

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. *Remote* Inframerah

*Remote* inframerah adalah teknologi komunikasi nirkabel yang menggunakan cahaya inframerah (IR) untuk mengirimkan sinyal pengendalian dari perangkat pengirim (seperti *remote control*) ke perangkat penerima (seperti perangkat elektronik). Cahaya inframerah memiliki panjang gelombang di luar rentang yang dapat dilihat oleh mata manusia. Oleh karena itu, cocok untuk digunakan pada komunikasi jarak dekat tanpa mengganggu aktivitas manusia [14].

Sumber cahaya inframerah digunakan dalam *remote* inframerah untuk mengirimkan sinyal pengendalian. Sinyal ini bisa berupa kombinasi pulsa dari cahaya inframerah yang akan diinterpretasikan oleh perangkat penerima untuk menjalankan perintah yang sesuai. Sinyal ini dapat mengandung informasi tentang tombol yang ditekan di *remote control*, seperti volume naik, volume turun, mengganti saluran televisi, dan lain-lain [15].

Prinsip dasar *remote* inframerah melibatkan penggunaan cahaya inframerah yang tidak terlihat oleh mata manusia sebagai media pengiriman sinyal antara *remote control* dan perangkat yang dikendalikan. Berikut adalah beberapa konsep teori dasar terkait *remote* inframerah:

- 1) Cahaya Inframerah: Inframerah adalah jenis cahaya yang memiliki panjang gelombang lebih panjang daripada cahaya tampak tetapi lebih pendek daripada gelombang radio. Cahaya inframerah digunakan dalam *remote* inframerah sebagai medium untuk mengirimkan sinyal. Karena cahaya inframerah tidak terlihat oleh mata manusia, sinyal ini tidak mengganggu aktivitas sehari-hari.

- 2) Pemancar dan Penerima Inframerah: *Remote control* dilengkapi dengan pemancar inframerah (IR LED), sedangkan perangkat yang dikendalikan memiliki penerima inframerah (IR sensor). Pemancar mengirimkan sinyal IR yang berisi informasi pengendalian, seperti tombol yang ditekan. Penerima menerima dan menguraikan sinyal tersebut.
- 3) Pulsa Inframerah: Sinyal pengendalian yang dikirimkan oleh *remote control* umumnya berupa pulsa pause dari cahaya inframerah. Pola ini dibaca oleh perangkat penerima. Setiap tombol pada *remote control* biasanya memiliki pola pulsa-pause yang unik, yang diinterpretasikan oleh perangkat penerima sebagai perintah tertentu.
- 4) Kode Inframerah: Setiap tombol pada *remote control* memiliki kode inframerah khusus yang dikirimkan saat tombol ditekan. Kode ini digunakan untuk mengidentifikasi tombol mana yang ditekan oleh pengguna. Ketika kode ini diterima oleh perangkat penerima, perangkat tersebut mengartikulasikan perintah yang sesuai.
- 5) Komunikasi Jarak Dekat: *Remote* inframerah lebih cocok digunakan untuk komunikasi jarak dekat. Sinyal cahaya inframerah dapat terhambat oleh halangan fisik, dan oleh karena itu, *remote* inframerah bekerja paling baik dalam garis pandang langsung antara pemancar dan penerima.

## 2.2 *Remote* Terpusat

*Remote* terpusat atau *remote* universal merupakan perangkat yang digunakan untuk mengendalikan beberapa alat elektronik. Televisi dan *Digital Versatile Disk* (DVD) *player* dapat dikendalikan dengan satu perangkat *remote*. Perangkat ini dibuat untuk menyederhanakan pengendalian beberapa alat elektronik secara jarak

jauh. *Remote* terpusat biasanya menggunakan sinyal inframerah untuk berkomunikasi dengan perangkat elektronik yang dikendalikan [16].

Konsep utama di balik *remote* terpusat adalah menyederhanakan dan mengintegrasikan proses pengendalian perangkat elektronik yang beragam. Dengan menggunakan satu perangkat pengendali, pengguna tidak perlu memiliki banyak *remote* yang berbeda untuk setiap perangkat. Sebagai contoh, pengguna dapat mengontrol televisi dan pemutar DVD hanya dengan menggunakan satu *remote* terpusat [16].

Salah satu teknologi yang umum digunakan dalam *remote* terpusat adalah sinyal inframerah. Sinyal inframerah digunakan sebagai medium komunikasi antara perangkat pengendali dan perangkat elektronik yang ingin dikendalikan. Perintah yang diberikan oleh pengguna melalui tombol-tombol pada *remote* terpusat dikirimkan dalam bentuk sinyal inframerah, yang kemudian diterjemahkan oleh perangkat elektronik menjadi perintah aksi yang sesuai [16].

Dengan menggabungkan kemampuan mengendalikan berbagai perangkat elektronik menjadi satu perangkat, *remote* terpusat memberikan kenyamanan dan efisiensi dalam mengelola perangkat-perangkat tersebut. Teknologi ini memainkan peran penting dalam meningkatkan pengalaman pengguna dalam mengoperasikan perangkat elektronik di dalam rumah atau lingkungan lainnya [16].

### **2.3. Suara**

Suara merupakan bunyi yang dapat didengar dengan perambatan getaran melalui media transmisi gas, cair, atau padat. Suara dapat dihasilkan oleh makhluk hidup terutama manusia. Suara dapat diukur kecepatan osilasinya atau frekuensinya dalam satuan Hertz (Hz). Contohnya adalah suara manusia yang memiliki rentang

frekuensi pada nilai 20 Hz – 4000 Hz [17]. Suara manusia dapat digunakan untuk berkomunikasi melalui pengiriman sinyal suara. Sinyal suara ini terdiri atas tiga bagian yaitu bagian *voiced*, *unvoiced*, dan *silence*.

Bagian *voiced* pada sinyal suara dihasilkan apabila suara terdapat huruf-huruf vokal misalnya a, i, u, e, dan o [18]. Bagian *voiced* pada sinyal suara ini juga akan membentuk sinyal yang periodik [19]. Kemudian, pada bagian *unvoiced* dihasilkan jika suara tidak berisi huruf-huruf vokal [18]. Hal ini menyebabkan sinyal yang dibentuk pada bagian *unvoiced* tersebut tidak periodik bahkan seperti sinyal derau [19]. Bagian *silence* pada sinyal suara menandakan bahwa suara tidak memiliki bagian *voiced* dan bagian *unvoiced* [18]. Ketiga bagian ini dapat dipakai untuk proses analisis sinyal suara ataupun untuk teknologi lainnya.

#### **2.4. *Speech Recognition***

*Speech recognition* adalah teknologi yang digunakan dalam proses mengubah kata atau kalimat yang diucapkan menjadi teks atau perintah yang dipahami oleh komputer [20]. Teknologi ini menggunakan algoritma dan model statistik yang rumit untuk membandingkan pola suara yang didengar dengan basis data sebagai model pelatihan.

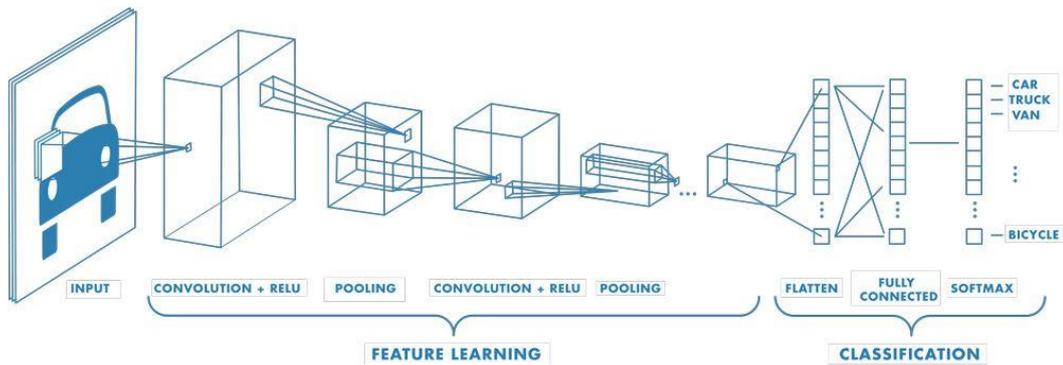
Teknologi *speech recognition* melibatkan penggunaan algoritma dan model statistik yang kompleks untuk membandingkan pola suara yang didengar dengan model pelatihan yang telah diberikan kepada komputer. Model pelatihan ini mencakup berbagai variasi suara manusia dan frasa yang sering digunakan dalam bahasa yang diinginkan. Dengan melakukan perbandingan ini, komputer dapat mencari kesamaan pola suara dan memutuskan kata atau kalimat mana yang paling cocok dengan apa yang didengar [20].

Salah satu metode yang digunakan dalam *speech recognition* adalah *Convolutional Neural Network* (CNN). CNN adalah jenis algoritma *deep learning* yang terinspirasi oleh struktur visual korteks otak manusia. Algoritma ini memiliki kemampuan untuk memahami fitur-fitur kompleks dari data, termasuk data suara. Dengan melakukan serangkaian transformasi dan analisis pada spektrum suara, CNN dapat mengidentifikasi pola suara yang berkaitan dengan kata atau kalimat tertentu [20].

Penerapan *speech recognition* memiliki berbagai macam aplikasi, mulai dari pengenalan perintah suara dalam teknologi smart home hingga pengembangan asisten virtual berbicara. Teknologi ini terus berkembang untuk meningkatkan akurasi dan kemampuan dalam mengenali beragam variasi suara manusia serta untuk mengatasi tantangan seperti adanya *noise* dan aksen berbeda [20].

## **2.5. Convolutional Neural Network (CNN)**

*Convolutional Neural Network* (CNN) merupakan salah satu algoritma atau metoda yang digunakan dalam teknologi *speech recognition*. CNN biasa digunakan untuk pemrosesan gambar dan video [21]. Metoda ini dirancang untuk memproses data masukan dua dimensi seperti gambar dan memanfaatkan hubungan spasial antar piksel [22]. CNN terdiri atas beberapa lapisan termasuk *convolutional layer*, *pooling layer*, dan *fully connected layer* [21]. Pada **Gambar 2.1** menunjukkan tahapan pada metoda CNN dalam memproses data masukan dua dimensi.



**Gambar 2.1.** Tahapan pada Metoda CNN [23]

Pada **Gambar 2.1** dapat dilihat tahapan dari metoda CNN. Tahapan ini terdiri atas dua bagian utama, yaitu *feature learning* dan *classification*. Adapun penjelasan kedua bagian utama ini adalah sebagai berikut.

### 2.5.1. *Feature Learning*

*Feature learning* merupakan salah satu bagian utama yang berfungsi dalam proses *encoding* gambar menjadi *feature*. *Feature* ini berupa nilai-nilai yang mewakili gambar tersebut. Proses pada *feature learning* menggunakan beberapa *layer* agar diperoleh suatu ciri dari gambar. Adapun pemaparan beberapa *layer* yang ada pada *feature learning* yaitu:

#### 1. *Convolution Layer*

*Convolution layer* adalah *layer* fundamental dari arsitektur CNN. *Layer* ini dirancang untuk mendeteksi *feature* dalam gambar masukan dengan menggabungkan serangkaian filter yang dapat dipelajari dengan gambar masukan. Keluaran dari *convolution layer* adalah sekumpulan *feature maps* yang merepresentasikan keberadaan *feature* pada gambar masukan [24].

#### 2. *Rectified Linear Unit (ReLU) Activation*

*Rectified Linear Unit (ReLU) Activation* adalah fungsi aktivasi yang digunakan dalam CNN. Fungsi ini merupakan fungsi sederhana yang

mengembalikan masukan jika bernilai positif dan nol jika bernilai negatif. Fungsi ini banyak digunakan karena komputasinya yang efisien dan telah terbukti bekerja dengan baik secara praktis [25].

### 3. *Pooling Layer*

*Pooling layer* pada CNN adalah *layer* yang digunakan dalam pemrosesan data yang mengurangi dimensi *feature maps* tanpa kehilangan pola utama. Ada berbagai jenis *pooling layer*, seperti *max pooling*, *average pooling*, dan *stochastic pooling*. *Layer* ini digunakan untuk mengurangi jumlah parameter dan untuk mengendalikan *overfitting* [26].

#### 2.5.2. *Classification*

*Classification* merupakan salah satu bagian utama yang berfungsi untuk melakukan klasifikasi. Proses klasifikasi dilakukan pada setiap *neuron* yang telah dilakukan ekstraksi pada *feature learning*. Proses pada *classification* menggunakan beberapa *layer* yang saling berhubungan satu dengan yang lain. Adapun pemaparan beberapa *layer* yang ada pada *classification* yaitu:

##### 1. *Flatten Layer*

*Flatten layer* adalah *layer* pada arsitektur CNN yang digunakan untuk mengubah keluaran dari bagian utama *feature learning*. Keluaran diubah menjadi satu dimensi *array* yang dapat dimasukkan ke dalam *fully connected layer* [27].

##### 2. *Dropout Layer*

*Dropout layer* adalah jenis teknik *regularization* yang digunakan dalam CNN untuk mencegah *overfitting* dan meningkatkan generalisasi model. *Layer* ini secara acak melakukan proses *drop out* pada persentase tertentu dari *neuron* di

*layer* selama iterasi pada pelatihan. *Layer* ini juga biasanya diterapkan pada CNN *dense layer* [28].

### 3. *Fully Connected Layer*

*Fully connected layer* pada CNN adalah *layer* yang semua *neuron* terhubung ke semua *neuron* di *layer* sebelumnya. *Layer* ini biasanya digunakan sebagai *layer* terakhir dalam arsitektur CNN untuk proses klasifikasi kategori. Parameter pada *layer* ini dapat mencapai ratusan juta *dataset* pelatihan yang berisi jutaan identitas dengan membutuhkan banyak sumber daya komputasi [29].

### 4. *Softmax Layer*

*Softmax layer* adalah jenis fungsi aktivasi yang digunakan dalam CNN. *Layer* ini biasanya digunakan sebagai *layer terakhir* pada CNN. Tujuan dari *layer* ini adalah untuk mengubah keluaran dari *layer* sebelumnya menjadi distribusi probabilitas pada kelas yang berbeda. Fungsi aktivasi *softmax* mengambil vektor masukan bilangan real dan mengeluarkan vektor dengan dimensi yang sama. Setiap elemen pada vektor yang dikeluarkan adalah probabilitas 0 dan 1 yang mewakili masukan milik suatu kelas tertentu [30].

## 2.6. *Background Noise*

*Background noise* merujuk pada suara yang tidak diinginkan dalam suatu lingkungan tertentu. *Background noise* dapat disebabkan oleh faktor-faktor seperti suara yang dihasilkan pada saat manusia beraktivitas. *Background noise* dapat menjadi masalah pada berbagai bidang termasuk *speech recognition*. Pada *speech recognition*, *background noise* bertindak sebagai gangguan yang dapat menurunkan kualitas sinyal informasi [31]. Dalam *speech recognition*, *background noise* menjadi masalah yang signifikan. Ketika seseorang berbicara atau menggunakan

*speech recognition*, suara latar belakang yang tidak diinginkan dapat mencampuri atau mencampurkan dengan suara utama, mengganggu proses pengenalan suara yang akurat. *Background noise* dapat menyebabkan informasi yang diucapkan menjadi kabur atau sulit diidentifikasi oleh algoritma pengenalan suara [31].

Penanganan *background noise* dalam *speech recognition* melibatkan teknik-teknik pengurangan *noise*. Salah satu pendekatan umum adalah penggunaan *band pass filter* dan *transformasi wavelet*. *Band pass filter* digunakan untuk menghilangkan frekuensi suara yang tidak diinginkan, sehingga hanya frekuensi yang relevan dengan suara manusia yang diteruskan ke algoritma pengenalan suara. *Transformasi wavelet*, memungkinkan pemisahan antara komponen suara latar belakang dan suara utama dalam domain frekuensi dan waktu. Dengan menggunakan teknik-teknik ini, tingkat akurasi pengenalan suara dapat ditingkatkan dengan mengurangi dampak dari *background noise* [31].

Pentingnya penanganan *background noise* dalam *speech recognition* menunjukkan betapa pentingnya memahami karakteristik lingkungan dan sumber suara yang mungkin mempengaruhi kualitas dan akurasi proses pengenalan suara. Solusi untuk mengurangi dampak *background noise* ini terus dikembangkan untuk memastikan bahwa teknologi *speech recognition* dapat memberikan hasil yang lebih baik dalam berbagai kondisi lingkungan [31].

## **2.7. Mikrofon Kondensor**

Mikrofon kondensor merupakan jenis mikrofon yang menggunakan kapasitor untuk mengubah gelombang suara menjadi sinyal listrik. Kapasitor pada mikrofon kondensor terdiri atas dua pelat. Salah satu pelat adalah diafragma tipis yang bergetar sebagai respons terhadap gelombang suara. Ketika diafragma bergetar,

jarak antara kedua pelat berubah sehingga menciptakan muatan listrik yang sebanding dengan gelombang suara. Kemudian, muatan ini diperkuat dan direkam sebagai sinyal audio [32].

Mikrofon kondensor biasanya digunakan di studio rekaman dan pertunjukan secara langsung karena sangat sensitif dan dapat menangkap suara dalam rentang frekuensi yang luas. Selain itu, mikrofon kondensor dapat digunakan dalam penelitian ilmiah [33]. Penelitian ini menggunakan mikrofon kondensor yang visualisasinya dapat dilihat pada **Gambar 2.2**.



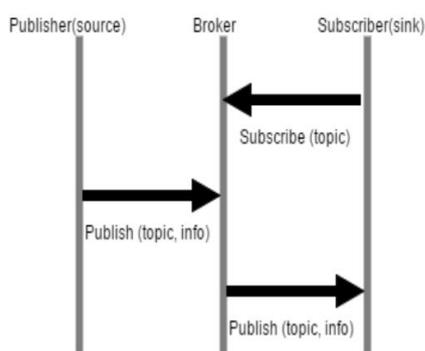
**Gambar 2.2.** Mikrofon Kondensor

Pada **Gambar 2.2** menunjukkan mikrofon kondensor yang digunakan pada penelitian ini. Mikrofon kondensor tersebut memiliki spesifikasi. Spesifikasi pada mikrofon kondensor tersebut yaitu:

1. Respons Frekuensi : 40 – 16.000 Hz
2. Sensitivitas : -30 dB +/- 3 dB
3. Impedansi : 16 K $\Omega$
4. *Signal to Noise Ratio* : 60 dB
5. Daya : 2 V +/- 5%
6. Ukuran *Jack* : 3,5 mm

## 2.8. Protokol *Message Queuing Telemetry Transport* (MQTT)

Protokol *Message Queuing Telemetry Transport* (MQTT) merupakan protokol pengiriman pesan ringan dalam konteks *Internet of Things* (IoT) untuk mengirimkan suatu data [34]. Protokol MQTT biasanya digunakan sebagai protokol koneksi M2M. Protokol ini bekerja pada TCP/IP *port*. Protokol MQTT menggunakan pola *publish/subscribe* untuk melakukan proses pengiriman data melalui jaringan internet [35]. Adapun pola *publish/subscribe* pada protokol MQTT dapat dilihat pada **Gambar 2.3**.



**Gambar 2.3.** Pola *Publish/Subscriber* pada Protokol MQTT [34]

Arsitektur pada protokol MQTT terdiri atas tiga komponen yaitu *publisher*, *broker*, dan *subscriber*. *Publisher* bertugas untuk mengirimkan informasi kepada *subscriber* melalui *broker*. *Broker* menjadi pusat pada protokol MQTT untuk mendistribusikan informasi untuk setiap MQTT *clients* (*publisher* dan *subscriber*). *Subscriber* akan menerima informasi tersebut setelah informasi dari *publisher* sudah dimuat pada *broker* [35].

Pada protokol MQTT terdapat tiga level *Quality of Service* (QoS). Qos ini bertujuan untuk mempertahankan kehandalan dari informasi yang didistribusikan. Tiga level QoS yaitu level 0, level 1, dan level 2. Level 0 disebut dengan *one*

*delivery (at most)*. Informasi pada level 0 akan dikirimkan sekali tanpa adanya konfirmasi penerimaan. Kemudian, level 1 dipanggil dengan nama *one delivery (at least)*. Pengiriman informasi pada level 1 dilakukan minimal sekali dan adanya konfirmasi penerimaan informasi. Level yang terakhir adalah level 2 yang disebut dengan *one delivery (exactly)*. Level 2 memberikan garansi bahwa informasi pasti dikirimkan hanya sekali. Level ini merupakan level tertinggi pada level QoS. Adapun penjelasan level QoS dapat dilihat pada **Tabel 2.1**.

**Tabel 2.1.** Penjelasan Level QoS pada Protokol MQTT [34]

Level QoS	Penjelasan
Level 0	Sebuah informasi disampaikan paling banyak sekali dan tidak ada konfirmasi penerimaan yang diperlukan.
Level 1	Setiap informasi disampaikan setidaknya sekali dan konfirmasi menerima informasi diperlukan.
Level 2	Mekanisme <i>a four-way handshake</i> digunakan tepat satu kali untuk penyampaian sebuah informasi.

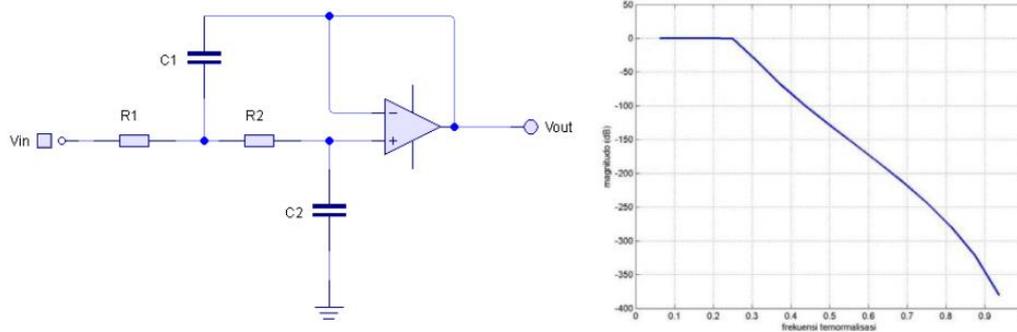
## 2.9. Filter

Filter merupakan jaringan listrik yang dirancang untuk mengizinkan frekuensi tertentu melewatinya dengan melemahkan yang lain. Filter digunakan untuk mengurangi *noise* yang tidak diinginkan [36]. Berdasarkan sifat, filter terdiri atas dua bagian yaitu filter pasif dan filter aktif. Kedua filter tersebut memiliki perbedaan dalam komponen yang digunakan.

Pada filter pasif terdiri atas komponen-komponen pasif seperti kapasitor, resistor, atau induktor. Akan tetapi, filter aktif terdiri atas komponen-komponen aktif seperti transistor atau op-amp. Filter pasif dan filter aktif tetap memiliki fungsi yang sama yaitu mengizinkan frekuensi tertentu melewatinya dengan melemahkan yang lain. Kemudian, filter juga dapat dibagi menjadi empat bagian berdasarkan jenis dari filter. Adapun empat bagian filter tersebut yaitu:

### 2.9.1. Low Pass Filter (LPF)

*Low Pass Filter* (LPF) merupakan filter yang mengizinkan sinyal frekuensi rendah melewatinya dan melemahkan sinyal frekuensi tinggi. LPF dapat dirancang dengan menggunakan filter *Butterworth* [37]. Adapun rangkaian dan respon frekuensi LPF *Butterworth* dengan menggunakan desain *Sallen-key* dapat dilihat pada **Gambar 2.4**.



**Gambar 2.4.** Rangkaian dan Respon Frekuensi LPF *Butterworth Sallen-key* [37]

Pada **Gambar 2.4** menunjukkan rangkaian dan respon frekuensi LPF *Butterworth Sallen-key* dengan orde 2. Apabila pada rangkaian tersebut parameter  $R_1$  dan  $R_2$  sama dengan  $R$ , maka dapat dirumuskan yaitu:

$$2\pi \cdot f_c = \frac{1}{R \cdot \sqrt{C_1 \cdot C_2}} \quad (2.1)$$

$$Q = 0,5 \cdot \sqrt{\frac{C_1}{C_2}} \quad (2.2)$$

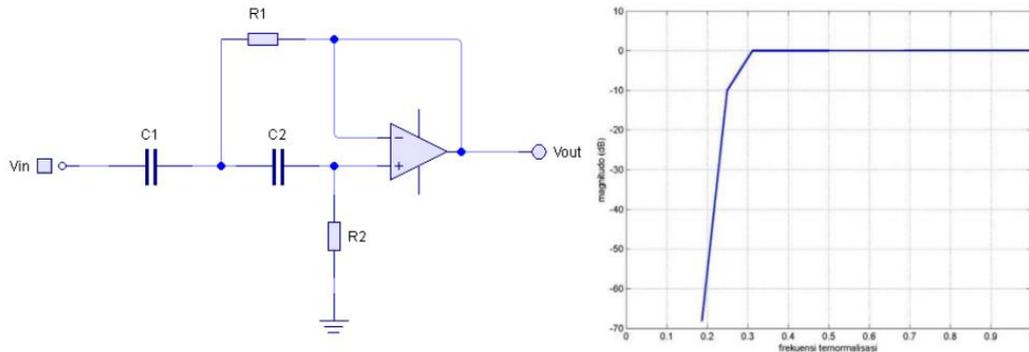
dengan:  $f_c$  : frekuensi *cut-off* (Hz)

$Q$  : faktor kualitas [36].

### 2.9.2. High Pass Filter (HPF)

*High Pass Filter* (HPF) merupakan filter yang mengizinkan sinyal frekuensi tinggi melewatinya dan melemahkan sinyal frekuensi rendah. HPF dapat dirancang

dengan menggunakan filter *Butterworth* [37]. Adapun rangkaian dan respon frekuensi HPF *Butterworth* dengan menggunakan desain *Sallen-key* dapat dilihat pada **Gambar 2.5**.



**Gambar 2.5.** Rangkaian dan Respon Frekuensi HPF *Butterworth Sallen-key* [37]

Pada **Gambar 2.5** menunjukkan rangkaian dan respon frekuensi HPF *Butterworth Sallen-key* dengan orde 2. Apabila pada rangkaian tersebut parameter  $C_1$  dan  $C_2$  sama dengan  $C$ , maka dapat dirumuskan yaitu:

$$2\pi \cdot f_c = \frac{1}{C^2 \cdot R_1 \cdot R_2} \quad (2.3)$$

$$Q = 0,5 \cdot \sqrt{\frac{R_2}{R_1}} \quad (2.4)$$

$$R_1 = \frac{1}{2 \cdot Q \cdot 2\pi \cdot f_c \cdot C} \quad (2.5)$$

$$R_2 = \frac{2 \cdot Q}{2\pi \cdot f_c \cdot C} \quad (2.6)$$

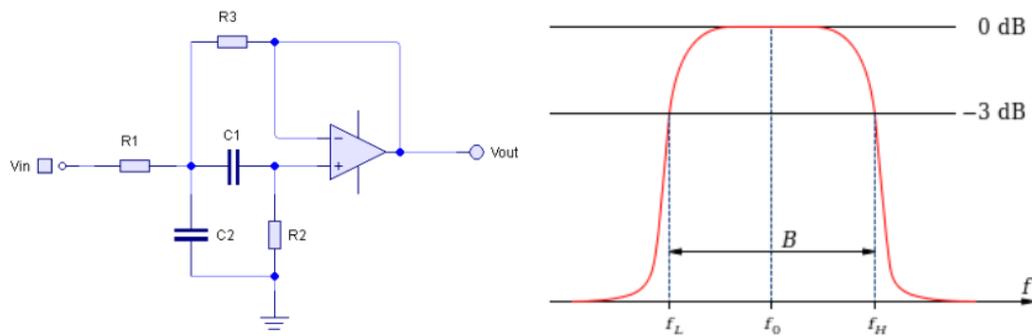
dengan:  $f_c$  : frekuensi *cut-off* (Hz)

$Q$  : faktor kualitas [37].

### 2.9.3. Band Pass Filter (BPF)

*Band Pass Filter* (BPF) merupakan filter yang terdiri atas LPF dan HPF. Gabungan antara LPF dan HPF ini dapat meloloskan frekuensi dengan rentang

tertentu dan rentang lainnya akan dilemahkan. Rentang frekuensi pada BPF disebut dengan *bandwidth*. BPF dapat dirancang dengan menggunakan filter *Butterworth* [37]. Adapun rangkaian dan respon frekuensi BPF *Butterworth* dengan menggunakan desain *Sallen-key* dapat dilihat pada **Gambar 2.6**.



**Gambar 2.6.** Rangkaian dan Respon Frekuensi BPF *Butterworth Sallen-key* [37]

Pada **Gambar 2.6** menunjukkan rangkaian dan respon frekuensi HPF *Butterworth Sallen-key* dengan orde 2. Apabila pada rangkaian tersebut parameter  $C_1$  dan  $C_2$  sama dengan  $C$ , parameter  $R_1$  dan  $R_3$  sama dengan  $R$ , dan parameter  $R_2$  sama dengan  $2R$ , maka dapat dirumuskan yaitu:

$$2\pi \cdot f_0 = \frac{1}{C \cdot R} \quad (2.7)$$

$$Q = \frac{f_0}{BW} \quad (2.8)$$

$$BW = f_H - f_L \quad (2.9)$$

$$f_0 = \sqrt{BW} \quad (2.10)$$

dengan:  $f_0$  : frekuensi tengah (Hz)

$Q$  : faktor kualitas

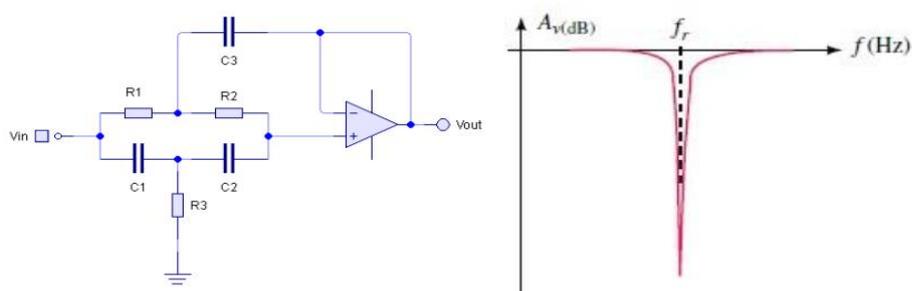
$BW$  : *bandwidth* atau lebar frekuensi (Hz)

$f_H$  : frekuensi tinggi (Hz)

$f_L$  : frekuensi rendah (Hz) [37].

### 2.9.4. Band Reject Filter/Band Stop Filter/Notch Filter

*Notch Filter* merupakan filter yang dapat meloloskan frekuensi dengan rentang tertentu dan rentang lainnya akan dilemahkan. *Notch Filter* biasanya digunakan untuk menghilangkan sebuah frekuensi atau rentang frekuensi yang sempit. *Notch Filter* dapat dirancang dengan menggunakan filter *Butterworth* [37]. Adapun rangkaian dan respon frekuensi *Notch Filter Butterworth* dengan menggunakan desain *Sallen-key* dapat dilihat pada **Gambar 2.7**.



**Gambar 2.7.** Rangkaian dan Respon Frekuensi *Notch Filter Butterworth Sallen-key* [37]

Pada **Gambar 2.7** menunjukkan rangkaian dan respon frekuensi *Notch Filter Butterworth Sallen-key* dengan orde 2. Apabila pada rangkaian tersebut parameter  $C_1$  dan  $C_2$  sama dengan  $C$ , parameter  $C_3$  sama dengan  $2C$ , parameter  $R_1$  dan  $R_2$  sama dengan  $R$ , dan parameter  $R_3$  sama dengan  $R/2$ , maka dapat dirumuskan yaitu:

$$2\pi \cdot f_r = \frac{1}{C \cdot R} \quad (2.11)$$

dengan:  $f_r$  : frekuensi tengah (Hz) [37].

## 2.10. Filter Digital

Filter digital merupakan filter yang memproses sinyal digital untuk menghapus atau mengubah frekuensi sinyal tertentu. Filter ini digunakan untuk menghilangkan frekuensi yang tidak diinginkan dan mendapatkan sinyal dengan

frekuensi yang sesuai. Ada beberapa jenis filter digital, misalnya filter digital filter digital *Infinite Impulse Response* (IIR) dan *Finite Impulse Response* (FIR)[38].

Filter digital IIR merupakan filter digital yang dirancang untuk menghasilkan respons impuls tidak terbatas pada suatu sistem dinamis [42]. Filter ini digunakan dalam aplikasi dengan fase linier yang tidak terlalu penting dan memori yang terbatas [41]. Keluaran dari filter digital ini bukan hanya bergantung pada masukan, tetapi juga bergantung pada keluaran sebelumnya [43]. Jumlah memori yang besar dan kinerja komputasi yang baik meningkatkan efisiensi waktu filter tersebut [42].

Filter digital FIR merupakan filter digital dengan respons impuls terbatas dan transformasi-z bernilai *nulls* [39]. Keluaran dari filter digital ini hanya bergantung pada masukan dan bukan pada keluaran sebelumnya [40]. Jumlah memori yang besar dan kinerja komputasi yang baik dibutuhkan pada proses filter digital FIR [41]. Filter digital FIR stabil dan mudah diimplementasikan pada sinyal dengan respons fase yang linier [39].

Akan tetapi, filter digital FIR dapat mengalami efek *lobes side*. Efek tersebut dapat menyebabkan beberapa daerah respon frekuensi dari filter digital FIR terdapat peningkatan dan penurunan amplitudo. Hal ini menandakan bahwa respons frekuensi filter memiliki variasi yang tidak merata [42]. Oleh karena itu, perancangan filter digital FIR biasanya menggunakan fungsi *window* yang diterapkan pada koefisien filter yang dihasilkan.

Fungsi *window* adalah fungsi matematis yang mengubah respons frekuensi filter digital FIR dengan mengurangi efek *lobes side* [42]. Salah satu fungsi *windowing* yang digunakan adalah *kaiser window*. *Kaiser window* biasanya digunakan dalam proses desain filter digital FIR untuk memperoleh karakteristik

respons frekuensi yang diinginkan. Fungsi *window* ini menghasilkan bentuk *window* yang dapat diatur dengan parameter beta ( $\beta$ ). Parameter ini dapat mengontrol tingkat penurunan *lobes side* [43]. Adapun rumus *kaiser window* adalah sebagai berikut:

$$w[n] = \frac{I_0 \left( \beta \sqrt{1 - \left( \frac{2n}{N-1} - 1 \right)^2} \right)}{I_0(\beta)} \quad (2.12)$$

$$N = \frac{9A - 8}{22.85 \cdot 2\pi b} + 1 \quad (2.13)$$

$$\beta = \begin{cases} 0.1102(A - 8.7), & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21), & 21 \leq A \leq 50 \\ 0, & A < 21 \end{cases} \quad (2.14)$$

dengan:  $w[n]$  : nilai *kaiser window*  
 $N$  : panjang *window*  
 $n$  : indeks dalam rentang 0 sampai dengan  $N-1$   
 $\beta$  : parameter pengontrol tingkat penurunan *lobes side*  
 $I_0$  : fungsi *bessel* orde nol  
 $A$  : level atenuasi sinyal yang diizinkan (dB)  
 $b$  : lebar pita transisi (Hz) [43]

## 2.11. Transformasi *Wavelet*

Transformasi *wavelet* merupakan teknik dalam dalam analisis sinyal dan data dalam domain waktu dan frekuensi. Transformasi ini berguna untuk menganalisis sinyal non-stasioner yang frekuensi sinyal berubah seiring waktu [44]. Dalam transformasi *wavelet*, analisis sinyal dilakukan dengan dekomposisi dan rekonstruksi menggunakan fungsi-fungsi *wavelet*. Transformasi *wavelet* dapat menguraikan sinyal. Penguraian sinyal dilakukan dengan adanya LPF dan HPF. LPF digunakan untuk deteksi aproksimasi dan HPF digunakan untuk deteksi detail. Dengan adanya deteksi aproksimasi dan deteksi detail ini, transformasi *wavelet*

dapat memperoleh informasi pola, perubahan, atau fitur yang ada dalam sinyal tersebut.

Transformasi *wavelet* dapat digunakan dalam berbagai aplikasi termasuk *noise reduction* dalam sinyal suara. Dalam bidang *noise reduction*, transformasi *wavelet* mengolah dan meningkatkan sinyal suara yang terganggu dari *noise*. Transformasi *wavelet* dapat membantu dalam mengurangi *noise* tersebut [45]. Proses pengurangan *noise* tidak hanya dilakukan transformasi *wavelet*, tetapi juga dibantu *thresholding wavelet*.

Salah satu metode *thresholding wavelet* yang dapat digunakan adalah *Stein's Unbiased Risk Estimate Shrink* (SUREShrink). Metode ini bertujuan untuk memilih *threshold* yang sesuai untuk *noise reduction*. Dalam praktiknya, metode SUREShrink bersama dengan transformasi *wavelet* memisahkan sinyal menjadi koefisien *wavelet*. Selanjutnya adalah perhitungan batas *threshold*. Koefisien *wavelet* dengan nilai di bawah *threshold* dianggap sebagai *noise* dan koefisien *wavelet* dengan nilai di atas *threshold* dianggap sinyal informasi [46]. Adapun rumus metode SUREShrink adalah sebagai berikut.

$$\text{SURE}(T) = N \sigma^2 - 2N \sigma^2 \left(1 - \frac{1}{N} \|S\|^2\right) + \|D\|^2 \quad (2.15)$$

$$\sigma = \frac{\text{median}(|c|)}{0,6745} \quad (2.16)$$

dengan:

- SURE(T) : estimasi risiko yang tidak bias untuk *threshold*  $T$
- $\sigma$  : standar deviasi *noise*
- $N$  : jumlah sampel sinyal yang terdapat *noise*
- $\|S\|^2$  : norm-kuadrat sinyal *denoised*
- $\|D\|^2$  : norm-kuadrat perbedaan sinyal *denoised* dan asli
- $c$  : koefisien detail terakhir

## 2.12. Protokol *Zigbee2MQTT*

Protokol *Zigbee2MQTT* adalah suatu protokol yang memungkinkan perangkat *Zigbee*, seperti perangkat pintar dalam lingkungan rumah cerdas, untuk berkomunikasi dengan broker *Message Queuing Telemetry Transport* (MQTT). Fungsi dari protokol ini adalah untuk menciptakan integrasi yang lebih efisien dan terpusat antara berbagai perangkat elektronik yang ada di dalam rumah. Selain itu, protokol ini juga memungkinkan pengguna untuk mengontrol dan memantau perangkat tersebut melalui suara dan melalui jaringan IoT [47].

Berikut adalah beberapa elemen inti dalam *Zigbee2MQTT*:

- 1) Protokol *Zigbee*: *Zigbee* adalah suatu protokol jaringan nirkabel yang diciptakan untuk memfasilitasi komunikasi perangkat kecil dan hemat energi dalam jaringan sensor nirkabel. Protokol ini menggunakan struktur jaringan berbasis mesh untuk menghubungkan perangkat secara spontan dan dapat diterapkan dalam berbagai aplikasi di rumah cerdas.
- 2) Protokol MQTT: MQTT merupakan protokol pesan yang ringan dan digunakan untuk bertukar pesan antara perangkat melalui jaringan. Protokol ini sangat sesuai untuk aplikasi IoT karena mengoptimalkan penggunaan daya baterai dan kapasitas bandwidth. MQTT beroperasi dalam pola publikasi-langganan, dimana perangkat (pengirim pesan) dapat mengirimkan pesan ke topik tertentu, dan perangkat lain (penerima pesan) dapat berlangganan topik tersebut untuk menerima pesan.
- 3) Broker MQTT: Broker MQTT adalah perantara yang menerima pesan dari perangkat yang mengirimkan pesan dan mengirimkannya kepada perangkat

yang berlangganan topik yang bersangkutan, dengan memfasilitasi komunikasi terpusat antara berbagai perangkat.

- 4) Jembatan *Zigbee2MQTT*: Protokol *Zigbee2MQTT* bertindak sebagai penghubung antara perangkat *Zigbee* dan broker MQTT. Protokol ini mengambil pesan dari perangkat *Zigbee*, mengkonversinya ke format yang sesuai dengan MQTT, dan mengirimkannya kepada broker MQTT. Sebaliknya, saat pesan diterima dari broker MQTT, jembatan *Zigbee2MQTT* akan mengartikulasikan pesan tersebut ke dalam format yang dapat dipahami oleh perangkat *Zigbee*.
- 5) Integrasi dengan Rumah Pintar: Protokol *Zigbee2MQTT* memungkinkan perangkat *Zigbee* untuk diintegrasikan dengan sistem rumah pintar yang lebih luas. Perangkat *Zigbee* dapat diatur dan diawasi melalui aplikasi atau kontrol lainnya yang terhubung ke broker MQTT.

Dalam penerapannya, *Zigbee2MQTT* memerlukan konfigurasi awal untuk menghubungkan perangkat *Zigbee* dengan broker MQTT, serta untuk mengatur topik dan pesan yang digunakan dalam proses komunikasi.