pengujian alat meliputi pengujian perangkat keras yang dilakukan baik per blok maupun keseluruhan sistem.

3.4.1. Pengujian Tiap Blok

Pengujian per blok dilakukan dengan tujuan untuk menyesuaikan nilai masukan dan nilai keluaran tiap-tiap blok sesuai dengan perancangan yang dilakukan sebelumnya.

3.4.2. Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

3.5. Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya.

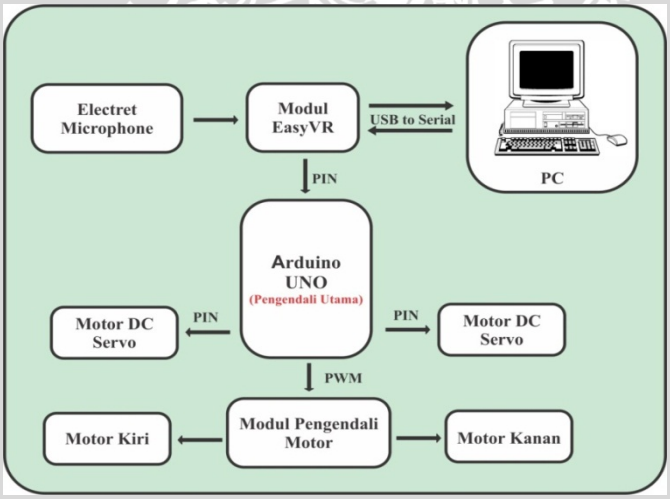
BAB IV
PERANCANGAN DAN PEMBUATAN ALAT

Perancangan robot ini dilakukan secara bertahap sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri dari:

- Perancangan sistem.
- Perancangan perangkat keras.
- Perancangan sistem *voice recognition*.
- Perancangan perangkat lunak.

4.1. Perancangan Sistem

Blok diagram keseluruhan sistem yang dirancang dibagi menjadi dua bagian, yaitublok mikrokontroler utama, blok pengendali motor DC, dan blok *EasyVR* seperti ditunjukkan dalam Gambar 4.1.



Gambar 4. 1. Diagram Blok Sistem

Fungsi masing-masing blok dalam diagram blok diatas adalah sebagai berikut :

1). Blok Mikrokontroler Utama

Mikrokontroler yang digunakan adalah *ArduinoUNO* yang berfungsi untuk mengolah data dari modul *EasyVR* dan mengakses servo untuk kendali arah.

2). Blok Modul *EasyVR*

Pergerakan dari robot ini dibantu dengan modul *EasyVR* yang berfungsi sebagai sensor suara. Data-data suara akan di sampling melalui komputer menggunakan program *EasyVRCommander*. Data-data yang sudah disampling akan dimasukkan kembali ke database *EasyVR*. Mikrokontroler utama berupa *ArduinoUNO* bekerja mengatur dan mengolah data dari modul *EasyVR* .

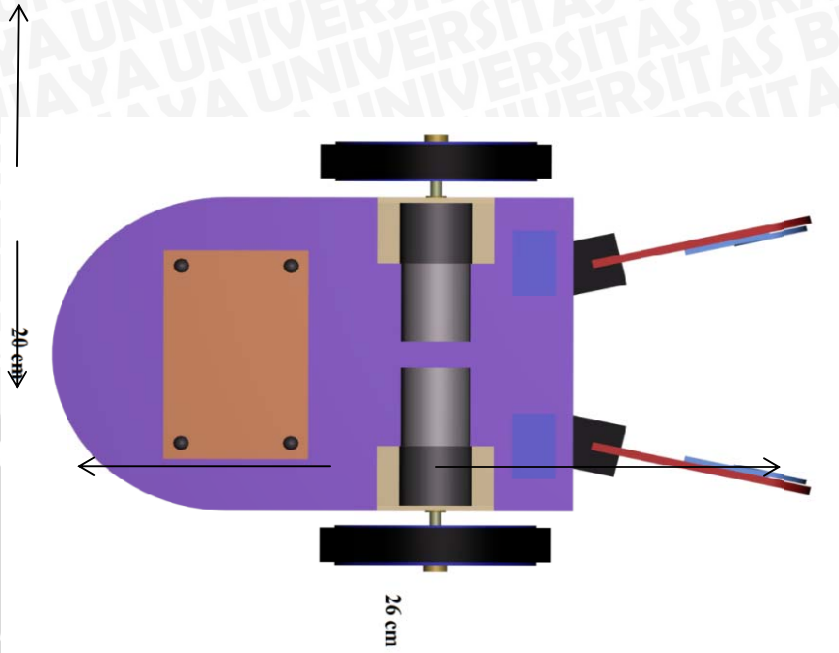
3). Blokmotor DC

Pada modul pengendali motor, data yang diterima dikonversi menjadi sinyal kendali arah. Motor kanan dan motor kiri berfungsi sebagai kemudi yang menggerakkan robot ketika harus maju, mundur, belok kanan atau belok kiri.

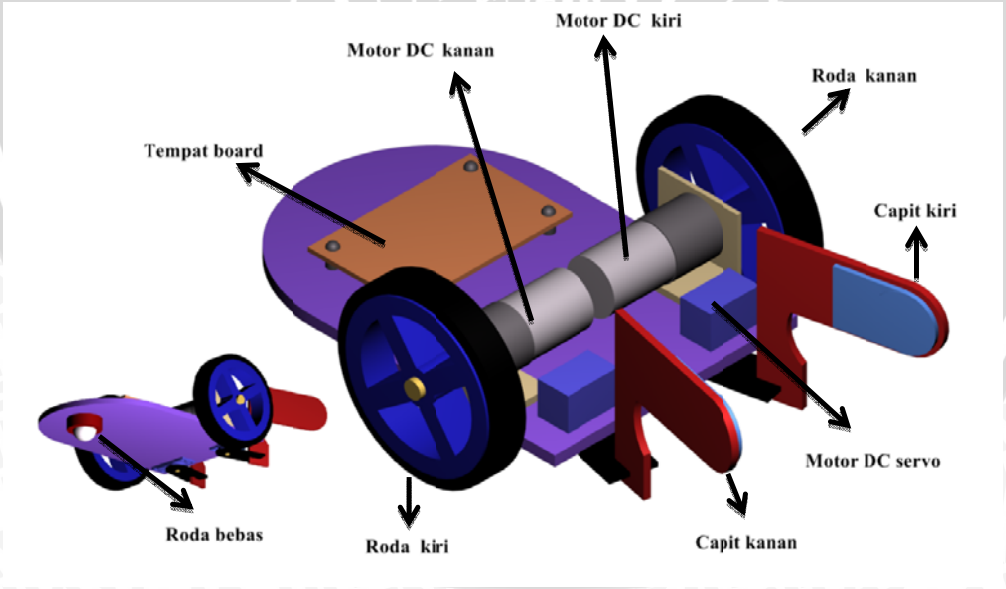
4.2. Perancangan Perangkat Keras

4.2.1. Perancangan Mekanik Robot

Sistem mekanik yang baik, mendukung pergerakan robot menjadi lebih baik, oleh karena itu perancangan mekanik dalam hal ini badan robot haruslah proporsional dengan panjang dan lebar serta tinggi dari robot. Gambar perspektif dan rancangan ukuran Robot ditunjukkan dalam Gambar 4.2 dan Gambar 4.3.



Gambar 4.2. Perspektif Tampak Atas Robot Beroda



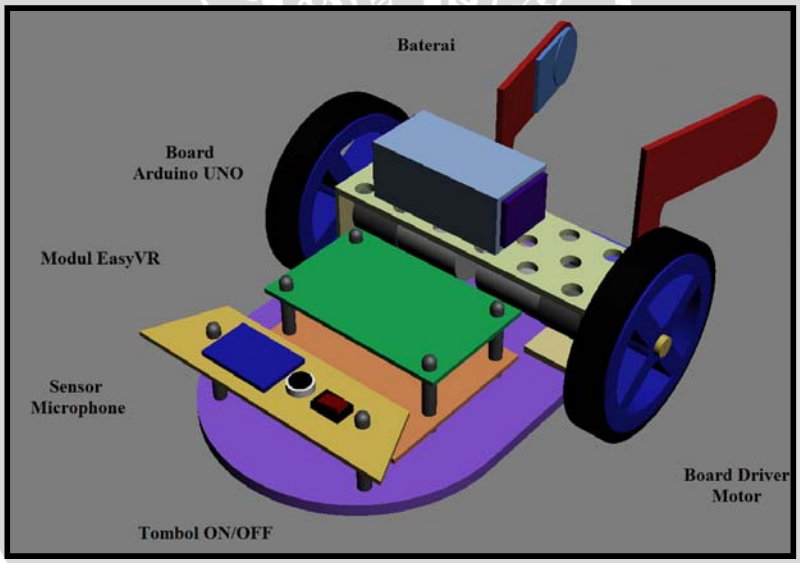
Gambar 4.3. Perspektif Desain Mekanik Robot Beroda

Badan robot terbuat dari bahan mika *acrylic* dengan ketebalan 3 mm, ke dua buah roda depan berbahan acrylic dengan tebal 6 mm dan berdiameter 10 cm.

Roda belakang menggunakan roda *castor* (roda bebas). Capit robot terbuat dari bahan acrylic dengan tebal 2 mm. Bentuk dan dimensi robot dirancang sesuai secara proporsional dengan harapan robot dapat bergerak sesuai dengan fungsinya yaitu memindahkan objek.

4.2.2. Perancangan Desain Sistem Elektronik

Diagram blok sistem elektronik, terdiri dari tiga bagian catu daya (baterai), masukan, bagian kendali, bagian keluaran. Pada bagian masukan berupa sebuah sensor microphoneyang berfungsi untuk menerima masukan berupa suara manusia yang terhubung dengan *EasyVR* sebagai pengolah data sinyal suara. Pada bagian kendali menggunakan *ArduinoUNO* sebagai miikrokontroller utama dan pengontrol gerak servo. Pada bagian keluaran berupa *driver* motor sebagai penggerak aktuator robot.

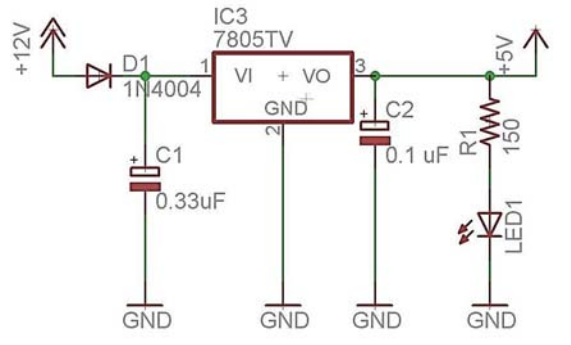


Gambar 4.4. Desain Sistem Elektronik Robot Beroda

4.2.2.1. Perancangan Catu Daya Sistem

Robot ini menggunakan satu jenis rangakaian catu daya, yaitu catu daya 5 V untuk rangkaian utama mikrokontroler *ArduinoUNO* yang meliputi *board EasyVR* dan *driver* motor DC L298. Sumber catu daya yang dipakai adalah satu baterai lipho (*lithium polimer*) 11,1 V.

ArduinoUNO dapat bekerja jika diberi catu tegangan antara 4,8 V hingga 5,5 V sesuai dengan datasheet *ArduinoUNO*. Pada perancangan digunakan catu daya sebesar 5V yang diperoleh dari rangkaian *Fixed Output Regulator* pada datasheet LM78XX. Pada rangkaian digunakan regulator LM7805 agar diperoleh tegangan keluaran sebesar 5 V. Skema rangkaian catu daya ditunjukkan dalam Gambar 4.5.



Gambar 4. 5. Rangkaian Catu Daya

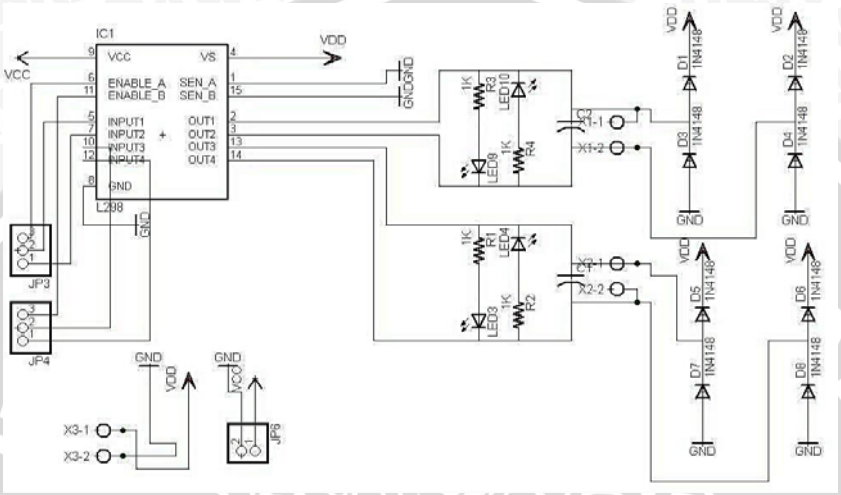
4.2.2.2. Perancangan Rangkaian *Driver* Pengendali Motor DC

Modul pengendali motor DC ini digunakan untuk mengendalikan arah dan putaran motor DC yang menjadi penggerak pada robot. Rangkaian ini dihubungkan dengan mikrokontroler ATMega32 (mikrokontroler utama). *Driver* pengendali menggunakan *driver* L298N yaitu sebuah perangkat keras berupa rangkaian yang berfungsi untuk menggerakkan motor DC. IC L298N terdiri dari transistor-transistor ogic (TTL) dengan gerbang nand yang memudahkan dalam menentukkan arah putaran suatu motor DC. *Driver* motor yang diperlukan oleh mobile robot cukup satu buah, karena IC L298N dapat mengendalikan dua buah motor DC sekaligus karena IC ini memiliki dua buah rangkaian *H-Bridge* di dalamnya. Spesifikasi IC L298N ini meliputi :

- $I_{O\ max}$ = 3A
- $I_{O\ min}$ = 2A
- Vs = 2,5 – 46 V (*Power Supply)*
- Vss = 4,5 – 7 V (*Logic Supply Voltage*)
- Ven = -0,3 – 7 V (*Input and Enable Voltage)*

Dalam perancangannya motor DC yang digunakan memiliki tegangan catu sebesar 12 V dengan kecepatan 75 rpm. Apabila robot menggunakan kecepatan maksimal. Sesuai dengan spesifikasi motor yang tertera dalam

datasheet, arus yang dibutuhkan motor saat tak berbeban dengan menggunakan *driver* L298N ini sebesar 80 mA. Sedangkan ketika berbeban atau kondisi robot sedang berjalan sebesar 2,2 A. Sehingga memenuhi dalam perancangan dengan menggunakan *driver* L298N. Skema rangkaian *driver* motor L298N ditunjukkan dalam Gambar 4.6.



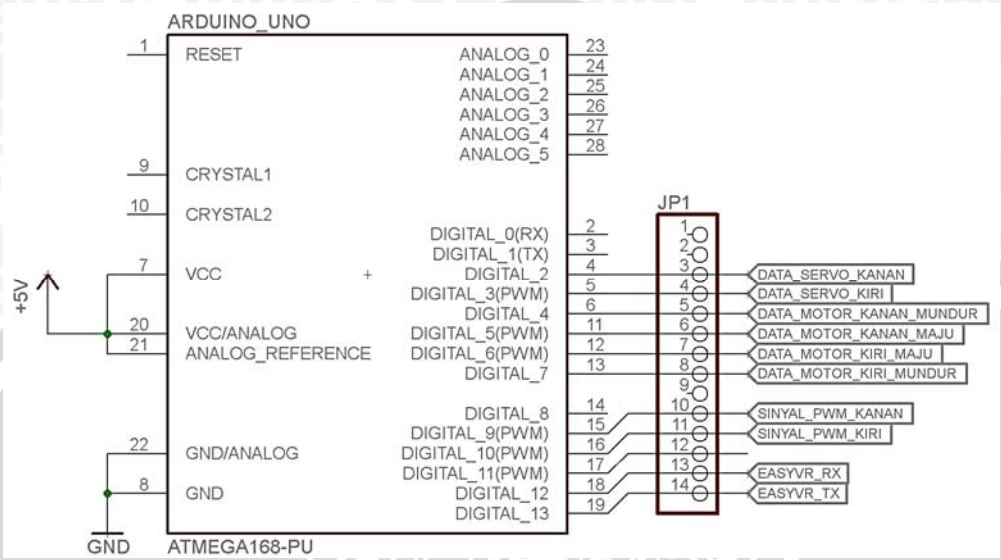
Gambar 4.6. Rangkaian *Driver* Motor DC

Dalam perancangan konfigurasi *driver* L298N ini untuk menjalankan motor, pin *enable* A dan *enable* B dihubungkan pada Pin 9 (PWM) dan pin 10 (PWM) pada *ArduinoUNO* dan harus diberi logika 1. *Current sensing* A dan *current sensing* B dihubungkan ke *ground*. Pada perancangan terdapat 4 *input* dimana *input* 1 dan *input* 2 untuk motor kanan masing-masing berlogika 1 dan 0, begitu juga dengan input 3 dan input 4 berlogika 1 dan 0 untuk motor kiri. Output 1, output 2, output 3 dan output 4 dihubungkan ke masing-masing motor . Pada VCC menggunakan tegangan sebesar 11,1 V dari baterai *lithium polymer* dan VDD sebesar 5 V dari mikrokontroler.

Output dari IC L298N ini tidak memiliki dioda pengaman oleh karena itu pada perancangan *driver* motor ini ditambahkan 2 buah dioda pada setiap titik output, sehingga total terdapat 8 buah dioda untuk 2 motor masing - masing motor 4 buah dioda. Menggunakan 4 buah LED sebagai indikator pergerakan motor. LED berwarna hijau untuk motor kanan dan LED berwarna merah untuk motor kiri.

4.2.2.3. Perancangan Rangkaian Mikrokontroler Pengatur Utama

Pada perancangan mikrokontroler pengatur utama menggunakan modul *ArduinoUNO* sebagai pengolah utama dalam melakukan proses pengolahan data. Konfigurasi kaki I/O dari *ArduinoUNO* ditunjukkan dalam Gambar 4.7.



Gambar 4.7. Minimum Sistem *ArduinoUNO*

ArduinoUNO mempunyai 4 port, 32 jalur yang dapat diprogram menjadi masukan atau keluaran, pada perancangan ini pin-pin yang digunakan adalah:

Pin D_2 = dihubungkan dengan servo sebagai data untuk menentukan sudut.

Pin D_3 = dihubungkan dengan servo sebagai data untuk menentukan sudut.

Pin D_4 = dihubungkan dengan *driver* motor sebagai data *enable* untuk logika.

Pin D_5 = dihubungkan dengan *driver* motor sebagai data *enable* untuk logika.

Pin D_6 = dihubungkan dengan *driver* motor sebagai data *enable* untuk logika.

Pin D_7 = dihubungkan dengan *driver* motor sebagai data *enable* untuk logika.

Pin D_9 = dihubungkan dengan *driver* motor sebagai sinyal PWM.

Pin D_10 = dihubungkan dengan *driver* motor sebagai sinyal PWM.

Pin D_12 = Tx untuk transmisi UART dengan *EasyVR*.

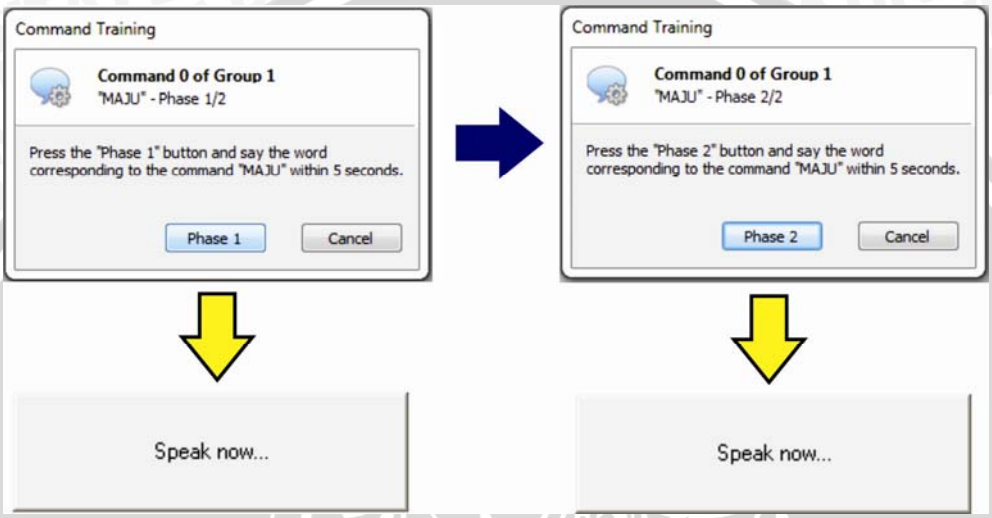
Pin D_13 = Rx untuk transmisi UART dengan *EasyVR*.

Pin VCC, AVCC dan AREF = dihubungkan dengan sumber tegangan 5 V.

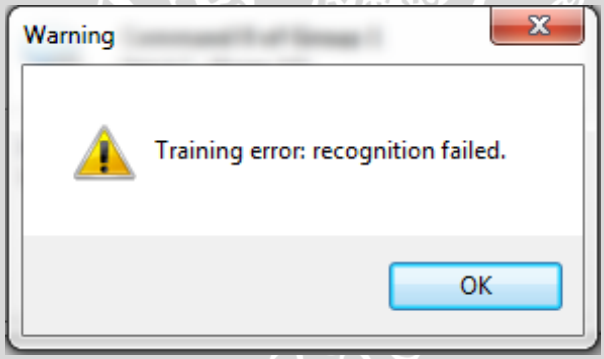
Pin GND dan AGND = dihubungkan dengan *ground.*

4.3. Perancangan Sistem*Voice Recognition* Menggunakan *EasyVR*

Perancangan ini bertujuan untuk mengambil sample suara yang akan disimpan ke dalam modul *EasyVR*. Pengambilan sampling suara dilakukan melalui PC dengan software bawaan dari *EasyVR* yaitu *EasyVRcommander*. Sample suara yang akan digunakan adalah sebanyak tujuh kata.



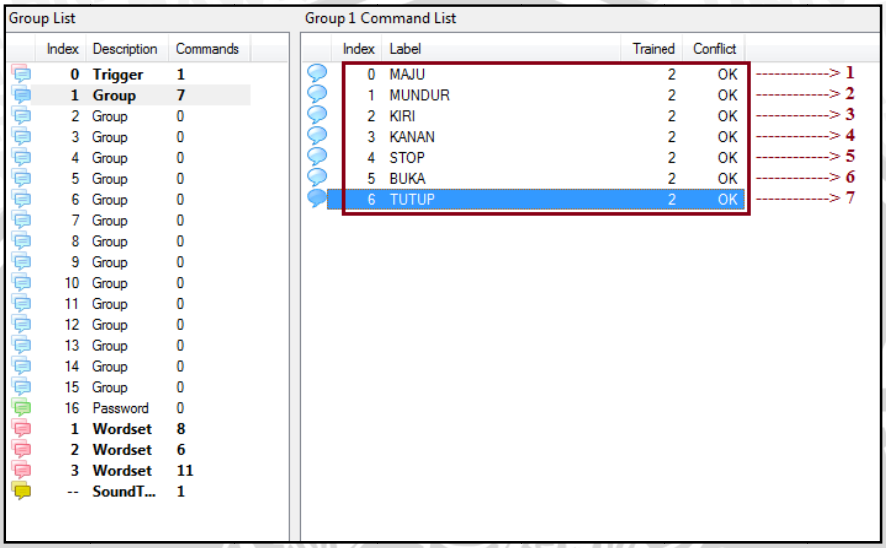
Gambar 4.8 Diagram blok pemberian *sample.*



Gambar 4.9 Tampilan kegagalan dalam pengambilan *sample.*

Diagram blok pemberian sample ditunjukkan dalam Gambar 4.8.Pengambilan *sample* suara dilakukan dalam kondisi ideal atau tanpa adanya *noise* dan dilakukan sebanyak dua kalipengambilan suara dengan variasi pengucapan yang relatif sama pada setiap kata. Hal ini dilakukan sesuai dengan kemampuan *EasyVR* yang tidak bisa menerima pengucapan variasi suara kedua jika berbeda dengan variasi pengucapan suara pertama. Kegagalan menerima variasi suara kedua akan mengakibatkan munculnya kalimat seperti yang

ditunjukkan dalam Gambar 4.9.Oleh karena itu, agar *EasyVR* dapat berfungsi atau dapat mengakses perintah suara, dibutuhkan variasi suara yang relatif sama dengan *sample*. Selain itu, *EasyVR commander* memberi batas waktu selama 5 detik untuk setiap pengucapan suara yang akan dijadikan *sample*.



Gambar 4.10. Sampling Pada yang Diambil pada *EasyVRCommander*

Gambar 4.10 menunjukkan sampling data yg diambil dan terdapat dalam group 1. Berikut adalah keterangan dari masing-masing hasil sampling:

1. MAJU : Berisi sampling suara dengan kata "maju" yang berfungsi untuk navigasi robot berjalan ke depan. Karakter ASCII yang digunakan adalah 0.
2. MUNDUR : Berisi sampling suara dengan kata "mundur" yang berfungsi untuk navigasi robot berjalan ke belakang. Karakter ASCII yang digunakan adalah 1.
3. KIRI : Berisi sampling suara dengan kata "kiri" yang berfungsi untuk navigasi robot belok ke kiri. Karakter ASCII yang digunakan adalah 2.
4. KANAN : Berisi sampling suara dengan kata "kanan" yang berfungsi untuk navigasi robot belok ke kanan. Karakter ASCII yang digunakan adalah 3.
5. STOP : Berisi sampling suara dengan kata "stop" yang berfungsi

untuk navigasi robot berhenti. Karakter ASCII yang digunakan adalah 4.

6. BUKA : Berisi sampling suara dengan kata "buka" yang berfungsi untuk navigasi robot membuka capit. Karakter ASCII yang digunakan adalah 5.
7. TUTUP : Berisi sampling suara dengan kata "tutup" yang berfungsi untuk navigasi robot menutup capit. Karakter ASCII yang digunakan adalah 6.

4.4. Perancangan dan Pembuatan Perangkat Lunak (Software)

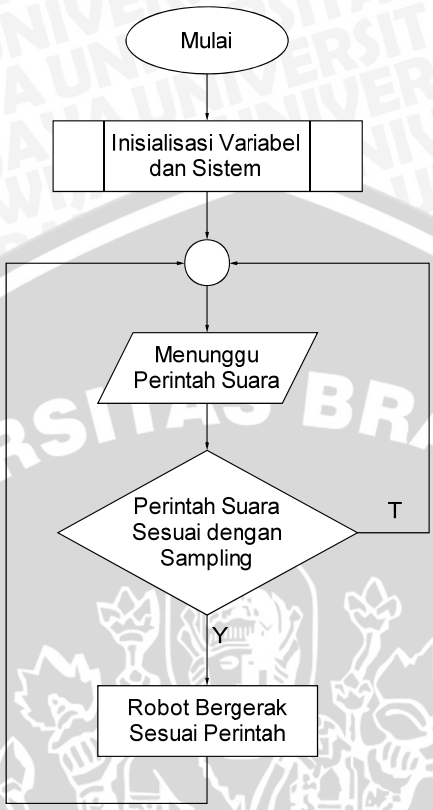
Setelah melalui proses perancangan dan pembuatan perangkat keras, selanjutnya akan dilakukan perancangan dan pembuatan perangkat lunak. Perangkat lunak berfungsi untuk memberikan instruksi kerja kepada perangkat keras tersebut.

4.4.1. Perancangan Susunan Perangkat Lunak

Tahapan proses yang terdapat pada sistem ini meliputi proses pengolahan data dari modul *EasyVR* ke *Arduino* dan proses pengontrolan motor DC maupun servo. Semua proses tersebut dilakukan oleh perangkat lunak yang terdapat dalam mirokontroler. Perangkat lunak ini tersusun dari instruksi-instruksi yang membentuk sebuah *listing* program atau *source code.*

Semua intruksi program disusun secara terstruktur dalam beberapa subrutin yang secara khusus menangani fungsi tertentu. *Software* mikrokontroler dibuat menggunakan program *Arduino*. Menggunakan bahasa pemrogaman yaitu bahasa C.

4.4.2. Diagram Alir Keseluruhan Sistem



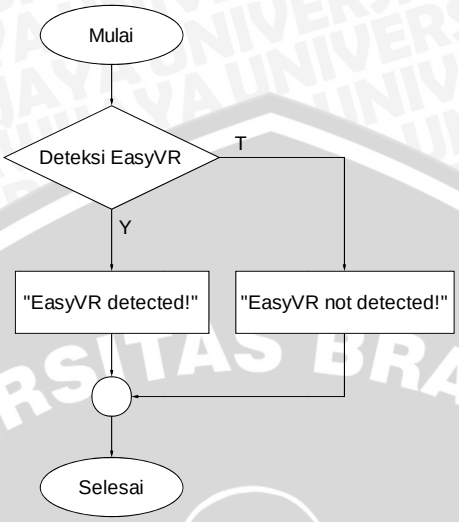
Gambar 4.11. Diagram alir keseluruhan sistem.

Diagram alir keseluruhan sistem ditunjukkan dalam Gambar 4.11 Robot akan menginisialisasi sistem yang didalamnya juga terdapat sistem pendeteksi keberadaan *EasyVR*. Kemudian robot menunggu perintah berupa suara. Setelah itu, data input berupa suara dibandingkan dengan data sampling yang ada pada *EasyVR*. Jika sesuai, board *ArduinoUNO* akan menerima perintah dari board *EasyVR* dan mengirimkan perintah kepada rangkaian driver motor dan servo berupa pergerakan atau navigasi.

4.4.3. Diagram Alir Program Pendeteksi *EasyVR*

Pendeteksi *EasyVR* berfungsi untuk mengetahui apakah modul *EasyVR* sudah connect atau belum dengan *ArduinoUNO*. Keluaran dari program ini akan muncul pada serial monitor yang terdapat dalam software *Arduino*. Jika *EasyVR* terdeteksi, akan muncul kalimat "*EasyVR* Detected" pada tampilan awal serial monitor. Sebaliknya jika tidak terdeteksi, akan muncul kalimat "*EasyVR* not Detected". Serial monitor selain berfungsi untuk mengetahui keberadaan *EasyVR*,

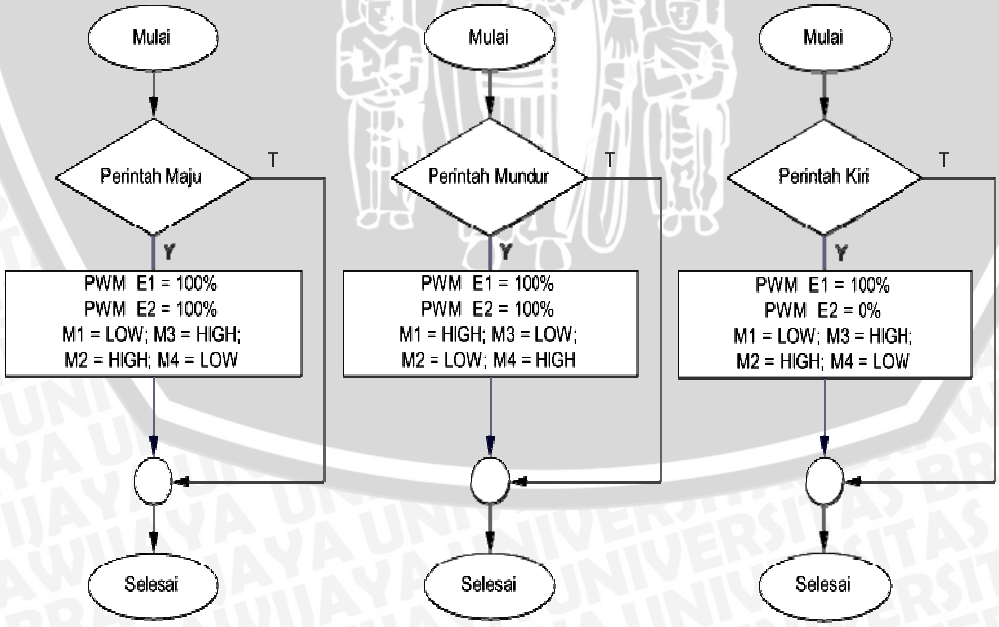
juga untuk mengetahui pengiriman data suara apakah sesuai atau tidak. Gambar 4.12 menunjukkan diagram alir program pendeteksi *EasyVR*.

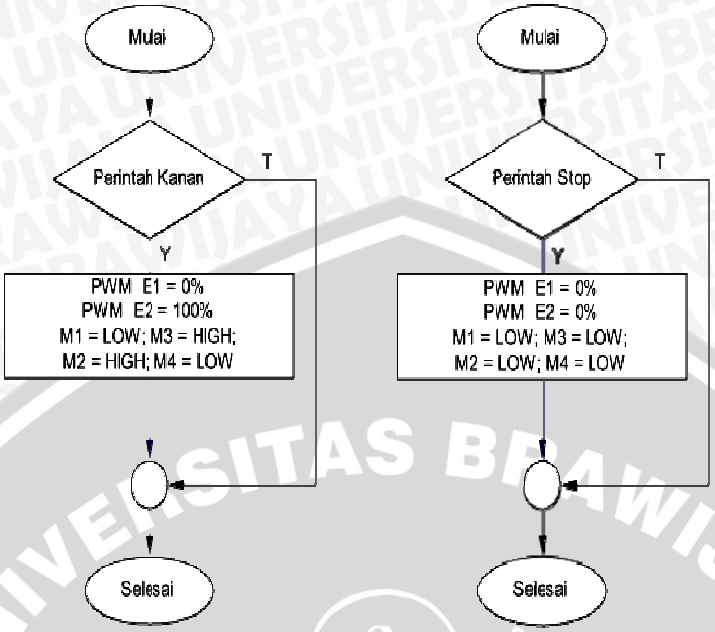


Gambar 4.12. Diagram alir program pendeteksi *EasyVR*

4.4.4. Diagram Alir Program Pengontrol Motor DC

Program pengontrol motor berfungsi untuk untuk mengatur arah gerak putaran motor dan mengatur kecepatan setiap motor berdasarkan perintah suara yang dikeluarkan. Arah gerak digunakan untuk manuver robot. Diagram alir proses manuver robot ditunjukkan dalam Gambar 4.13..





Gambar 4.13. Diagram alir program pengontrol motor DC arah maju, arah mundur, arah kanan, arah kiri, dan stop/berhenti

BAB V
PENGUJIAN DAN ANALISIS

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan yang telah dilakukan. Pengujian dilakukan per blok sistem kemudian secara keseluruhan. Adapun pengujian yang dilakukan sebagai berikut:

1) Pengujian rangkaian pengendali motor DC *brushed*
2) Pengujiaan komunikasi serial UART ke PC atau laptop
3) Pengujian *voice recognition*
4) Pengujian keseluruhan sistem

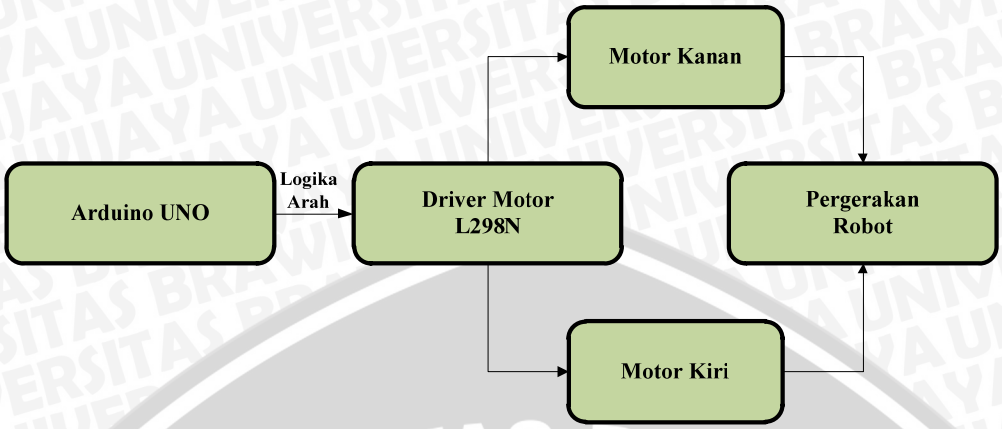
5.1. Pengujian Rangkaian *Driver*Pengendali Motor DC

Pengujian ini bertujuan untuk mengetahui kinerja dan respons dari rangkaian *driver* motor L298N sebagai pengendali motor DC dengan membandingkan dan menguji sinyal keluaran dari *driver* motor L298Nterhadap sinyal masukan arah dan sinyal masukan PWM yang diberikan oleh mikrokontroler.

Pada pengujan ini ada dua prosedur yang harus dilakukan, yaitu prosedur untuk pengujian respons*driver* motor L298Nterhadap sinyal masukan arah dari mikrokontroler dan pengujian respons*driver* motor L298N terhadap sinyal masukan PWM dari mikrokontroler.

5.1.1. Pengujian Respons*driver* Motor L298Nterhadap Masukan Sinyal Arah

Prosedur pengujian dilakukan dengan menghubungkan *driver* motor L298N, *ArduinoUNO*, dan motor DC sesuai Gambar 5.1.



Gambar 5.1. Diagram Blok Pengujian Respons Sinyal Arah Rangkaian *Driver* Motor L298N

*ArduinoUNO*akan memberikan instruksi berupa arah dan kecepatan pada *driver* motor L298N. Kecepatan dan arah putar motor DC diukur dengan menggunakan perangkat *timer 0* dan *timer 2*. Motor DC kemudian berputar sesuai intruksi arah yang diberikan oleh *ArduinoUNO*. Dengan memberi masukan sinyal arah motor 1 atau 0 ke *driver* motor L298N maka di dapatkan hasil pengujian dalam Tabel 5.1.

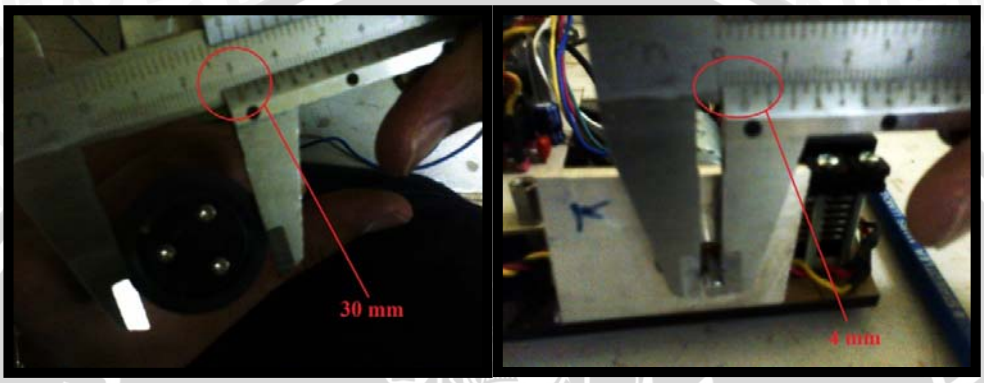
Tabel 5.1 Hasil pengujian respons driver motor L298N.

Logika Arah Motor Kanan		**Logika Arah Motor Kiri**		**Respons Robot**	
PIN D.5	**PIN D.4**	**PIN D.6**	**PIN D.7**	**Pengujian yang diharapkan**	**Hasil Pengujian**
1	0	1	0	Maju	Maju
0	1	0	1	Mundur	Mundur
1	0	0	0	Belok ke kiri	Belok ke kiri
0	0	1	0	Belok ke kanan	Belok ke kanan
0	0	0	0	Stop	Stop

Pada Tabel 5.1 dapat diketahui bahwa respons*driver* motor L298N terhadap masukan sinyal arah pada mikrokontroler berjalan sesuai perancangan yang diharapkan. Sehingga dapat disimpulkan bahwa *driver* motor L298N dapat bekerja dengan baik saat mendapatkan sinyal logika arah dari mikrokontroler.

5.1.2. Pengujian Keluaran*Driver* Motor L298Nterhadap Masukan Sinyal PWM

Pengujian ini dilakukan untuk membandingkan nilai masukan *duty cycle* yang kita berikan dengan keluaran *duty cycle* pada *driver* motor L298N berupa kecepatan putaran motor.Pengujian sinyal PWM mikrokontroler dan keluaran *driver* motor L298N menggunakan tachometer analog.



(a) (b)

Gambar 5.2 (a) Pengukuran diameter as pada tachometer. (b) Pengukuran diameter as pada motor

Gambar 5.2 menunjukkan diameter as pada tachometer dan pada motor. As tachometer memiliki diameter sebesar 30 mm, sedangkan diameter as motor sebesar 4 mm. Pengukuran kecepatan putaran motor dilakukan dengan membandingan antara kecepatan as pada motor dan as pada tachometer analog. Rumus perbandingannya adalah sebagai berikut :

$$\frac{diameter\ As\ tachometer}{diameter\ As\ motor} = \frac{kecepatan\ putaran\ motor}{kecepatan\ putaran\ tachometer} \quad \ldots\ldots(5.1)$$

Gambar 5.3 Pengukuran kecepatan motor kanan dengan masukan sinyal PMW 100% menggunakan tachometer analog.

Gambar 5.3 menunjukkan kecepatan putaran pada tachometer yang dikopel dengan motor kanan saat kondisi maju yaitu 10 rpm. Oleh karena itu, bisa dihasilkan kecepatan motor sebenarnya sebesar 75 rpm melalui rumus perbandingan. Tabel 5.2 menunjukkan hasil pengujian seluruh putaran motor dengan perhitungan sesuai dengan rumus perbandingan (5.1).

Tabel 5.2 Hasil pengujian respons rangkaian *driver* motor L298N terhadap masukan sinyal PWM

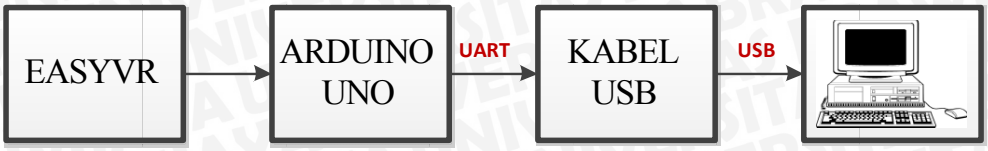
Gerakan Motor	PWM MK		Kecepatan putaran motor sebenarnya
	Duty Cycle	Kecepatan motor sesuai datasheet	
Kanan Maju	100%	75 rpm	75 rpm
Kanan Mundur	100%	75 rpm	75 rpm
Kiri Maju	100%	75 rpm	75 rpm
Kiri Mundur	100%	75 rpm	75 rpm

Kecepatan yang diukur hanya menggunakan duty cycle sebesar 100% saja karena seluruh navigasi hanya menggunakan duty cycle 100%. Berdasarkan Tabel, pada setiap pengujian terdapat selisihkecepatan motor rata-rata sebesar 0%. Selesih rata-rata 0% sangat bagus dan efisien karena tidak memberikan pengaruh pada kinerja sistem yang dirancang, sehingga dapat disimpulkan bahwa *driver* motor L298N dapat bekerja dengan baik saat mendapatkan sinyal PWM dari mikrokontroler.

5.2. Pengujiaan Komunikasi Serial UART ke PC

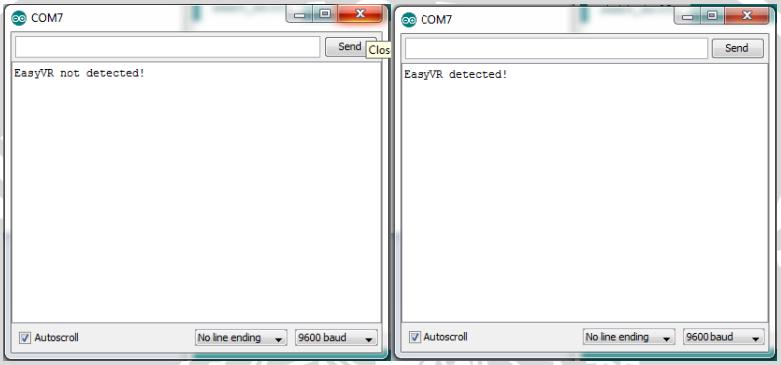
Pengujian ini dilakukan untuk mengetahui apakah rangkaian mikrokontroler dapat mengirimkan data melalui komunikasi serial UART. Untuk pengujian ini, mikrokontroler mengirimkan data pada terminal komputer menggunakan konfigurasi *baudrate* 9600 bps, 8 bit data, tanpa paritas dan 1 stop bit. Data yang dikirimkan merupakan paket data yang berisi pendeteksian modul *EasyVR*.

Prosedur pengujian dilakukan dengan menghubungkan *EasyVR*,*ArduinoUNO*, kabel serial USB, dan komputer sesuai Gambar 5.4.



Gambar 5.4. Diagram Blok Pengujian Respons Sinyal PWM Rangkaian *Driver* Motor L298N

ArduinoUNO akan memberikan instruksi sinyal pada pin Tx dan Rx yang terhubung oleh kabel USB yang kemudian tersambung juga oleh komputer melalui USB. Hasil pengujian ditunjukan pada Gambar 5.5.



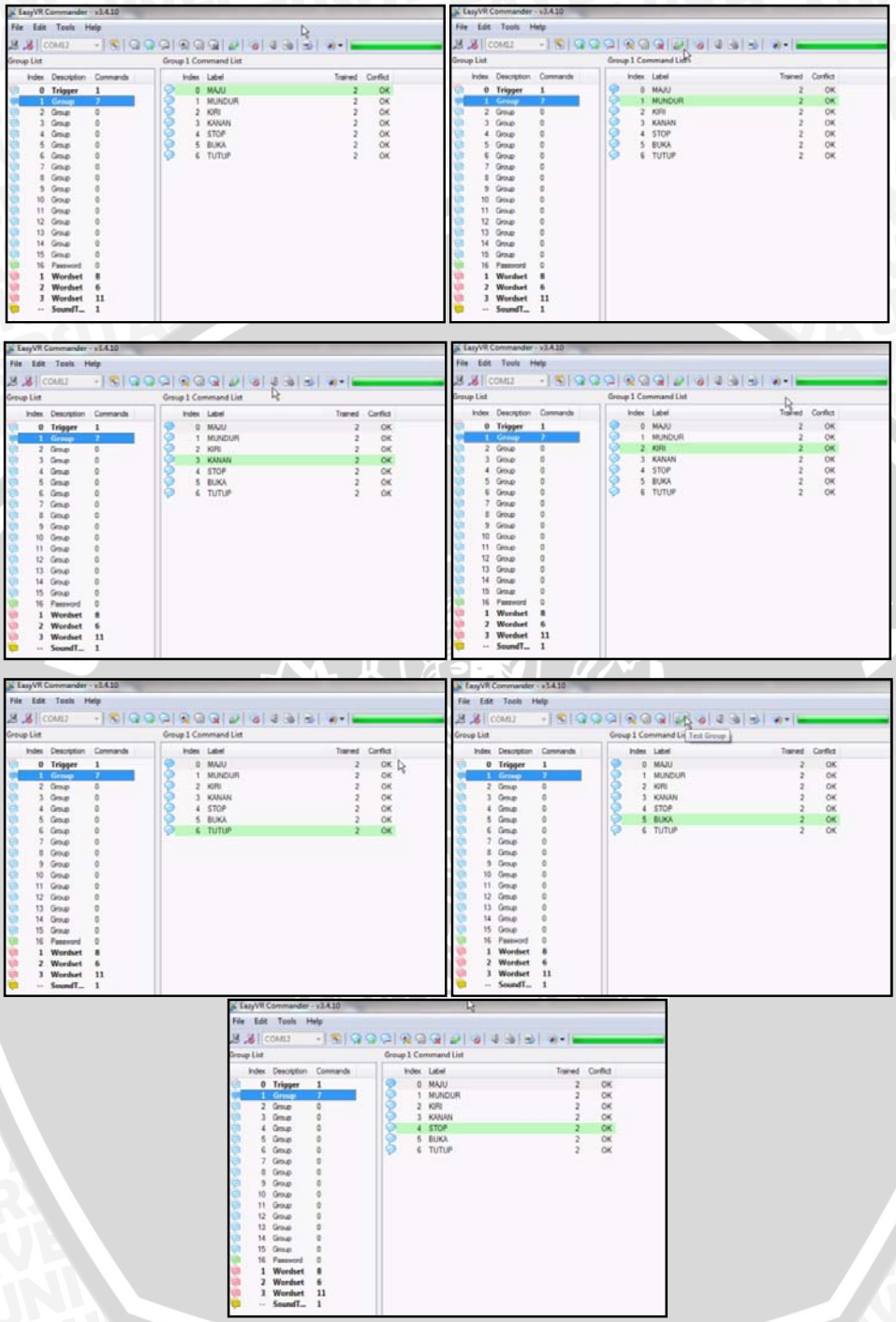
Gambar 5.5. Tampilan Serial Monitor pada *Arduino* untuk pengujian komunikasi serial dengan USB

Berdasarkan hasil pengujian yang telah dilakukan, apabila *EasyVR* tidak terdeteksi maka akan muncul kalimat bertuliskan "*EasyVR* not detected!". Sebaliknya, apabila terdeteksi maka akan muncul kalimat bertuliskan "*EasyVR* detected!".

5.3. Pengujian *Voice Recognition*

5.3.1. Pengujian Hasil *Sampling*

Tujuan dari pengujian ini adalah untuk menguji apakah hasil sampling berupa suara manusia khususnya pengucapan "maju", "mundur", "kanan", "kiri", "stop", "buka", dan "tutup" dapat digunakan dalam penerapan sistem pergerakan robot. Pengujian ini dilakukan dengan cara menghubungkan *EasyVR* ke PC atau laptop menggunakan konektor berupa USB to TTL.



Gambar 5.6. Tampilan pada *EasyVRCommander* saat pengujian setiap hasil sampling.

Gambar 5.6 menunjukkan hasil pengujian sampling. Tanda berwarna hijau menunjukkan bahwa kata yang diucapkan sesuai dengan hasil sampling yang tersimpan dalam modul *EasyVR*.

5.3.2. Pengujian Jarak Ideal Pemberian Perintah Suara Terhadap Jarak Pemberian *Sample* Suara

Tujuan dari pengujian ini adalah untuk mengetahui berapa jarak ideal pada modul *EasyVR* untuk dapat menerima perintah dengan baik jika ditentukan variasi pemberian *sample* suara. Masing-masing perintah diucapkan 10 kali pada setiap jarak yang ditentukan. Sehingga, dapat diketahui keberhasilan pada setiap jarak dan diketahui *range* jarak ideal untuk memberi perintah suara dan memberi *sample* suara. Pegucapan perintah suara baik pemberian *sample* maupun pemberian perintah dilakukan dengan variasi pengucapan yang relatif sama.

Tabel 5.3 Hasil pengujian jarak ideal pemberian perintah suara

Keberhasilan Pemberian Perintah Berdasarkan Jarak Pemberian Sample Suara									
Kata **MAJU**		Jarak Pemberian Perintah							
		1 cm	2 cm	3 cm	4 cm	5 cm	10 cm	50 cm	100 cm
Jarak Pemberian Sample	1 cm	10	10	10	10	0	0	0	0
	2 cm	10	10	10	10	10	0	0	0
	3 cm	0	10	10	10	10	10	10	10
	4 cm	0	10	10	10	10	10	10	10
	5 cm	0	10	10	10	10	10	10	10
	10 cm	0	0	10	10	10	10	10	10
	50 cm	0	0	0	0	10	10	10	10
	100 cm	0	0	0	0	10	10	10	10
Keberhasilan Pemberian Perintah Berdasarkan Jarak Pemberian Sample Suara									
Kata **MUNDUR**		Jarak Pemberian Perintah							
		1 cm	2 cm	3 cm	4 cm	5 cm	10 cm	50 cm	100 cm
Jarak Pemberian Sample	1 cm	10	10	10	10	0	0	0	0
	2 cm	10	10	10	10	10	0	0	0
	3 cm	0	10	10	10	10	10	10	10
	4 cm	0	10	10	10	10	10	10	10
	5 cm	0	10	10	10	10	10	10	10
	10 cm	0	0	10	10	10	10	10	10
	50 cm	0	0	0	0	10	10	10	10
	100 cm	0	0	0	0	10	10	10	10
Keberhasilan Pemberian Perintah Berdasarkan Jarak Pemberian Sample Suara									

Kata **KANAN**		Jarak Pemberian Perintah							
		1 cm	2 cm	3 cm	4 cm	5 cm	10 cm	50 cm	100 cm
Jarak Pemberian Sample	1 cm	10	10	10	10	0	0	0	0
	2 cm	10	10	10	10	10	0	0	0
	3 cm	0	10	10	10	10	10	10	10
	4 cm	0	10	10	10	10	10	10	10
	5 cm	0	10	10	10	10	10	10	10
	10 cm	0	0	10	10	10	10	10	10
	50 cm	0	0	0	0	10	10	10	10
	100 cm	0	0	0	0	10	10	10	10

Keberhasilan Pemberian Perintah Berdasarkan Jarak Pemberian Sample Suara

Kata **KIRI**		Jarak Pemberian Perintah							
		1 cm	2 cm	3 cm	4 cm	5 cm	10 cm	50 cm	100 cm
Jarak Pemberian Sample	1 cm	10	10	10	10	0	0	0	0
	2 cm	10	10	10	10	10	0	0	0
	3 cm	0	10	10	10	10	10	10	10
	4 cm	0	10	10	10	10	10	10	10
	5 cm	0	10	10	10	10	10	10	10
	10 cm	0	0	10	10	10	10	10	10
	50 cm	0	0	0	0	10	10	10	10
	100 cm	0	0	0	0	10	10	10	10

Keberhasilan Pemberian Perintah Berdasarkan Jarak Pemberian Sample Suara

Kata **STOP**		Jarak Pemberian Perintah							
		1 cm	2 cm	3 cm	4 cm	5 cm	10 cm	50 cm	100 cm
Jarak Pemberian Sample	1 cm	10	10	10	10	0	0	0	0
	2 cm	10	10	10	10	10	0	0	0
	3 cm	0	10	10	10	10	10	10	10
	4 cm	0	10	10	10	10	10	10	10
	5 cm	0	10	10	10	10	10	10	10
	10 cm	0	0	10	10	10	10	10	10
	50 cm	0	0	0	0	10	10	10	10
	100 cm	0	0	0	0	10	10	10	10

Keberhasilan Pemberian Perintah Berdasarkan Jarak Pemberian Sample Suara

Kata **BUKA**	Jarak Pemberian Perintah							
	1 cm	2 cm	3 cm	4 cm	5 cm	10 cm	50 cm	100 cm

Jarak Pemberian Sample	1 cm	10	10	10	10	0	0	0	0
	2 cm	10	10	10	10	10	0	0	0
	3 cm	0	10	10	10	10	10	10	10
	4 cm	0	10	10	10	10	10	10	10
	5 cm	0	10	10	10	10	10	10	10
	10 cm	0	0	10	10	10	10	10	10
	50 cm	0	0	0	0	10	10	10	10
	100 cm	0	0	0	0	10	10	10	10

Keberhasilan Pemberian Perintah Berdasarkan Jarak Pemberian Sample Suara

Kata **TUTUP**		Jarak Pemberian Perintah							
		1 cm	2 cm	3 cm	4 cm	5 cm	10 cm	50 cm	100 cm
Jarak Pemberian Sample	1 cm	10	10	10	10	0	0	0	0
	2 cm	10	10	10	10	10	0	0	0
	3 cm	0	10	10	10	10	10	10	10
	4 cm	0	10	10	10	10	10	10	10
	5 cm	0	10	10	10	10	10	10	10
	10 cm	0	0	10	10	10	10	10	10
	50 cm	0	0	0	0	10	10	10	10
	100 cm	0	0	0	0	10	10	10	10

Tabel 5.3 menunjukkan tingkat keberhasilan pemberian perintah suara pada jarak yang ditentukan berdasarkan jarak pemberian *sample* yang ditentukan pula. Hasil pengujian membuktikan tingkat keberhasilan pemberian perintah suara berdasarkan jarak sangat bergantung darijarak pemberian *sample* suara yang diberikan.

5.3.3. Pengujian Pemberian Perintah dari Orang yang Berbeda

Tujuan dari pengujian ini adalah untuk mengetahui tingkat keberhasilan *EasyVR* dalam menerima perintah suara dari orang yang berbeda. Perintah diberikan oleh sepuluhorang dengan etnis Jawa dan berusia rata-rata 20 tahun hingga 23 tahun. Setiap pemberian perintah dilakukan 10 kali per perintah dengan jarak 10 cm dari sensor *microphone*pada setiap orang. Pengujian dilakukan dengan kondisi ideal atau *noise* yang sangat kecil.

Tabel 5.4 Hasil pengujian pemberian perintah dari orang yang berbeda

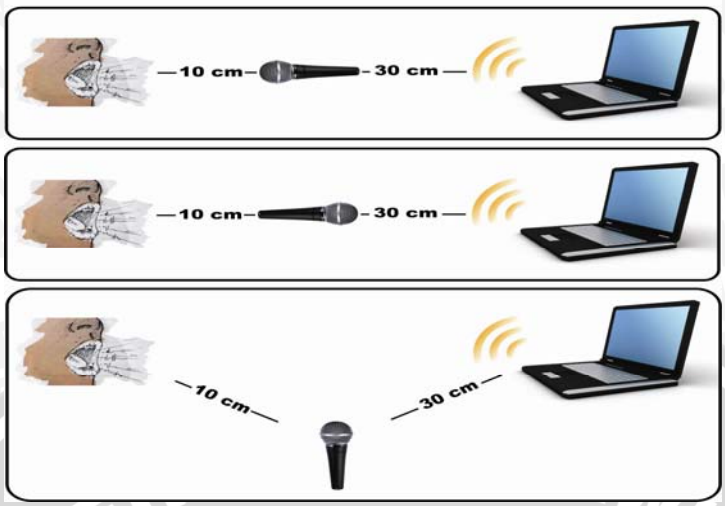
Orang ke-	Jumlah Keberhasilan Pemberian Perintah Suara	% Keberhasilan

	Maju	Mundur	Kanan	Kiri	Buka	Tutup	Stop	
1	10	10	10	10	10	10	10	100 %
2	9	10	9	10	9	10	9	94,2 %
3	10	10	9	10	10	10	10	98,5 %
4	10	9	10	10	10	8	9	94,2 %
5	9	8	10	10	10	9	10	94,2 %
6	9	10	10	9	10	9	10	95,7 %
7	8	9	10	10	10	8	9	91,4 %
8	10	8	9	10	9	9	10	92,8 %
9	9	7	9	10	9	8	10	88,5 %
10	9	9	9	9	10	10	10	94,2 %
Rata-rata %keberhasilan								**94,37 %**

Tabel 5.4 menunjukkan berapa kali perintah harus diucapkan agar *EasyVR* mampu mengolah perintah yang diberikan agar sama dengan hasil sampling. Hasil pengujian menunjukkan tingkat keberhasilan menerima perintah suara dari kondisi yang sudah ditentukan dan pengucapan perintah suara yang memiliki variasi sama dengan sample yang direkam mencapai 94,37 %. Kegagalan dalam pengolahan perintah suara disebabkan oleh pengucapan tutur kata yang kurang jelas dan terlalu besar atau terlalu kecil volume suara yang diberikan.

5.3.4. Pengujian keberhasilan menerima perintah dengan adanya *noise*

Tujuan dari pengujian ini adalah mengetahui tingkat keberhasilan menerima perintah dengan adanya *noise*. *Noise* yang diberikan berupa sebuah rekaman suara yang berisi kondisi bising pada saat Kontes Robot Indonesia Regional IV di gedung Robotika ITS. Jarak antara *noise* dengan *microphone* ditentukan, yaitu 30 cm dan jarak pemberian perintah adalah 10 cm.Pengujian dilakukan dengan posisi hadap *microphone* yang berbeda, yaitu menghadap pemberi perintah, menghadap sumber *noise*, dan diantara pemberi perintah dan sumber *noise*. Ilustrasi pengujian dengan posisi hadap microphone ditunjukkan dalam Gambar 5.7.



Gambar 5.7 Ilustrasi pengujian dengan posisi hadap microphone.

Tabel 5.5 Hasil pengujian keberhasilan menerima perintah dengan adanya *noise* dengan posisi *microphone* menghadap pemberi perintah.

Volume Speaker PC	Sumber Noise (dB)	Keberhasilan menerima perintah terhadap *noise*							% Keberhasilan
		Maju	Mundur	Kanan	Kiri	Stop	Buka	Tutup	
15 %	61 – 69	10	10	10	10	10	10	10	100 %
30 %	66 – 74	10	9	10	10	10	10	10	98,57 %
45 %	75 – 82	10	9	10	9	9	10	9	94,28 %
60%	84 – 90	5	8	7	6	5	7	6	62,85 %
75 %	88 – 102	0	0	0	0	0	0	0	0 %

Tabel 5.6 Hasil pengujian keberhasilan menerima perintah dengan adanya *noise* dengan posisi *microphone* menghadap sumber *noise*.

Volume Speaker PC	Sumber Noise (dB)	Keberhasilan menerima perintah terhadap *noise*							% Keberhasilan
		Maju	Mundur	Kanan	Kiri	Stop	Buka	Tutup	
15 %	61 – 69	10	10	10	10	10	10	10	100 %
30 %	66 – 74	10	9	10	9	10	10	10	98,57 %
45 %	75 – 82	10	9	10	9	9	10	9	94,28 %
60%	84 – 90	5	8	7	6	5	7	6	62,85 %
75 %	88 – 102	0	0	0	0	0	0	0	0 %

Tabel 5.7 Hasil pengujian keberhasilan menerima perintah dengan adanya *noise* dengan posisi *microphone*berada diantara pemberi perintah dan sumber *noise*.

Volume Speaker PC	Sumber Noise (dB)	Keberhasilan menerima perintah terhadap *noise*							% Keberha-silan
		Maju	Mundur	Kanan	Kiri	Stop	Buka	Tutup	
15 %	61 – 69	6	7	5	6	5	7	7	60 %
30 %	66 – 74	3	5	4	4	5	5	6	45,71 %
45 %	75 – 82	0	1	0	0	0	2	0	4,28 %
60%	84 – 90	0	0	0	0	0	0	0	0 %
75 %	88 – 102	0	0	0	0	0	0	0	0 %

Tabel 5.5, tabel 5.6, dan tabel 5.7 menunjukkan tingkat keberhasilan menerima perintah dengan adanya *noise* dengan posisi hadap *microphone* yang berbeda. Kegagalan dalam menerima perintah disebabkan oleh besarnya noise dalam keadaan tertentu. Jika dilihat dalam dapat disimpulkan bahwa semakin besar *noise* yang diberikan, maka akan semakin kecil tingkat keberhasilan dalam menerima perintah suara.

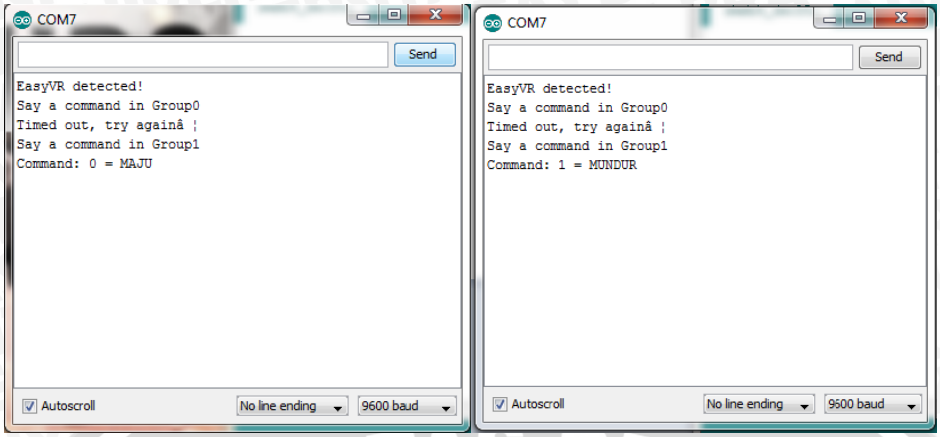
5.4. Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dibagi menjadi dua bagian, yaitu:

1. Pengujian melalui Serial Monitor dalam program *Arduino*.
2. Pengujian respon robot setelah menerima perintah dalam satuan waktu.

5.4.1. Pengujian Melalui Serial Monitor pada Program *Arduino*

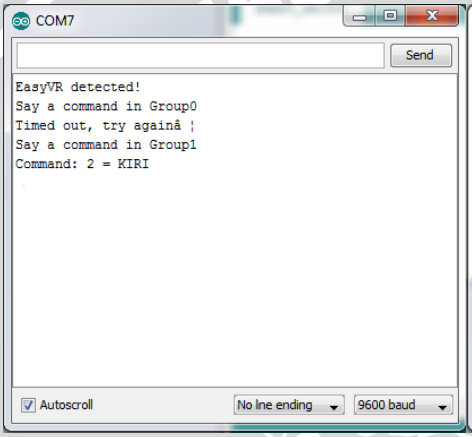
Tujuan dari pengujian ini adalah untuk mengetahui apakah board *ArduinoUNO* mampu menerima data ASCII dari *EasyVR* dan mengeksekusi data tersebut. Pengujian ini dilakukan dengan cara menghubungkan *ArduinoUNO* yang sudah terpasang dengan perancangan system secara keseluruhan dengan PC atau laptop melalui kabel USB. Hasil pengujian dapat dilihat melalui Serial Monitor yang terdapat pada program *Arduino*.


COM7
Send
EasyVR detected!
Say a command in Group0
Timed out, try againâ ¦
Say a command in Group1
Command: 0 = MAJU
Autoscroll
No line ending
9600 baud

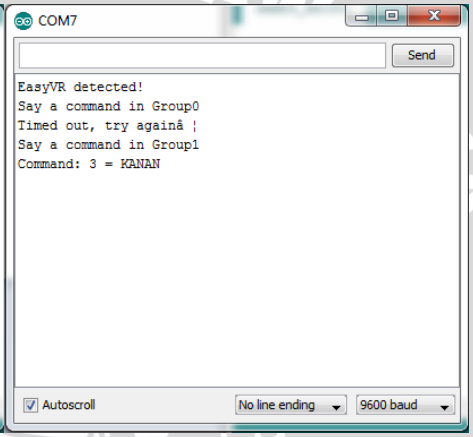
(a)

COM7
Send
EasyVR detected!
Say a command in Group0
Timed out, try againâ ¦
Say a command in Group1
Command: 1 = MUNDUR
Autoscroll
No line ending
9600 baud

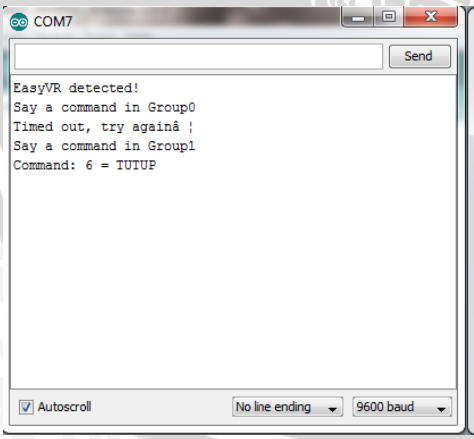
(b)


COM7
Send
EasyVR detected!
Say a command in Group0
Timed out, try againâ ¦
Say a command in Group1
Command: 2 = KIRI
Autoscroll
No line ending
9600 baud

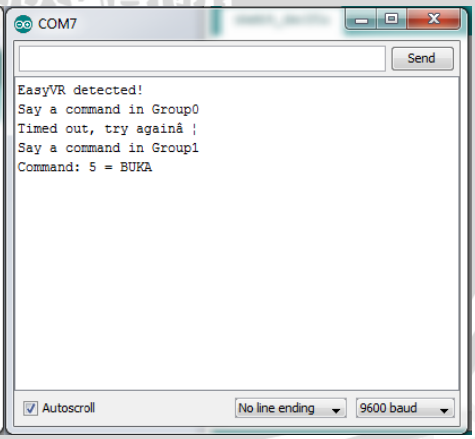
(c)


COM7
Send
EasyVR detected!
Say a command in Group0
Timed out, try againâ ¦
Say a command in Group1
Command: 3 = KANAN
Autoscroll
No line ending
9600 baud

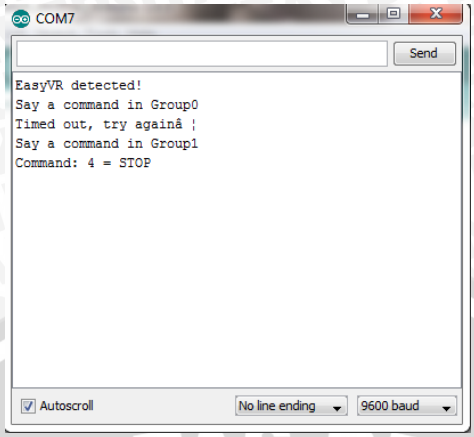
(d)


COM7
Send
EasyVR detected!
Say a command in Group0
Timed out, try againâ ¦
Say a command in Group1
Command: 6 = TUTUP
Autoscroll
No line ending
9600 baud

(e)


COM7
Send
EasyVR detected!
Say a command in Group0
Timed out, try againâ ¦
Say a command in Group1
Command: 5 = BUKA
Autoscroll
No line ending
9600 baud

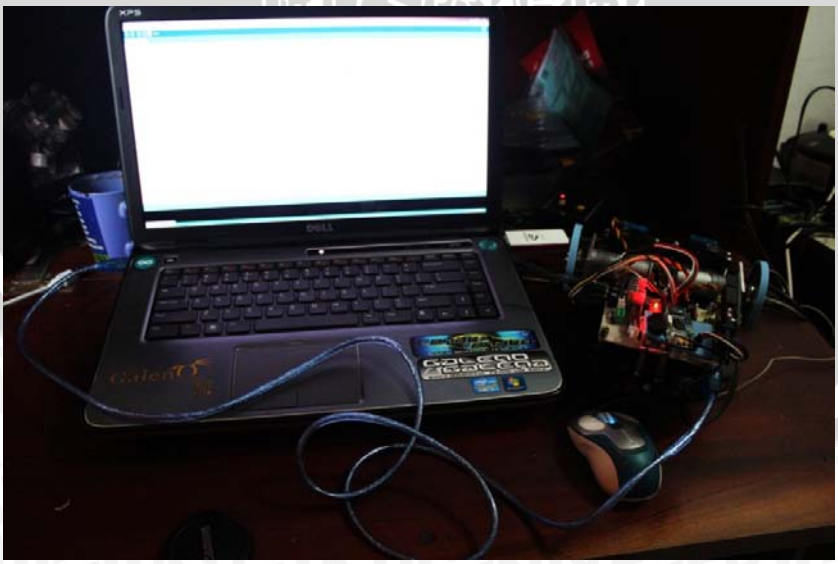
(f)



(g)

Gambar 5.7. Tampilan pada Serial Monitor untuk pengujian navigasi (a) robot berjalan ke depan, (b) robot berjalan ke belakang, (c) robot belok ke kanan, (d) robot belok ke kiri, (e) robot membuka capit, (f) robot menutup capit, (g) robot berhenti.

Gambar 5.7 menunjukkan respons dari setiap kata yang diucapkan. Dari hasil pengujian terlihat bahwa robot mampu menerima perintah berupa suara. Terbukti dari setiap pengucapan kata sesuai dengan command yang diberikan. Gambar 5.8 menunjukkan respons robot terhadap kata "maju" yang berarti bernavigasi jalan ke depan atau maju dengan adanya tanda yaitu LED berwarna merah menyala.



Gambar 5.8. Pengujian melalui Serial Monitor pada program *Arduino*.

5.4.2. Pengujian Respon Robot Setelah Menerima Perintah dalam Satuan Waktu

Tujuan dari pengujian ini adalah untuk mengetahui berapa lama respons robot melakukan pergerakan atau navigasi terhadap perintah suara yang diberikan. Pengujian dilakukan dengan cara manual yaitu dengan menggunakan jam analog. Perhitungan waktu dimulai setelah perintah selesai diucapkan.

Tabel 5.8 Hasil pengujian rata-rata waktu respon robot setelah menerima perintah.

Perintah	Waktu Respon
Maju	≤ 1 detik
Mundur	≤ 1 detik
Kanan	≤ 1 detik
Kiri	≤ 1 detik
Buka	≤ 1 detik
Tutup	≤ 1 detik
Stop	≤ 1 detik

Tabel 5.8 menunjukkan bahwa rata-rata waktu respon dari setelah perintah diucapkan sampai pergerakan atau navigasi robot adalah kurang dari sama dengan 1 detik. Hal ini disebabkan berjalannyaproses ketika *microphone* menerima perintah yang kemudian dicocokkan oleh *EasyVR* apakah perintah yang diberikan sesuai dengan sample. Jika cocok, *EasyVR* mengirim perintah ke *Arduino UNO*yang kemudian akan mengakses aktuator melalui *driver* motor L298N.Hal ini cukup bagus karena tidak membutuhkan waktu terlalu lama agar data dapat dieksekusi.

BAB VI
PENUTUP

6.1. Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1) Metode pengambilan *sample* suara pada modul *EasyVR*dilakukan dalam kondisi ideal atau tanpa adanya *noise* dan dilakukan sebanyak dua kali pengambilan suara dengan variasi pengucapan yang relatif sama pada setiap kata. Hal ini dilakukan sesuai dengan kemampuan *EasyVR* yang tidak bisa menerima pengucapan variasi suara kedua jika berbeda dengan variasi pengucapan suara pertama. Oleh karena itu, agar *EasyVR* dapat berfungsi atau dapat mengakses perintah suara, dibutuhkan variasi suara yang relatif sama dengan *sample*. Selain itu, *EasyVR commander* memberi batas waktu selama 5 detik untuk setiap pengucapan suara yang akan dijadikan *sample*.
2) Modul *EasyVR* memiliki tingkat keberhasilan sebesar 94,37 % jika menerima perintah suara dari orang yang berbeda dengan variasi pengucapan yang relatif sama. Kegagalan dalam pengolahan perintah suara disebabkan oleh pengucapan tutur kata yang kurang jelas dan terlalu besar atau terlalu kecil volume suara yang diberikan.
3) Keberhasilan modul EasyVR dalam menerima perintah suara dengan baik segi jarak ditentukan oleh jarak pemberian perintah suara yang sangat bergantung pada jarak pemberian *sample*.
4) Rata-rata *waktu respon robot* setelah pengucapan perintah ke pergerakan atau navigasi robot adalah kurang dari sama dengan 1 detik. Hal ini cukup bagus karena tidak membutuhkan terlalu waktu lama agar data dapat dieksekusi.
5) Penerapan sistem *voice recognition* pada robot pemindah objek dapat digunakan sebagai sistem navigasi robotmeskipun belum mampu menggantikan penggunaan perangkat berupa *joystick*.

6.2. Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah sebagai berikut:

1) Sistem*voice recognition* sangat bergantung pada variasi *sample* yang direkam. Oleh karena itu, disarankan untuk perancangan pengambilan sample dengan variasi pengucapan dan variasi jarak yang lebih banyak.
2) Jarak terjauh modul *EasyVR* untuk mampu menerima perintah suara dengan bagus hanya 100 cm. Jika ingin jarak yang lebih jauh disarankan menggunakan sistem tambahan atau pembantu berupa *wifi*.

DAFTAR PUSTAKA

Aditya, Rezza. 2012. *Prototipe Pengenalan Suara Sebagai Penggerak Dinamo Starter Pada Mobil*. Depok: Universitas Gunadarma.

Akbar, Arnas Elmiawan. 2013. *Implementasi Sistem Navigasi Wall Following Menggunakan Kontroler PID dengan Metode Tuning pada Robot Kontes Robot Cerdas Indonesia (KRCI) Divisi Senior Beroda*. Malang. Skripsi Jurusan Teknik Elektro FT-UB.

Arduino Development Environment. http://*arduino*.cc/en/Guide/Environment. Diakses tanggal 27 Juli 2013.

ArduinoUno. http://*arduino*.cc/en/Main/*Arduino*Board*Uno*. Diakses tanggal 27 Juli 2013.

Banzi, Massimo. 2008. *Getting Started with Arduino*. Sebastopol:O'Reilly Media.

User Manual EasyVR.http://www.veear.eu/. Diunduh tanggal: 15 Juni 2013. Austria: TIGAL KG.

Fezzari, Mohamed. 2009. *New Speech Processor and Ultrasonic Sensors Based Embedded System to Improve the Control of a Motorised Wheelchair*. Annaba: Science Research of University of Annaba.

Siegwart, Roland dan Illah R. Nourbakhsh. 2004. *Introduction to Autonomous Mobile robots*. London: MIT Press.

Soebhakti, Hendawan. 2008. *AVR Application Note*. Erlangga. Jakarta.

Tara, Rayi Yanu. 2005. *Desain dan Implementasi Logika Fuzzy sebagai Sistem Navigasi Wall Following pada Mobile Robot KRCI*. Malang: Skripsi Jurusan Teknik Elektro FT-UB.

Taufik, Ahmad Sulkhan. 2013. *Sistem Navigasi Waypoint pada Autonomous Mobile Robot*. Malang: Skripsi Jurusan Teknik Elektro FT-UB.

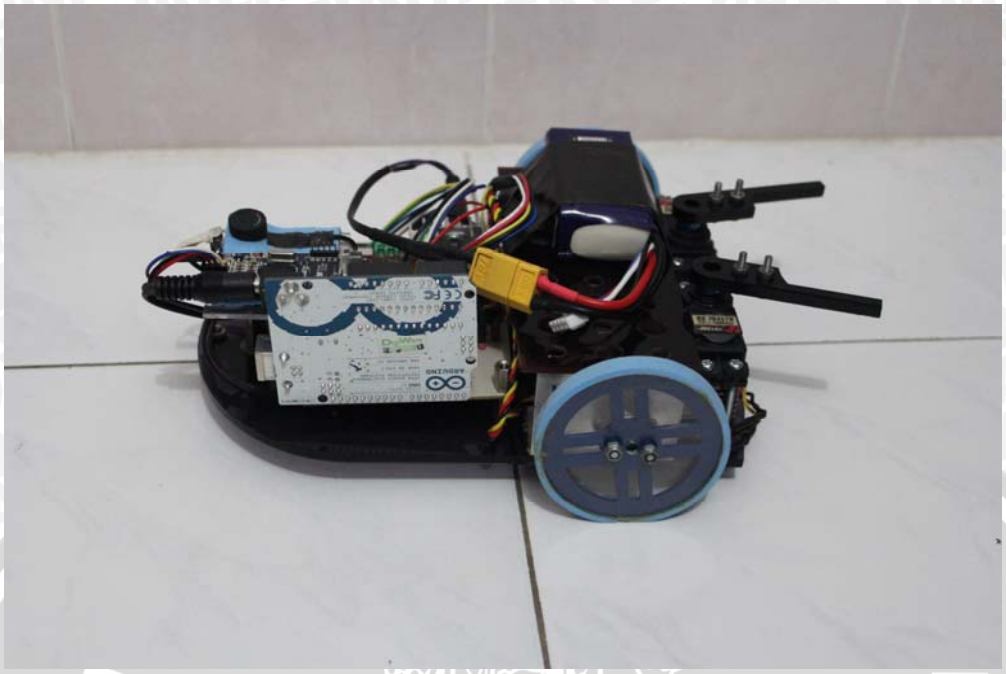
Utomo, Aryo Baskoro dkk. 2007. *Analisis Karakteristik Suara Manusia Berdasarkan Frekuensi Fundamental dan Tingkat Usia pada Pelajar SLTP dan SMA.* Semarang: Skripsi Jurusan Teknik Elektro UNDIP.

Wheat, Dale. 2011. *Arduino Internals*. New York: Apress.

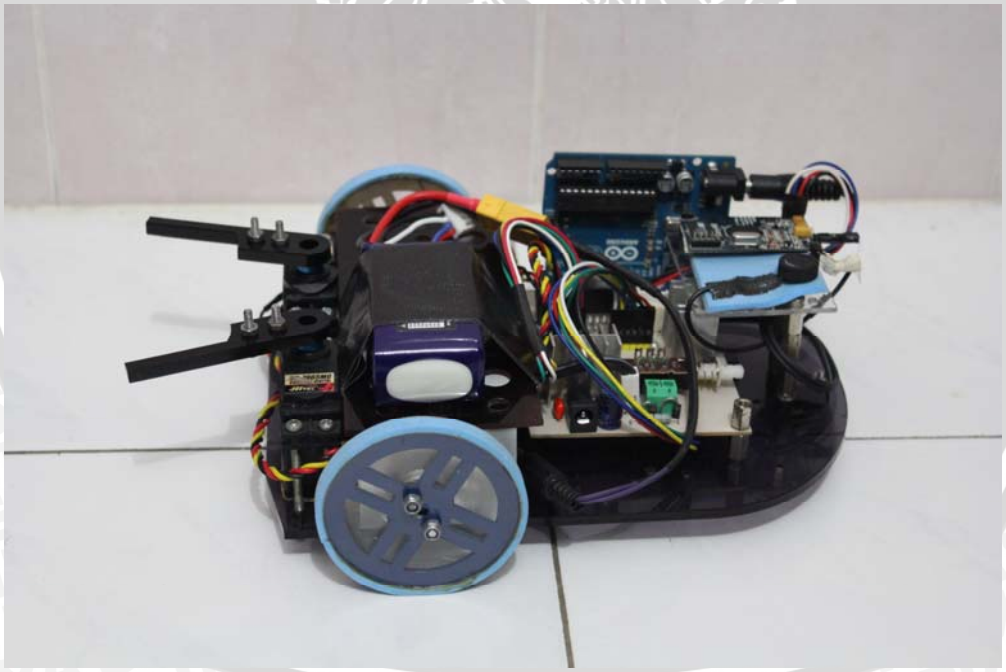
Zulkarnain, Muhammad Yusuf. 2011. *Sistem Pemetaan Posisi pada Robot Kontes Robot Cerdas Indonesia (KRCI) Divisi Senior Beroda dengan Metode Decision Tree*. Malang: Skripsi Jurusan Teknik Elektro FT-UB.

LAMPIRAN 1

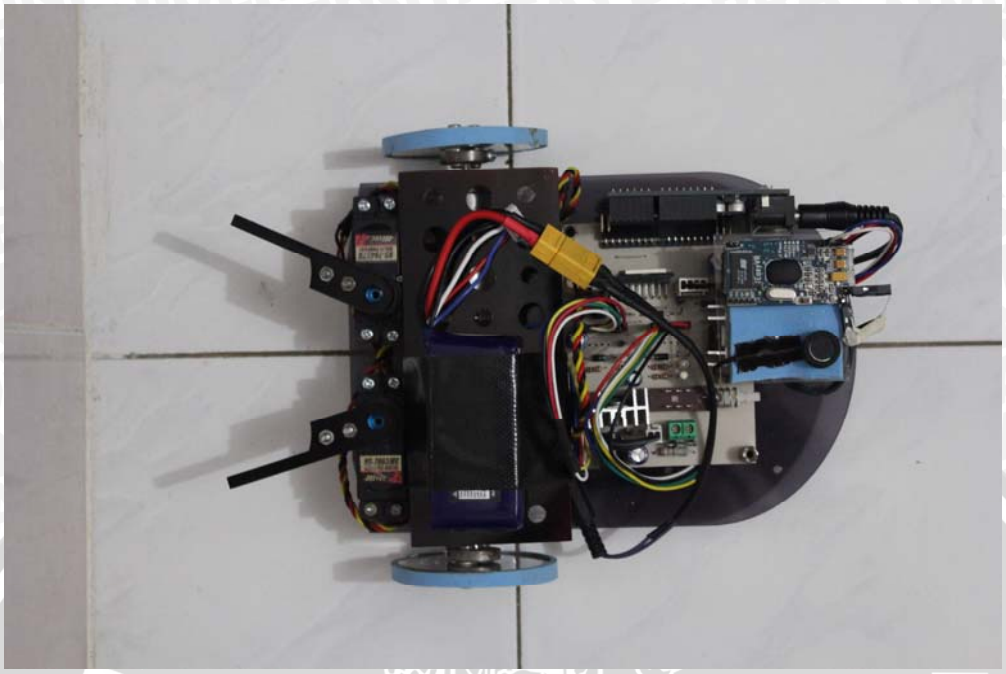
FOTO ALAT



Gambar 1. Perspektif KananRobot



Gambar 2. Perspektif Kiri Robot



Gambar 4. RobotTampak Atas

Gambar 5. Perspektif Belakang Robot

LAMPIRAN 2

***LISTING* PROGRAM**

```
#include <Servo.h>
   Servo myservo1;                    // variabel untuk servo
   Servo myservo2;                    // variabel untuk servo

   int M1=4;                          // Port keluaran untuk motor kanan mundur
   int M2=5;                          // Port keluaran untuk motor kanan maju
   int E1=9;                          // Port enable pada L298N untuk motor kanan
   int M3=6;                          // Port keluaran untuk motor kiri maju
   int M4=7;                          // Port keluaran untuk motor kiri mundur
   int E2=10;                         // Port enable pada L298N untuk motor kiri

   #if defined(ARDUINO)&&ARDUINO>= 100
     #include "Arduino.h"
     #include "SoftwareSerial.h"
     SoftwareSerial port(12,13);
   #else
     #include "WProgram.h"
     #include "NewSoftSerial.h"
     NewSoftSerial port(12,13);
   #endif
   #include "EasyVR.h"

EasyVREasyVR(port);

   enum Groups
   {
   GROUP_1  = 1,
   };

   enum Group0
   {
     G0_ARDUINO = 0,
```

```
};

enum Group1
{
 G1_MAJU = 0,
 G1_MUNDUR = 1,
 G1_KIRI = 2,
 G1_KANAN = 3,
 G1_STOP = 4,
 G1_BUKA = 5,
 G1_TUTUP = 6,
};

EasyVRBridge bridge;

int8_t group, idx;

void setup()
{
 pinMode(M1, OUTPUT);
 pinMode(M2, OUTPUT);
 pinMode(E1, OUTPUT);
 pinMode(M3, OUTPUT);
 pinMode(M4, OUTPUT);
 pinMode(E2, OUTPUT);

 myservo1.attach(2);
 myservo2.attach(3);
 if (bridge.check())
 {
  cli();
  bridge.loop(0, 1, 12, 13);
 }
 Serial.begin(9600);
```

// Menggabungkan variabel ke dalam grup 1

// Inisialisasi bridge mode

// Inisialisasi port serial

```
    port.begin(9600);                       // Memulai komunikasi

    if (!EasyVR.detect())                   // Inisialisasi pendeteksi EasyVR
    {
      Serial.println("EasyVR not detected!");
      for (;;);                             // Menunggu sampai EasyVR terdeteksi
    }
EasyVR.setPinOutput(EasyVR::IO1, LOW);
    Serial.println("EasyVR detected!");
EasyVR.setTimeout(5);
EasyVR.setLanguage(EasyVR::ENGLISH);

    group = EasyVR::TRIGGER;

  }
  void action();

  void loop()
  {
EasyVR.setPinOutput(EasyVR::IO1, HIGH);     // Memulai pengenalan suara

    Serial.print("Say a command in Group");  // Tampilan untuk memulai pemberian perintah suara
    Serial.println(group);
EasyVR.recognizeCommand(group);             // Pengenalan perintah suara

    do
    {

    }
    while (!EasyVR.hasFinished());

EasyVR.setPinOutput(EasyVR::IO1, LOW);      // Mengakhiri pengenalan suara
    if (idx >= 0)
```

```
{
    return;
    }
    if (idx >= 0)
    {
     uint8_t train = 0;                                    // Menampilkan command,idx & name
     char name[32];
     Serial.print("Command: ");
     Serial.print(idx);                                    // Menampilkan kode ASCII dari perintah
     if (EasyVR.dumpCommand(group, idx, name, train))
     {
      Serial.print(" = ");
      Serial.println(name);
     }
     else
      Serial.println();
EasyVR.playSound(0, EasyVR::VOL_FULL);
     action();
    }
    else                                                   // Tampilan errors number atau timeout
    {
     if (EasyVR.isTimeout())
      Serial.println("Timed out, try again…");
     int16_t err = EasyVR.getError();
     if (err >= 0)
     {
      Serial.print("Error ");
      Serial.println(err, HEX);
     }
     group = GROUP_1;
    }
  }
```

```
void action()
{
   switch (group)
     {
   case GROUP_1:
   switch (idx)
     {
      case G1_MAJU:
        Serial.print(" ");

        digitalWrite(M1, LOW);
        digitalWrite(M2, HIGH);
        analogWrite(E1, 255);

        digitalWrite(M3, HIGH);
        digitalWrite(M4, LOW);
        analogWrite(E2,255);
        break;

      case G1_MUNDUR:
        Serial.print(" ");

        digitalWrite(M1, HIGH);
        digitalWrite(M2, LOW);
        analogWrite(E1, 255);

        digitalWrite(M3, LOW);
        digitalWrite(M4, HIGH);
        analogWrite(E2,255);
        break;

      case G1_KIRI:
        Serial.print(" ");
```

```
  digitalWrite(M1, LOW);
  digitalWrite(M2, HIGH);
  analogWrite(E1, 255);

  digitalWrite(M3, HIGH);
  digitalWrite(M4, LOW);
  analogWrite(E2,0);
  break;

case G1_KANAN:
  Serial.print(" ");

  digitalWrite(M1, LOW);
  digitalWrite(M2, HIGH);
  analogWrite(E1, 0);

  digitalWrite(M3, HIGH);
  digitalWrite(M4, LOW);
  analogWrite(E2,255);
  break;
case G1_STOP:
  Serial.print(" ");

  digitalWrite(M1, LOW);
  digitalWrite(M2, LOW);
  analogWrite(E1, 0);

  digitalWrite(M3, LOW);
  digitalWrite(M4, LOW);
  analogWrite(E2,0);
  break;
case G1_BUKA:
  Serial.print(" ");
```

```
 myservo1.write(0);
        myservo2.write(90);
        delay(5000);
        break;

      case G1_TUTUP:
        Serial.print(" ");

    myservo1.write(45);
    myservo2.write(45);
        delay(5000);
        break;
     }
     break;
    }
}
```

LAMPIRAN 3

DATASHEET