

2.3. Teori Pendukung

Pada bagian ini dijelaskan mengenai teori pendukung seperti konsep, metoda, rumus perhitungan yang digunakan untuk merealisasikan sistem yang dibuat dalam proyek akhir ini.

2.3.1. Pengenalan Ucapan (*Speech Recognition*)

Sistem Pengenalan Ucapan (*Speech Recognition System*) adalah sistem yang berfungsi untuk mengubah bahasa lisan menjadi bahasa tulisan. Masukan sistem adalah ucapan manusia, selanjutnya sistem akan mengidentifikasi kata atau kalimat yang diucapkan dan menghasilkan teks yang sesuai dengan apa yang diucapkan. Sinyal ucapan pertama kali akan dilewatkan pada bagian penganalisis ucapan untuk mendapatkan besaran-besaran atau ciri-ciri yang mudah diolah pada tahap berikutnya. Untuk setiap ucapan yang berbeda akan dihasilkan pola ciri yang berbeda.

Penganalisis sintaks akan melakukan transformasi sinyal ucapan dari domain waktu ke domain frekuensi. Pada domain frekuensi, untuk kurun waktu yang singkat, setiap sinyal dapat terlihat memiliki ciri-ciri yang unik. Namun demikian, pengucapan suatu unit bunyi ucapan (fonem) seringkali bervariasi antar orang yang berbeda, juga terpengaruh oleh fonem-fonem disekitarnya, kondisi emosi, noise, dan faktor-faktor lainnya. Sistem *Speech Recognition* akan melakukan pengenalan untuk setiap unit bunyi pembentuk ucapan (fonem), selanjutnya mencoba mencari kemungkinan kombinasi hasil ucapan yang paling dapat diterima. Sistem yang lebih sederhana adalah sistem yang hanya dapat mengenal sejumlah kata yang jumlahnya terbatas. Sistem ini biasanya lebih akurat dan lebih mudah dilatih, tetapi tidak dapat mengenal kata yang berada di luar kosakata yang pernah diajarkan. Ada 2 tipe *Speech Recognition*, dilihat dari ketergantungan pembicara yaitu:

a. *Independent Speech Recognition*

Independent Speech Recognition yaitu sistem pengenalan ucapan tanpa terpengaruh dengan siapa yang berbicara, tetapi mempunyai keterbatasan dalam jumlah kosakata. Model ini akan mencocokkan setiap ucapan dengan kata yang dikenali dan memilih yang "sepertinya" cocok. Untuk mendapatkan kecocokan kata yang diucapkan maka digunakan model statistik yang dikenal dengan nama *Hidden Markov Model (HMM)*

b. *Dependent Speech Recognition*

Dependent Speech Recognition yaitu sistem pengenalan ucapan yang memerlukan pelatihan khusus dari pembicara, dimana hasil pelatihan dari masing-masing pembicara akan disimpan dalam sebuah profil. Profil inilah yang nantinya digunakan untuk berinteraksi dengan sistem pengenalan ucapan dan sistem akan bergantung siapa yang berbicara. Sistem ini biasanya lebih mudah

untuk dikembangkan, dimana contoh suara sudah dibuat sebelumnya dan disimpan dalam database (basis data) dan jumlah kosakatanya lebih besar dibandingkan dengan independent speech recognition. Proses pengenalan ucapan dengan cara membandingkan ucapan pembicara dengan contoh suara yang sudah ada. [6]

2.3.2. Google Speech API pada Raspberry Pi

Google sebagai mesin pencarian terbesar dan nomor satu di dunia yang terus mengembangkan layanannya. Tidak hanya sebagai mesin pencarian di internet, Google juga mempunyai banyak produk dan layanan untuk pengguna seperti Search, GMail, Map dan Android. Raksasa internet ini juga punya beberapa tool, library dan API (Application Programming Interface) untuk para pengembang aplikasi (developer) seperti Google Search API, Google Map API dan Google Blogger API. Pengembangan ini meliputi pengembangan layanan, salah satunya adalah layanan *Machine Learning Model*. Dengan adanya pengembangan layanan *Machine Learning Model*, yaitu penggabungan *Google Translate API* dan *Cloud Vision API*, menjadi *Google Cloud Speech API* maka dapat dimanfaatkan sebagai pengembangan media pembelajaran Bahasa Inggris menggunakan teknologi *speech recognition*.

Google Speech API atau *Google Voice Search* diluncurkan di Amerika Serikat pada tahun 2008. *Google Speech API* merupakan sebuah framework yang dikembangkan oleh Google untuk mengenali suara kemudian diubah dalam bentuk teks. Proses Pengenalan suara dilakukan pada server Google menggunakan algoritma *Hidden Markov Model* (HMM). Prinsip kerja secara umum dari *Google Speech API* ini yaitu input suara yang diterima oleh perangkat dikirimkan ke server Google, yang selanjutnya server Google melakukan proses pengenalan dan mengubahnya menjadi teks. Algoritma yang digunakan untuk mengenali suara yaitu HMM (*Hidden Markov Model*). [7]

2.3.3. Fuzzy Logic

Dalam kondisi yang nyata, beberapa aspek dalam dunia nyata selalu atau biasanya berada di luar model matematis dan bersifat samar. Konsep ketidakpastian inilah yang menjadi konsep dasar munculnya konsep logika fuzzy. Pencetus gagasan logika fuzzy adalah L.A. Zadeh (1965) dari California University.

Logika fuzzy adalah salah satu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Logika fuzzy berbeda dengan logika digital biasa, dimana logika digital biasa hanya mengenal 2 keadaan yaitu: “ya” dan “tidak” atau “on” dan “off” atau “high” dan “low” atau “1” dan “0”. Sedangkan

logika fuzzy meniru cara berpikir manusia dengan menggunakan konsep sifat kesamaran suatu nilai. Dengan himpunan fuzzy, suatu objek dapat menjadi anggota dari banyak himpunan dengan derajat keanggotaan yang berbeda dalam masing-masing himpunan.

2.3.3.1. Himpunan *Fuzzy*

Pada himpunan tegas (*Crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang ditulis dengan $\mu_A[x]$ memiliki 2 kemungkinan, yaitu:

- a. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- b. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Sedangkan pada himpunan *fuzzy*, nilai keanggotaan terletak pada rentang 0 sampai 1. Dengan artian, apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A[x] = 0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A[x] = 1$ berarti x menjadi anggota penuh pada himpunan A .

Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan *real* pada interval $[0,1]$. Nilai keanggotaannya menunjukkan bahwa suatu item tidak hanya bernilai benar atau salah. Tetapi masih ada nilai-nilai yang terletak antara benar dan salah.

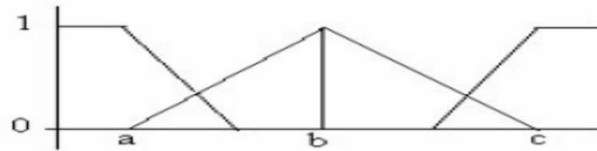
2.3.3.2. Fungsi Keanggotaan

Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut sebagai derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

Terdapat dua definisi fungsi keanggotaan (*membership function*) untuk himpunan *fuzzy* yaitu numerik dan fungsional. Numerik didefinisikan sebagai pernyataan tingkat dari fungsi keanggotaan dari himpunan *fuzzy* dinyatakan dengan vektor bilangan. Sedangkan *functional* didefinisikan dengan menentukan fungsi keanggotaan dari himpunan *fuzzy* dalam pernyataan analitik yang menyatakan tingkat keanggotaan untuk setiap elemen yang ditentukan pada himpunan universal. Berikut merupakan contoh Fungsi keanggotaan:

$$\text{Fungsi segitiga } T(u; a; b; c) = \begin{cases} 0 & \text{for } u < a \\ \frac{u-a}{b-a} & \text{for } a \leq u \leq b \\ \frac{c-u}{c-b} & \text{for } b \leq u \leq c \\ 0 & \text{for } u > c \end{cases}$$

Persamaan fungsi segitiga tersebut, ditunjukkan dalam bentuk grafik pada Gambar 2.6 berikut.



Gambar 1. 1. Grafik Fungsi Segitiga

2.3.3.3. Cara Kerja Kontrol Logika Fuzzy

Dalam sistem kontrol logika fuzzy, terdapat beberapa tahapan proses operasional, diantaranya yaitu:

1. Fuzzifikasi

Fuzzifikasi adalah suatu proses pengubahan nilai tegas (*crisp/real*) yang ada kedalam fungsi keanggotaan.

2. Penalaran (*Inference Machine*)

Penalaran yaitu proses implikasi dalam menalar nilai masukan untuk menentukan nilai keluaran sebagai bentuk pengambilan keputusan.

3. Aturan Dasar (*Rule Based*)

Aturan Dasar adalah suatu bentuk aturan relasi/implikasi “Jika-Maka” atau “*If-Then*”. Misalnya pada pernyataan berikut:

[R1] *IF* (x is A1) *and* (y is B2) *THEN* (Z is C1).

4. Defuzzifikasi

Defuzzifikasi merupakan proses pemetaan himpunan fuzzy kedalam bentuk himpunan tegas (*crisp*). Kerangka operasional kontrol logika fuzzy ditunjukkan pada Gambar 2.7 berikut: [8]

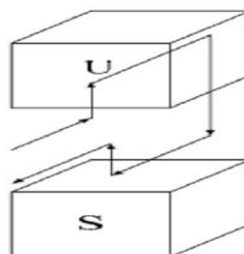


Gambar 2.2. Kerangka Operasional Kontrol Logika Fuzzy

2.3.4. Motor DC

Motor DC merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik dengan menggunakan arus searah. Pada motor dc kumparan medan disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar).

Daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus berfungsi sebagai tempat berlangsungnya dua proses perubahan energi dan daerah tersebut dapat dilihat pada Gambar 2.2:



Gambar 2.3. Prinsip Kerja Motor DC

Mekanisme kerja untuk seluruh jenis motor listrik pada umumnya adalah sama, yaitu:

- a. Arus listrik dalam medan magnet akan memberikan gaya.

- b. Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran /*loop*, maka kedua sisi *loop*, yaitu sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- c. Pasangan gaya menghasilkan tenaga putar/torsi untuk memutar kumparan. Beberapa macam motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan. [9]

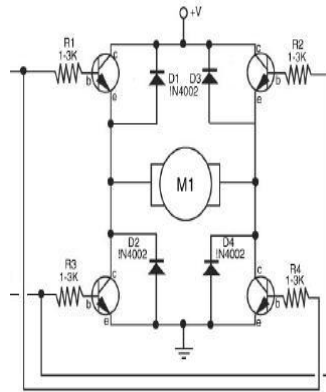
2.3.5. PWM (*Pulse Width Modulation*)

Modulasi merupakan suatu proses dimana parameter gelombang pembawa (*carrier signal*) frekuensi tinggi diubah sesuai dengan salah satu parameter sinyal informasi/pesan. Dengan modulasi, sinyal informasi diubah menjadi pulsa-pulsa persegi dengan frekuensi dan amplitudo tetap tapi dengan lebar pulsa sebanding dengan *amplitude* sinyal informasi.

PWM merupakan salah satu teknik modulasi pulsa untuk memanipulasi lebar pulsa dalam suatu periode yang digunakan untuk mentransfer data pada telekomunikasi atau mengatur tegangan sumber yang konstan untuk mendapatkan tegangan rata-rata berbeda sehingga dapat digunakan untuk mengatur kecepatan motor. Semakin besar perbandingan lama sinyal *high* dengan periode sinyal maka semakin cepat motor berputar. Sinyal PWM dapat dibangun dengan banyak cara, dapat menggunakan metode analog menggunakan rangkaian *op-amp* atau dengan menggunakan metode *digital*. Dengan metode analog setiap perubahan PWM-nya sangat halus, sedangkan menggunakan metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Misalkan PWM digital 8 bit berarti PWM tersebut memiliki resolusi $2^8 = 256$, maksudnya nilai keluaran PWM ini memiliki 256 variasi, variasinya mulai dari 0 – 255 yang mewakili *duty cycle* 0 – 100% dari keluaran PWM tersebut. [10]

2.3.6. Driver Motor DC *H-Bridge*

Driver motor DC *H-bridge* adalah rangkaian yang tersusun dari transistor yang digunakan untuk menggerakkan motor DC. Rangkaian *driver* motor ini dapat mengendalikan arah putaran motor DC dalam 2 arah dan dapat dikontrol dengan metode PWM (*Pulse Width Modulation*) maupun metode sinyal logika dasar TTL (*High*) and (*Low*). Untuk pengendalian motor DC dengan metode PWM maka dengan rangkaian driver motor ini kecepatan putaran motor DC dapat dikendalikan dengan baik. Apabila menggunakan metode logika TTL 0 dan 1, maka rangkaian ini hanya dapat mengendalikan arah putaran motor DC saja dengan putaran motor DC maksimum.



Gambar 2.4. Rangkaian Driver Motor H-Bridge

Rangkaian *driver* motor DC *H-Bridge* ini menggunakan rangkaian jembatan transistor 4 unit dengan *proteksi impuls* tegangan induksi motor DC berupa dioda yang dipasang paralel dengan masing-masing transistor secara *reverse bias*. [11]

2.3.7. Wall Following

Wall Following adalah suatu metode untuk menyediakan orientasi navigasi kepada robot dengan menyusuri dinding. Salah satu keuntungannya adalah tidak perlu adanya garis penuntun ataupun suatu tanda khusus sebagai arahan bagi robot. Cara kerjanya adalah dengan mengatur jarak dinding dengan robot tetap konstan. Bila terjadi perubahan, maka robot akan bergerak untuk menyesuaikan jarak lagi. Proses ini akan dilakukan secara berulang – ulang^[21]. Ada dua metode dari *wall following*:

a. Left Wall Following

Pada metode ini, robot akan menggunakan sensor sebelah kiri untuk menghindari dinding yang ada di sebelah kiri robot. Pada saat robot terlalu dekat dengan dinding sebelah kiri, maka motor kiri akan dipercepat dan motor kanan akan diperlambat, sehingga robot akan menjauhi dinding kiri. Sedangkan apabila robot terlalu jauh dengan dinding kiri, maka motor sebelah kanan akan dipercepat, dan motor sebelah kiri akan diperlambat, sehingga robot akan berjalan mendekati dinding kiri^[21].

b. Right Wall Following

Pada metode ini, robot akan menggunakan sensor sebelah kanan untuk menghindari dinding yang ada di sebelah kanan robot. Pada saat robot terlalu dekat dengan dinding sebelah kanan, maka motor kanan akan dipercepat dan motor kiri akan diperlambat, sehingga robot akan menjauhi dinding kanan. Sedangkan apabila robot terlalu jauh dengan dinding kanan, maka motor sebelah kiri akan dipercepat, dan motor sebelah kanan akan diperlambat, sehingga robot akan berjalan mendekati dinding kanan [12].