

**PENGENALAN GESTUR STATIK TANGAN PADA  
INTERAKSI DISABILITAS MENGGUNAKAN  
METODE DEEP LEARNING BERBASIS RADAR  
FMCW**

***(RECOGNITION OF STATIC HAND GESTURES IN  
DISABILITY INTERACTIONS USING DEEP LEARNING  
METHOD BASED ON FMCW RADAR)***

**TUGAS AKHIR**

Disusun sebagai syarat untuk memperoleh gelar Sarjana Teknik pada Program  
Studi S1 Teknik Elektro Universitas Telkom

Disusun oleh:

**FASYA ABHIRAMA KAUSAR**

**1102180002**



**Universitas  
Telkom**

**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2022**

# **LEMBAR PENGESAHAN**

## **TUGAS AKHIR**

**Pengenalan Gestur Statik Tangan pada  
Interaksi Disabilitas Menggunakan Metode  
Deep Learning Berbasis Radar FMCW  
(*RECOGNITION OF STATIC HAND GESTURES IN  
DISABILITY INTERACTIONS USING DEEP LEARNING  
METHOD BASED ON FMCW RADAR*)**

**Telah disetujui dan disahkan sebagai Buku Tugas Akhir**

**Program Studi Teknik Elektro**

**Fakultas Teknik Elektro**

**Universitas Telkom**

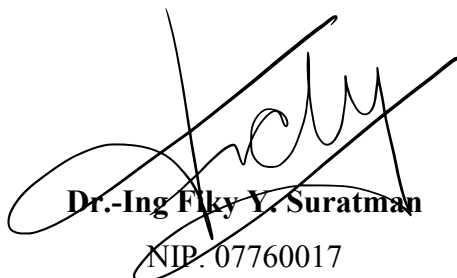
**Disusun oleh:**

**FASYA ABHIRAMA KAUSAR**

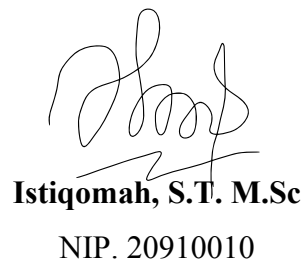
**1102180002**

**Bandung, 17 September 2022**

**Pembimbing I**

  
**Dr.-Ing Fiky Y. Suratman**  
NIP. 07760017

**Pembimbing II**

  
**Istiqomah, S.T. M.Sc**  
NIP. 20910010

## LEMBAR PERNYATAAN ORISINALITAS

Nama : Fasya Abhirama Kausar  
NIM : 1102180002  
Alamat : Jl. Pindahan No. 19, Kec. Pancoran Mas, Kel. Pancoran Mas,  
Depok, Jawa Barat  
No Tlp/HP : 082219728488  
E-mail : fasyakausar@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri,  
dengan judul :

**Pengenalan Gestur Statik Tangan Pada Interaksi Disabilitas Menggunakan  
Metode Deep Learning Berbasis Radar FMCW**

***Recognition Of Static Hand Gestures In Disability Interactions Using Deep  
Learning Method Based On FMCW Radar***

Atas pernyataan ini, saya siap menanggung resiko / sanksi yang dijatuhkan kepada  
saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran  
akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang  
menunjukkan ketidak aslian karya ini.

**Bandung, 17 September 2022**



**Fasya Abhirama Kausar**

**1102180002**

## ABSTRAK

Kesulitan penyandang tunarungu dan tunawicara dalam berinteraksi bermasyarakat sering dialami. Terutama jika penyandang tunarungu dan tunawicara masih baru, dikarenakan kurangnya masyarakat Indonesia dalam memahami bahasa isyarat. Negara Indonesia sering menggunakan bahasa isyarat SIBI (Sistem Isyarat Bahasa Indonesia) sebagai bahasa isyarat yang sudah terstandarisasi oleh pemerintah. Dalam mitigasi mempermudah masyarakat Indonesia dalam memahami bahasa isyarat, penelitian dengan sistem sistem untuk mengklasifikasikan bahasa SIBI menjadi solusi dari permasalahan tersebut.

Penelitian ini merancang sistem klasifikasi bahasa isyarat SIBI dengan gerak statik tangan dengan berbasis radar FMCW (*Frequency Modulated Continuous Wave*) dengan metode *deep learning* dengan algoritma CNN (*Convolutional Neural Network*). Radar FMCW bekerja mentransmisikan sinyal lalu terdapat sinyal pantul yang diakuisisi dari pendeteksian gerakan statik tangan yang terdeteksi dan menghasilkan dataset. Dataset akan digunakan sebagai sampel *training* dan *test* pada algoritma CNN yang di mana memiliki tahap *feature learning* dan *classification*. Data CNN yang diolah akan digunakan dalam klasifikasi kata SIBI agar memudahkan interaksi dan komunikasi antara penyandang tunarungu dan tunawicara terhadap masyarakat.

Tujuan dari penelitian ini mendapatkan hasil tingkat akurasi klasifikasi lebih dari 90%. Hasil penelitian didapatkan dengan pengujian menggunakan parameter perbandingan jumlah *layer* konvolusi, pengaruh jumlah *epoch*, dan *pooling layer*. Parameter pengujian menunjukkan bahwa mendapatkan keseluruhan akurasi dengan menggunakan parameter tersebut masing-masing mendapatkan akurasi sebesar 99,91% dan 99,97%. Penelitian ini berhasil memenuhi tujuan yang diusulkan dan bahasa isyarat SIBI dengan gerak statik tangan dapat diklasifikasikan dengan baik.

**Kata Kunci:** Bahasa Isyarat SIBI, Radar FMCW, CNN.

## ABSTRACT

Difficulties for people who are deaf and speech impaired in interacting in society are often experienced. Especially if people who are deaf and speech impaired are still new due to the lack of Indonesian people understanding of sign language. The Indonesian state often uses the SIBI (*Sistem Isyarat Bahasa Indonesia*) sign language as a sign language that the government has standardized. To make it easier for the Indonesian people to understand sign language, research with a system for translating the SIBI language solves these problems.

This study designed a SIBI sign language classification system with static hand motion based on FMCW (Frequency Modulated Continuous Wave) radar with a deep learning method with a CNN (Convolutional Neural Network) algorithm. The FMCW radar works to transmit signals, and then there is a reflected signal that is acquired from detecting static hand movements that are detected and generates a dataset. The dataset will be used as a training and *test* sample on the CNN algorithm, which has feature learning and classification stages. The processed CNN data will be used for the classification of SIBI words in order to facilitate interaction and communication between deaf and speech impaired people in the community.

This study aims to obtain a classification accuracy rate of more than 90%. The results were obtained by *testing* the comparison parameters of the number of convolution *layers*, the effect of the number of epochs, and two pooling *layers*. The *test* parameters show that getting the overall accuracy by using these parameters each get an accuracy of 99.91% and 99.97%. This study met the proposed objectives, and SIBI sign language with static hand movements can be classified well.

**Keywords:** *SIBI Language, Radar FMCW, CNN.*

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kepada Allah SWT atas izinnya untuk menyelesaikan Tugas Akhir ini yang berjudul **“PENGENALAN GESTUR STATIK TANGAN PADA INTERAKSI DISABILITAS MENGGUNAKAN METODE DEEP LEARNING BERBASIS RADAR FMCW”**. Maksud dan tujuan dibuatnya buku Tugas Akhir ini adalah untuk memenuhi persyaratan mendapatkan gelar sarjana. terselesaikannya Tugas Akhir ini tidak lepas dari pertolongan Allah SWT, terbantu dengan doa, dan dukungan banyak pihak baik materil maupun immateriil, untuk mengatasi segala rintangan dan rintangan yang dihadapi. Semoga Allah SWT membalas semua kebaikan yang telah diberikan kepada penulis. Pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah berkontribusi banyak dalam terwujudnya Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi kita, khususnya para pembaca. Tidak ada yang lebih sempurna dari Tugas Akhir ini, sehingga kritik dan saran yang membangun sangat dibutuhkan untuk masa yang akan datang.

## **UCAPAN TERIMA KASIH**

Dengan menyelesaikan Tugas Akhir di program studi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom, penulis mendapatkan banyak bantuan. baik berupa materi, dukungan, dan informasi dari banyak pihak. Pada kesempatan ini juga, penulis ingin mengucapkan terima kasih kepada pihak yang membantu menyusun Buku Tugas Akhir ini terutama kepada:

1. Allah SWT yang selalu memberikan rahmat dan karunia-Nya sehingga penulis diberi kelancaran dalam pelaksanaan hingga penyusunan buku Tugas Akhir.
2. Papah dan Mamah, orang tua terhebat yang tidak pernah lelah untuk membimbing, memberikan nasehat, melindungi dan menyertakan penulis dalam setiap doanya. Terima kasih dari hati yang terdalam atas semua yang sudah diberikan hingga saat ini, semoga Papah dan Mamah selalu berada dalam lindungan Allah S.W.T dan dilimpahkan Rezekinya, aamiin.
3. Pembimbing I, Bapak Dr.-Ing Fiky Y Suratman atas segala arahan dan bimbingan yang telah diberikan. Terima kasih telah meluangkan waktu untuk membimbing dari awal hingga selesai, semoga ilmu yang diberikan dapat bermanfaat khususnya bagi penulis.
4. Pembimbing II, Ibu Istiqomah, S.T., M.Sc. atas segala arahan dan bimbingan yang telah diberikan. Terima kasih telah meluangkan waktu untuk membimbing dari awal hingga selesai, semoga ilmu yang diberikan dapat bermanfaat khususnya bagi penulis.
5. Dosen Wali, Bapak, Dr. Ahmad Rizal atas segala ilmu dukungan dan bimbingan dari awal perkuliahan hingga penyelesaian Tugas Akhir. Semoga bapak dan keluarga selalu diberi kesehatan.
6. Rayhan Rafli, Dhimas Radhityo Sardjono, Bagas Wibisono, Rayhan Erdyarahman, serta Wilma Christy selaku rekan Tugas Akhir dan kelompok riset radar. Semoga kalian sehat selalu dan dilancarkan rezekinya.
7. Teman-teman Kontrakan GBA (Rayhan, Dhimas, Billy, dan Ilham) yang selalu ada sedari awal kuliah hingga selesainya Tugas Akhir ini. Semoga kalian selalu diberi kesehatan dan kelancaran.

8. Teman - teman kelas EL-42-02, semoga kalian selalu diberi kesehatan dan kelancaran hingga akhir perkuliahan.
9. Teman – teman S1 Teknik Elektro 2018 dan Keluarga Mahasiswa Teknik Elektro yang telah mewadahi penulis untuk berkembang semasa perkuliahan.
10. Teman-teman UKM Paskibra Telkom University sebagai keluarga pertama saya di Telkom University ini ketika dalam masa awal perkuliahan, semoga kalian selalu diberi kesehatan dan dilindungi Allah SWT.
11. Bagas Wibisono yang telah menjadi rekan kelompok Tugas Akhir selama perkuliahan dikala susah dan senang disaat pengerjaan Tugas Akhir.
12. Annisa Dewi Lestari sahabat saya dalam keadaan susah dan senang sedari perkuliahan tingkat akhir, semoga perkuliahannya diberikan kelancaran dan diberikan kesehatan selalu.
13. Untuk teman-teman yang namanya tidak dapat disebutkan satu persatu, terima kasih atas dukungan dan bantuan yang tulus diberikan kepada penulis.



## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
LEMBAR PERNYATAAN ORISINALITAS .....	ii
ABSTRAK .....	iii
ABSTRACT .....	iv
KATA PENGANTAR .....	v
UCAPAN TERIMA KASIH .....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan dan Manfaat .....	2
1.4. Batasan Masalah .....	3
1.5. Sistematika Penulisan .....	3
BAB II .....	5
TINJAUAN PUSTAKA .....	5
2.1. Desain Solusi Deteksi Bahasa Isyarat .....	5
2.2. SIBI (Sistem Isyarat Bahasa Indonesia) .....	6
2.3. Penelitian Terkait .....	7
2.4. Radar .....	10
2.5. Radar FMCW .....	11
2.6. <i>Fast Fourier Transform</i> (FFT) .....	13
2.7. Menghilangkan <i>DC Component &amp; Clutter Suppression</i> .....	14

2.8	<i>Deep Learning</i>	15
2.8.1	<i>Convolutional Neural Network</i>	15
2.8.2	<i>Convolutional Layer</i>	16
2.8.3	<i>Activation Rectified Linear Units (ReLU)</i>	16
2.8.4	<i>Pooling Layer</i>	17
2.8.5	<i>Fully Connected Layer</i>	18
2.8.6	<i>Softmax</i>	18
2.8.7	<i>Loss Categorical Cross Entropy</i>	18
2.8.8	<i>Confusion Matrix</i>	19
2.8.9	<i>Accuracy</i>	20
2.8.10	<i>Precision</i>	20
2.8.11	<i>Recall</i>	20
2.8.12	<i>F-1 Score</i>	20
<b>BAB III PERANCANGAN SISTEM</b>		21
3.1.	<b>Desain Sistem</b>	21
3.1.1.	<b>Diagram Blok Sistem</b>	21
3.1.2.	<b>Fungsi dan Fitur</b>	22
3.2.	<b>Desain Perangkat Keras</b>	23
3.3.	<b>Desain Perangkat Lunak</b>	25
3.3.1.	<b>Dataset</b>	25
3.3.2.	<b>Diagram Alir Sistem dan Training</b>	26
<b>BAB IV HASIL DAN ANALISIS</b>		28
4.1	<b>Pengujian Sistem</b>	28
4.2	<b>Pengambilan Dataset</b>	28
4.3	<b>Metode Pre-processing Data</b>	30
4.4	<b>Perancangan arsitektur (model) Deep Learning CNN</b>	36

4.5	Pengujian hasil <i>training</i> dan hasil <i>testing</i> arsitektur (model) .....	38
BAB V KESIMPULAN DAN SARAN .....		47
5.1	Kesimpulan .....	47
5.2	Saran .....	47
DAFTAR PUSTAKA .....		49
LAMPIRAN .....		54

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Desain Konsep Solusi .....	5
<b>Gambar 2. 2</b> Huruf SIBI Sesuai Dengan Kamus .....	7
<b>Gambar 2. 3</b> Angka SIBI Sesuai Kamus SIBI.....	7
<b>Gambar 2. 4</b> Diagram Blok Prinsip Dasar Kerja Radar .....	11
<b>Gambar 2. 5</b> Diagram Blok Radar FMCW .....	12
<b>Gambar 2. 6</b> Arsitektur CNN.....	16
<b>Gambar 2. 7</b> Ilustrasi Proses <i>Convolutional Layer</i> .....	16
<b>Gambar 2. 8</b> Aktivasi ReLU .....	17
<b>Gambar 2. 9</b> Proses Dari <i>Pooling Layer</i> .....	17
<b>Gambar 2. 10</b> Proses <i>Fully Connected Layer</i> .....	18
<b>Gambar 3. 1</b> Diagram Blok Sistem.....	21
<b>Gambar 3. 2</b> Desain Perangkat Keras .....	23
<b>Gambar 3. 3</b> uRAD Raspberry Pi v1.2 .....	24
<b>Gambar 3. 4</b> Raspberry Pi 4.....	25
<b>Gambar 3. 5</b> Diagram Alir Sistem .....	26
<b>Gambar 3. 6</b> Diagram Alir <i>Training</i> .....	27
<b>Gambar 4. 1</b> Tinggi Tripod dan Radar .....	29
<b>Gambar 4. 2</b> Jarak Radar ke Objek Target .....	30
<b>Gambar 4. 3</b> Bentuk Sinyal I dan Q dalam 2 dimensi .....	31
<b>Gambar 4. 4</b> <i>Plot Spectrogram FFT</i> .....	33
<b>Gambar 4. 5</b> <i>Plot Spectrogram Range</i> Indeks Frekuensi 2048 Sampai 4096.....	33
<b>Gambar 4. 6</b> <i>Plot Spectrogram</i> Hasil Peredaman <i>Clutter</i> .....	34
<b>Gambar 4. 7</b> <i>Plot Spectrogram</i> Hanya Target Terdeteksi.....	35
<b>Gambar 4. 8</b> Ilustrasi Proses <i>Cropping</i> Per 2.25 Detik .....	36
<b>Gambar 4. 9</b> <i>Plot Accuracy</i> dan <i>Loss Max Pooling Layer</i> .....	40
<b>Gambar 4. 10</b> <i>Plot Accuracy</i> dan <i>Loss Average Pooling Layer</i> .....	40
<b>Gambar 4. 11</b> Arsitektur CNN 6 <i>Layer</i> Konvolusi <i>Max Pooling Layer</i> .....	41
<b>Gambar 4. 12</b> Arsitektur CNN 6 <i>Layer</i> Konvolusi <i>Average Pooling Layer</i> .....	43
<b>Gambar 4. 13</b> <i>Confusion Matrix Max Pooling Layer</i> .....	43
<b>Gambar 4. 14</b> <i>Confusion Matrix Average Pooling Layer</i> .....	44

## DAFTAR TABEL

<b>Tabel 2. 1</b> Penelitian Terkait Sebelumnya.....	8
<b>Tabel 2. 2</b> Tabel <i>Confusion Matrix Binary Classification</i> .....	19
<b>Tabel 3. 1</b> Tabel Spesifikasi uRAD Raspberrh Pi v1.2 .....	24
<b>Tabel 3. 2</b> Tabel Spesifikasi Raspberry Pi 4 .....	25
<b>Tabel 4. 1</b> Tabel Konfigurasi uRAD .....	28
<b>Tabel 4. 2</b> Pengaruh Jumlah <i>Layer</i> Konvolusi, <i>Max Pooling Layer</i> , dan Jumlah <i>Epoch</i> .....	36
<b>Tabel 4. 3</b> Pengaruh Jumlah <i>Layer</i> Konvolusi, <i>Max Pooling Layer</i> , dan Jumlah <i>Epoch</i> .....	37
<b>Tabel 4. 4</b> Tabel <i>Classification Report Max Pooling Layer</i> .....	45
<b>Tabel 4. 5</b> Tabel <i>Classification Report Average Pooling Layer</i> .....	45

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Penyandang disabilitas khususnya tunarungu dan tunawicara mengalami kesulitan untuk berinteraksi secara lisan maupun fisik, salah satu interaksi ialah ketika menggunakan perangkat komputasi untuk kegiatan pendidikan [1]. Pada tahun 2010 Biro Pusat Statistik (BPS) mendata bahwa jumlah penyandang disabilitas di Indonesia mencapai 5% atau sekitar 22,5 juta. Sementara itu pada tahun 2019, Sistem Informasi Manajemen Disabilitas (SIMPD) Kementerian Sosial RI menunjukkan bahwa kehadiran anak penyandang disabilitas di Indonesia mencapai 7,03%. Negara Indonesia menggunakan dua bahasa isyarat yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) [2]. Perlu diperhatikan bahwa masyarakat di Indonesia mempunyai kesulitan untuk berinteraksi serta berkomunikasi dengan penyandang tunarungu dan tunawicara. Maka dirancanglah sebuah sistem yang dapat menerjemahkan Sistem Bahasa Isyarat Indonesia (SIBI) huruf alfabet dan angka. Kehadiran sistem tersebut diharapkan masyarakat awam nondisabilitas dapat memahami SIBI melalui huruf alfabet dan angka.

Dalam mitigasi dalam mempermudah masyarakat Indonesia dalam memahami bahasa isyarat, penelitian dengan sistem *Hand Gesture Recognition* dan *Sign Language Recognition* sangat penting untuk dirancang. Penelitian sebelumnya yang terkait dengan *Sign Language Recognition* dengan mengimplementasikan bahasa SIBI yang menggunakan sensor gerak sebagai perangkat untuk mengambil data pernah dikembangkan. Menggunakan bantuan *leap motion* dan memiliki tingkat akurasi sebesar 78% dalam mendeteksi semua gerakan kata komunikasi sederhana dari sistem dan tingkat akurasi 84% untuk mendeteksi gerakan huruf dan angka [3]. Penelitian lain yang menggunakan Sistem Bahasa Isyarat Indonesia (SIBI) pernah dikembangkan dengan menggunakan *WebCam* dan metode *Machine Learning Artificial Neural Network*. Gerakan tangan SIBI didapatkan dengan mengevaluasi kontur representasi dari segmentasi gambar sarung tangan dipakai

oleh pengguna. Akurasi terjemahan mencapai 90% dengan gambar sampel 100 untuk setiap gerakan dan iterasi 90.000 kali per *training* data [4]. Untuk mempermudah penyandang tunarungu dan tunawicara berinteraksi dengan masyarakat nondisabilitas, penggunaan sensor atau kamera memiliki kekurangan dibanding dengan menggunakan radar. Kelebihan yang dimiliki oleh radar yaitu privasi pengguna akan lebih terjaga, dapat mendeteksi tanpa perlu intensitas cahaya, dan dapat mendeteksi pergerakan kecil (*small displacement*). Dalam penelitian ini penggunaan radar *Frequency Modulated Continuous Wave* (FMCW) diusulkan. Penggunaan radar FMCW yaitu dapat mendeteksi target dengan jarak yang luas dan kecepatan yang efektif untuk mendapatkan informasi yang valid. Selain itu pemilihan radar FMCW dikarenakan radar jenis ini menggunakan daya yang kecil dan desain cukup sederhana sehingga tidak menggunakan banyak ruang [5].

Penulis merancang dan mengimplementasikan sebuah sistem klasifikasi bahasa isyarat yang didukung oleh penelitian-penelitian sebelumnya. Sistem klasifikasi bahasa isyarat SIBI ini akan menjadi sarana media komunikasi serta interaksi bagi para penyandang tunawicara dan tunarungu dengan masyarakat nondisabilitas. Penggunaan radar FMCW didukung dengan metode *Deep Learning* dengan algoritma *Convolutional Neural Network* (CNN) dalam sistem klasifikasi bahasa isyarat.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang di atas, maka permasalahan yang dapat diambil adalah bagaimana rancangan sistem pengenalan bahasa isyarat SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan pergerakan statik tangan berbasis Radar FMCW.

## **1.3. Tujuan dan Manfaat**

Berdasarkan uraian rumusan masalah maka tujuan dan manfaat yang ingin dicapai dari penyusunan tugas akhir ini sebagai berikut:

1. Mengimplementasikan sistem radar FMCW yang dapat digunakan untuk pengenalan *hand gesture* dan *sign language* dengan tingkat akurasi klasifikasi lebih dari 90%.
2. Data yang didapatkan oleh radar FMCW didukung dari metode *Deep Learning* dengan algoritma *Convolutional Neural Network* (CNN) agar mendapatkan hasil yaitu 3 jenis huruf alfabet dan 2 angka isyarat.

Manfaat yang diharapkan dalam penelitian Tugas Akhir ini adalah untuk diimplementasikan kepada tunarungu dan tunawicara dalam pengenalan bahasa isyarat yang digunakan untuk membantu mereka dalam berinteraksi dimasyarakat.

#### **1.4. Batasan Masalah**

Adapun batas ruang lingkup dari penelitian ini antara lain:

1. Gestur yang dideteksi berdasarkan pergerakan jari tangan namun lengan diam tidak bergerak (statik).
2. Posisi gestur tangan kurang lebih sejajar dengan radar.
3. Penelitian ini tidak membahas desain pembuatan radar.
4. Huruf alfabet dan Angka sesuai dengan Kamus SIBI.
5. Jarak radar dengan tangan sejauh kurang lebih 120 cm.
6. Penelitian ini belum terdapat sistem *real time* dan *auto detection* target.
7. Pengklasifikasian huruf alfabet dan angka SIBI menggunakan *Deep Learning* dengan algoritma *Convolutional Neural Network* (CNN).

#### **1.5. Sistematika Penulisan**

Berikut ini merupakan sistematika penulisan buku tugas akhir ini:

##### **1. BAB I PENDAHULUAN**

Bab I berisi latar belakang masalah, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

##### **2. BAB II TINJAUAN PUSTAKA**



Bab II berisi penjelasan desain konsep solusi, teori yang digunakan pada penelitian ini seperti radar FMCW, *preprocessing*, dan metode klasifikasi.

### 3. BAB III PERANCANGAN SISTEM

Bab III berisi penjelasan penggambaran sistem secara menyeluruh, baik penggunaan perangkat keras beserta pemilihan komponen dan perangkat lunak.

### 4. BAB IV HASIL DAN ANALISIS

Bab IV berisi hasil pengujian yang dilakukan dari beberapa parameter dan analisis dari hasil pengujian yang didapatkan.

### 5. BAB V KESIMPULAN DAN SARAN

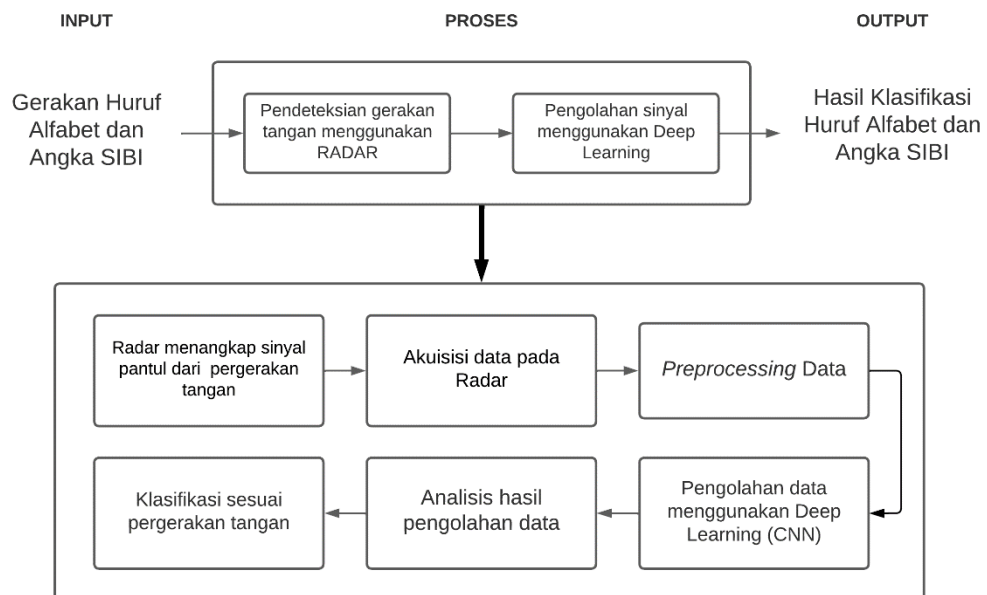
Bab V berisi kesimpulan dari hasil penelitian dan saran yang diberikan untuk pengembangan alat dan sistem pada penelitian selanjutnya.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Desain Solusi Deteksi Bahasa Isyarat

Konsep solusi yang dirancang oleh penulis dapat ditunjukkan pada **Gambar 2.1**. Implementasi sebuah sistem radar FMCW yang mampu mendeteksi gerakan statik tangan dengan menggunakan metode *Deep Learning* sebagai pengklasifikasian yang sesuai dengan pergerakan bahasa isyarat. Sistem Isyarat Bahasa Indonesia (SIBI) merupakan bahasa isyarat yang digunakan pada Tugas Akhir ini.



**Gambar 2. 1** Desain Konsep Solusi

**Gambar 2.1** merupakan desain konsep solusi yang dirancang untuk membuat sistem pengenalan gestur statik tangan berbasis radar FMCW. Radar FMCW akan terus memancarkan dan menerima gelombang untuk mendeteksi gestur statik tangan huruf alfabet dan angka SIBI. Data gelombang yang diterima sudah dalam bentuk data digital (*raw data*), karena radar FMCW sudah terintegrasi dengan Raspberry Pi. Data digital tersebut diproses melalui tahap *preprocessing* dan dikumpulkan dalam bentuk *dataset*, apabila *dataset* sudah terkumpul

selanjutnya akan diolah menggunakan metode *Deep Learning CNN*. Setelah proses pengolahan data maka didapatkan hasil klasifikasi huruf alfabet dan angka SIBI.

## 2.2. SIBI (Sistem Isyarat Bahasa Indonesia)

Sistem Isyarat Bahasa Indonesia (SIBI) adalah sarana media untuk komunikasi sesama penyandang tunarungu dan tunawicara dalam masyarakat dengan bahasa isyarat yang telah dibakukan. SIBI digunakan dengan alasan jenis bahasa isyarat SIBI adalah bahasa isyarat standar dan sudah dibakukan sesuai dengan tata bahasa, kata awalan, dan imbuhan. Kementrian Pendidikan dan Kebudayaan sudah menyepakati untuk menggunakan SIBI khususnya di SLB (Sekolah Luar Biasa) dan institusi pendidikan [1], karena setiap daerah di Indonesia memiliki arti bahasa isyarat tangan yang sama maka bahasa SIBI tidak dipengaruhi oleh Bahasa daerah.

Pada tahun 2001 pemerintah telah menyebarluaskan bahasa SIBI pada sekolah atau instansi pendidikan. Dalam Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia No. 0161/U/1994 (Kemendikbud: 1997) Sistem Isyarat Bahasa Indonesia (SIBI) telah dibakukan [2]. SIBI dapat mudah dipelajari karena sudah sesuai dengan Ejaan Yang Disempurnakan (EYD). Penulis dalam penelitian ini menggunakan beberapa huruf alfabet dan isyarat angka yang umum digunakan dalam kehidupan sehari-hari. Bahasa SIBI memiliki berbagai macam bentuk seperti, ekspresi, kata, huruf dan angka. Bahasa SIBI dengan bentuk huruf dapat ditunjukkan pada **Gambar 2.2** dan bentuk angka dapat ditunjukkan pada **Gambar 2.3**.



**Gambar 2. 2** Huruf SIBI Sesuai Dengan Kamus



**Gambar 2. 3** Angka SIBI Sesuai Kamus SIBI

### 2.3. Penelitian Terkait

Referensi penulis dalam tugas akhir ini menggunakan beberapa penelitian yang terkait dengan sistem *Hand Gesture Recognition* dan *Sign Language Recognition* menggunakan radar FMCW dan klasifikasi menggunakan metode

*Deep Learning* dengan algoritma CNN. Berikut referensi yang terkait pada tugas akhir ini dalam Tabel 2.1 sebagai berikut:

**Tabel 2. 1** Penelitian Terkait Sebelumnya

Referensi	Judul Jurnal	Penulis (Tahun)	Sistem yang Diteliti	Metode	Kesimpulan
[4]	<i>Implementation of Real-Time Static Hand Gesture Recognition Using Artificial Neural Network</i>	Lita Yusnita, Rosalina, Rusdianto Roestam, and Raden Bagio Wahyu (2017)	SIBI <i>Hand Gestures Recognition</i>	ANN	Metode ANN dengan menggunakan <i>webcam</i> . Gerakan tangan diperoleh dengan mengevaluasi kontur representasi dari segmentasi gambar sarung tangan dipakai oleh pengguna. Akurasi terjemahan mencapai 90% dengan gambar sampel 100 untuk setiap gerakan.
[6]	<i>Improving Classification Accuracy of Hand Gesture Recognition Based on 60 GHz FMCW Radar with Deep Learning Domain Adaptation</i>	Hyo Ryun Lee 1, Jihun Park 2 and Young-Joo Suh (2020)	Radar FMCW - <i>Hand gesture recognition</i>	CNN	Menggunakan radar FMCW 60 GHz untuk mendeteksi pergerakan tangan yang kecil. Menggunakan CNN ( <i>deep learning</i> ) dan mendapatkan rata-rata akurasi sebesar 98,8%.

[7]	<i>Sign Language Gesture Recognition using Doppler Radar and Deep Learning</i>	Hovannes Kulhandjiany, Prakshi Sharmay, Michel Kulhandjianz, Claude D'Amours (2019)	<i>Doppler Radar - Sign Language Gesture</i>	D-CNN	Menggunakan <i>micro doppler</i> radar mendeteksi bahasa isyarat amerika untuk dalam keperluan kewaspaadan, menggunakan algoritma DCNN dengan akurasi sebesar 87,5%.
-----	--	---	--	-------	--

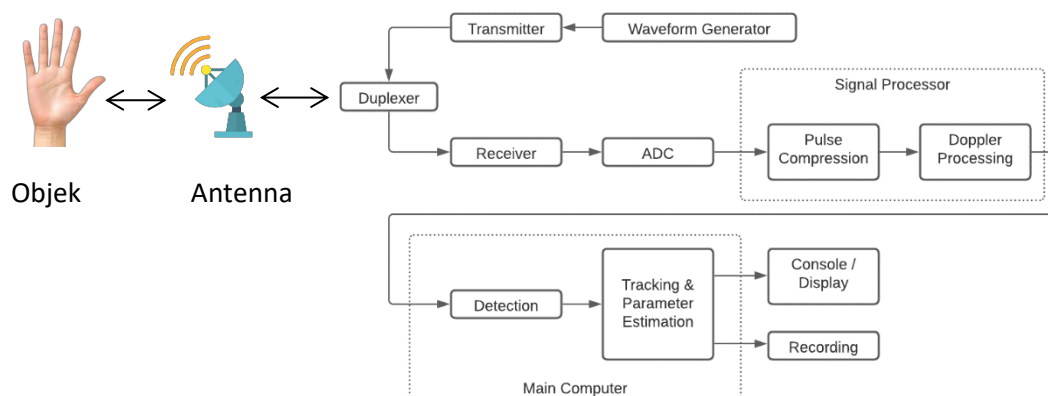
Sistem pengenalan gestur statik tangan (bahasa SIBI) dalam interaksi disabilitas dengan metode *deep learning* berbasis radar FMCW memiliki inspirasi dari penelitian sebelumnya [4]. Sistem pada penelitian ini mengambil inspirasi mengenai implementasi pengenalan *static hand gesture* dengan menggunakan huruf alfabet dan angka SIBI. Penggunaan metode pada penelitian sebelumnya menggunakan *machine learning* ANN yang di mana ANN merupakan dasar dari CNN dan menggunakan *WebCam* sebagai alat dalam pendeteksi gestur statik tangan [4]. Nilai tambah pada sistem pengenalan gestur statik tangan (bahasa SIBI) dengan berbasis radar FMCW memiliki keunggulan dibandingkan menggunakan *WebCam* dalam sistem pengenalan. Sistem pengenalan gestur statik tangan dibutuhkan pemodelan metode *deep learning* CNN yang dimana penelitian sebelumnya pernah dilakukan dengan menggunakan radar FMCW dan *deep learning* CNN [6]. Dalam penelitian sistem pengenalan gestur statik tangan (bahasa SIBI) mengambil peran mengimprovisasi metode penelitian sebelumnya, letak improvisasi pada penelitian sebelumnya adalah pada pemodelan *deep learning* CNN. Pemodelan *deep learning* CNN pada penelitian ini dilakukan dengan menambahkan *layer* yang kompleks sehingga hasil klasifikasi menjadi lebih baik dari penelitian sebelumnya. Sistem pengenalan gestur statik tangan tidak luput dari penggunaan bahasa SIBI sebagai objek yang dideteksi. Pada penelitian [7], menggunakan bahasa isyarat amerika yang di mana bahasa SIBI merupakan bahasa isyarat yang mengambil inspirasi dari bahasa isyarat amerika atau *American Sign Language*

(ASL). Penggunaan alat deteksi pada penelitian sebelumnya menggunakan *doppler* radar namun pada penelitian pengenalan gestur statik tangan (bahasa SIBI) memodifikasi pada penggunaan radar. Penggunaan radar FMCW yang didukung dengan metode *deep learning* CNN. Penelitian ini berhasil memodifikasi penggunaan radar sehingga mendapatkan nilai tambah dan hasil klasifikasi serta akurasi yang baik dari penelitian sebelumnya.

## 2.4. Radar

Radar (*Radio Detection and Ranging*) adalah suatu sistem yang memiliki fungsi mendapatkan informasi berupa jarak, *azimuth*, dan kecepatan dengan mendeteksi target yang berada di lingkungan sekitar [8][9]. Prinsip dasar kerja Radar seperti pada **Gambar 2.4** adalah dengan memancarkan gelombang elektromagnetik yang akan mengenai sebuah target yang dilaluinya, target memantulkan sinyal yang telah dipancarkan oleh radar ke antena *receiver*. Pantulan sinyal transmisi radar ke antena *receiver* akan melalui tahap analisis agar mendapatkan informasi yang dibutuhkan. Informasi tersebut dapat berupa jarak, kecepatan dan identifikasi objek. Radar memiliki redaman yang dapat memperlemah pantulan sinyal, tetapi masih dapat diterima dan diperkuat oleh *amplifier* pada radar [10].

Perbandingan menggunakan citra kamera atau sensor, radar memiliki beberapa keunggulan, radar dapat mendeteksi sebuah target yang kecil dari jarak yang dekat atau jauh, dan pengaruh cuaca tidak dapat mempengaruhi radar saat mendeteksi suatu objek. Pengguna radar akan mendapatkan privasi yang terjaga karena radar tidak memerlukan intensitas cahaya seperti kamera, dan ketika menggunakan radar pengguna tidak memerlukan tempat atau ruang yang banyak.



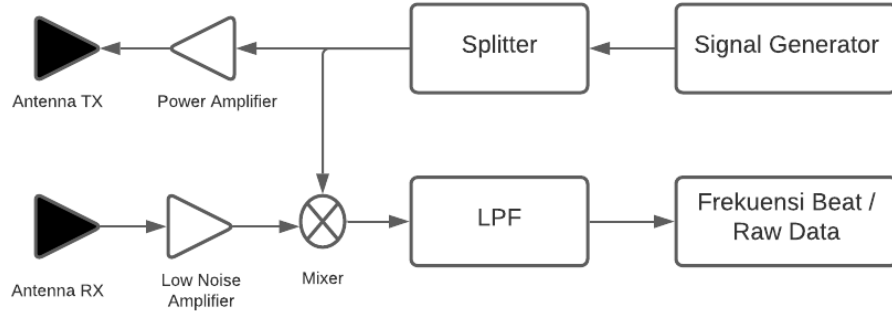
### **Gambar 2. 4** Diagram Blok Prinsip Dasar Kerja Radar

Sebelum *transmitter* memancarkan sinyal, sinyal tersebut sudah dibangkitkan oleh *waveform generator*. Untuk memisahkan antara sinyal pemancar (*transmitter*) dan penerima (*receiver*), sinyal tersebut akan diteruskan ke *duplexer*. Pada radar terdapat antena yang digunakan untuk mengirimkan sinyal elektromagnetik yang dihasilkan dari *transmitter* melalui *reflector antenna*. Setelah sinyal elektromagnetik mengenai suatu target, sinyal elektromagnetik akan terpantul ke *reflector* antena dan diterima *receiver*. Karena terdapat redaman pada radar maka sinyal akan diperkuat dengan *amplifier*, sinyal tersebut masih berbentuk data analog sehingga akan melalui ADC untuk menghasilkan sinyal ke dalam bentuk diskrit. Dalam pemrosesan sinyal, fungsinya adalah untuk mengambil sinyal yang diperlukan dan menghilangkan sinyal yang tidak perlu (*noise*). Kemudian komputer utama dengan fungsi mengolah sinyal yang telah diterima menjadi data yang akan diperlukan.

### **2.5. Radar FMCW**

Frequency Modulated Continuous Wave (FMCW) merupakan radar yang mengeluarkan gelombang kontinu, dengan memodulasi frekuensi dalam rentang waktu dan periode tertentu [11]. Dalam radar FMCW satu periode menghasilkan sinyal dengan frekuensi yang meningkat [12][13]. Radar FMCW dapat berfungsi dalam mengukur atau mendeteksi suatu objek dengan jarak, posisi dan kecepatan tertentu. Untuk mendapatkan jangkauan objek yang lebih luas dan resolusi yang lebih tinggi radar FMCW diberikan keunggulan dapat mendeteksi pergeseran yang kecil (*small displacement*) [14]. Cara kerja radar FMCW dapat ditunjukkan dengan diagram blok pada **Gambar 2.5**.





**Gambar 2. 5** Diagram Blok Radar FMCW

*Signal generator* akan membangkitkan sinyal FMCW yang di mana sinyal tersebut akan dipancarkan oleh antena TX (*Transmitter*) untuk mengenai suatu objek. Keluaran dari sinyal yang dipancarkan ke suatu objek akan masuk ke dalam *Splitter* yang berfungsi membagi 2 *channel* menjadi sinyal referensi yang terhubung ke *mixer* dan sinyal yang dipancarkan terhubung dengan *amplifier* dan antena TX (*Transmitter*). Persamaan sinyal yang ditransmisikan dapat ditulis dalam bentuk matematika pada persamaan 2.1.

$$S_T(t) = A_T \cos \left( 2\pi \left( f_0 + \frac{\Delta_f}{T} t \right) t \right) \quad (2.1)$$

Amplitudo yang ditransmisikan adalah  $A_R$ , frekuensi terendah dari bandwidth adalah  $f_0$ ,  $\Delta_f$  dan  $T$  mewakili *bandwidth* dan amplitudo *bandwidth*. Saat sinyal yang ditransmisikan mengenai sebuah target, lalu target tersebut memantulkam kembali sinyal transmisi yang akan diterima oleh antena *receiver*. Sinyal yang ditransmisikan yang telah tertunda dan dilemahkan oleh atenuasi merupakan sinyal pantul tersebut. Sinyal pantul yang diterima memiliki persamaan seperti dalam persamaan 2.2.

$$S_R(t) = A_R \cos \left( 2\pi \left( f_0 + \frac{\Delta_f}{T} t \right) (t - \Delta t) \right) \quad (2.2)$$

$A_R$  adalah amplitudo yang diterima,  $\Delta t$  adalah propagasi waktu *delay* dari jarak target. Dalam *mixer*, sinyal transmisi dan sinyal pantul akan dikalikan secara terus menerus yang akan menghasilkan penambahan dan pengurangan fasa. *Output* dari *mixer* dapat ditunjukkan dengan persamaan (2.3).

$$S_M(t) = A_M \left[ \cos \left( \left( 4\pi f_0 + 4\pi \frac{\Delta_f t}{T} \right) t \right) + \cos \left( 2\pi f_0 \Delta t + 2\pi \frac{\Delta_f t}{T} \Delta t \right) \right] \quad (2.3)$$

*Low Pass Filter* (LPF) berfungsi untuk menghilangkan elemen penjumlahan fasa pada rumus (2.4), dikarenakan frekuensi *beat* yang akan diperoleh terkait dengan elemen reduksi fasa [15]. Sinyal beat diterima setelah terjadi *filtering*, pada persamaan 2.4.

$$S_{LPF} = A_{LPF} \cos \left( 2\pi f_0 \Delta t + 2\pi \frac{\Delta_f t}{T} \Delta t \right) \quad (2.4)$$

Selanjutnya dari turunan pertama terhadap fasa akan mendapatkan persamaan frekuensi *beat* [16] :

$$f_{beat} = 2\pi \frac{\Delta_f}{T} \Delta t \quad (2.5)$$

Setelah sinyal yang ditransmisikan dan sinyal yang dipantulkan digabungkan, untuk mengubah domain waktu ke dalam frekuensi dari *output* LPF maka diperlukan *Fast Fourier Transform* (FFT) untuk menghasilkan puncak frekuensi *beat* pada spektrum frekuensinya [16].

## 2.6 *Fast Fourier Transform* (FFT)

Mengubah sinyal analog menjadi sinyal digital dengan basis frekuensi yang menggunakan operasi perhitungan matematika merupakan definisi dari *Fast Fourier Transform* (FFT) [17][18]. Dalam fungsi eksponensial yang kompleks, metode FFT dapat membagi sinyal dalam bentuk frekuensi yang berbeda-beda. Sinyal-sinyal dalam sistem komunikasi bersifat kontinu dan hasilnya dapat digunakan untuk transformasi *fourier* sehingga metode FFT merupakan algoritma menghitung transformasi *fourier* diskrit [17]. Persamaan *Fast Fourier Transform* dapat ditunjukkan pada persamaan 2.6.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{n}{N} k} \quad (2.6)$$

## 2.7 Menghilangkan *DC Component* & *Clutter Suppression*

Sebuah objek selain target yang di mana objek tersebut merupakan objek statik atau kondisi lingkungan yang dapat menyebabkan terganggunya pendeteksian radar ketika memantulkan sinyal disebut dengan *Clutter*. *DC Component* merupakan sinyal yang mengambang di atas 0 sumbu x, dan setelah sinyal melalui *Fast Fourier Transform* terdapat *noise* pada frekuensi 0 Hz [19]. Sinyal yang tidak perlu harus dihilangkan agar dapat mengekstrak hanya sinyal yang sesuai dengan gerakan. Dalam mencegah penurunan kinerja deteksi gerakan manusia, metode menghilangkan *DC Component* dan meredam *Clutter* diperlukan dengan cara pengurangan rata-rata [20][21]. Dalam penghilangan *DC Component* dapat dinyatakan dalam persamaan 2.7.

$$\mathbf{x} = \mathbf{x}_{N_s} - \frac{1}{N_s} \sum_{n=0}^{N_s-1} x_{N_s}[n] \quad (2.7)$$

Di mana  $\mathbf{x}_{N_s} = [x[0], x[1], x[2], \dots, x[N_s - 1]]$  adalah sinyal dari radar FMCW dengan jumlah sampel  $N_s$ .

Dalam peredaman *clutter* dapat dinyatakan dengan persamaan 2.8.

$$\mathbb{Y}^{(N_p)} = \mathbb{X}^{(N_p)} - \frac{1}{N_p} \left[ \sum_{i=0}^{N_p} X^{(i)}[0], \sum_{i=0}^{N_p} X^{(i)}[1], \dots, \sum_{i=0}^{N_p} X^{(i)}[K-1] \right] \times J_{1,N_p} \quad (2.8)$$

$$\mathbb{X}^{(N_p)} = [\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N_p-1)}] \quad (2.9)$$

Di mana persamaan 2.9 merupakan kolom spektrogram dan  $N_p$  merupakan jumlah periode (*frame*),  $\mathbf{X}^{(i)} = [X^{(i)}[0], X^{(i)}[1], \dots, X^{(i)}[K-1]]$  adalah baris pada spektrogram,  $K$  adalah jumlah indeks frekuensi,  $i$  adalah indeks frekuensi per satu *frame*, dan  $J_{1,N_p}$  merupakan matriks  $1 \times N_p$  yang semua elemennya berisi 1.

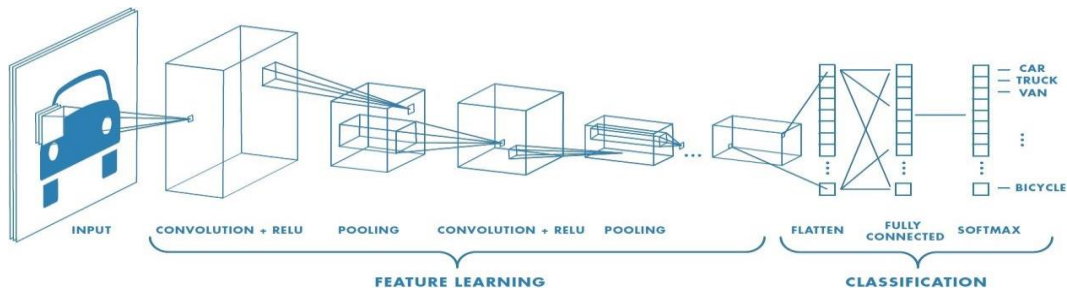
## 2.8 *Deep Learning*

Metode *Deep Learning* didalam sistem pendeteksian gerakan bahasa isyarat dengan tangan bergerak statik berfungsi sebagai metode pendekatan untuk mempelajari fitur tingkat tinggi secara otomatis dan dapat memiliki *performance* yang unggul untuk radar FMCW. Metode *Deep Learning* dirancang untuk pengenalan aktivitas pada radar sesuai dengan dimensi yaitu dua dimensi.

### 2.8.1 *Convolutional Neural Network*

*Convolutional Neural Network* (CNN) merupakan arsitektur *Deep Learning* pertama dengan urutan tingkatan lapisan [22]. CNN adalah algoritma yang berkembang dari MLP (*Multilayer Perceptron*) yang dirancang untuk mengolah data dua dimensi. Algoritma CNN sudah banyak diterapkan untuk data gambar dan sudah dikategorikan sebagai *Deep Neural Network*. Perbedaan MLP dan CNN hanya pada ukuran *neuron*, MLP mengadopsi *neuron* dengan ukuran hanya satu dimensi sementara CNN setiap *neuron* digambarkan dalam bentuk dua dimensi [23]. CNN terinspirasi oleh struktur korteks visual yang terdiri dari sel sederhana dan sel kompleks. Kedua sel tersebut mengadopsi empat ide utama yaitu *local connections*, *paramater sharing*, *pooling*, dan *multi-layers*. Dalam CNN terdapat operasi konvolusi yang mengubah perkalian matriks umum dalam *General Neural Network*. Dengan cara mengubah tersebut, kompleksitas *Neural Network* berkurang karena adanya penurunan jumlah bobot [22].

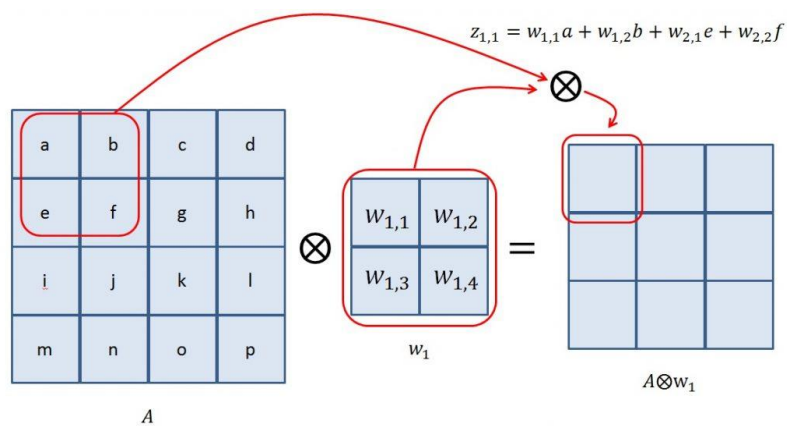
Dalam penelitian ini saat radar membaca pergerakan statik tangan akan menghasilkan data dua dimensi dalam klasifikasi bahasa isyarat. *Convolutional Neural Network* (CNN) dirancang untuk mengolah data dua dimensi, CNN memiliki kedalaman jaringan yang tinggi dan telah banyak diterapkan untuk data gambar. Dalam CNN memiliki bobot, bias, dan fungsi aktivasi [23]. CNN memiliki arsitektur seperti pada **Gambar 2.6** yang terdiri dari *Convolution*, *Max-Pooling*, *Classification*, dan ReLU.



**Gambar 2. 6** Arsitektur CNN

### 2.8.2 Convolutional Layer

Proses konvolusi merupakan proses utama yang mendasari *Convolutional Neural Network* (CNN) adalah pada *Convolution Layer*. *Convolution Layer* adalah lapisan pertama yang akan memproses gambar sebagai *input* model pada arsitektur. *Input* gambar akan berkonvolusi dengan filter untuk mengekstrak fitur dari *input* gambar yang disebut dengan *Feature Maps* seperti yang diilustrasikan pada **Gambar 2.7** [23]. Output dari *Convolution Layer* akan melalui fungsi unit linier yang diperbaiki oleh ReLU dan diikuti oleh *Pooling Layer* yang menggunakan proses *Max-Pooling* dengan maksimal matriks 2x2. Setelah itu 32 *convolution layer* yang memiliki *convolution filter* berukuran 5x5 yang akan diproses oleh fungsi ReLU dan *pooling layer* dengan ukuran maksimal matriks 2x2 [7].

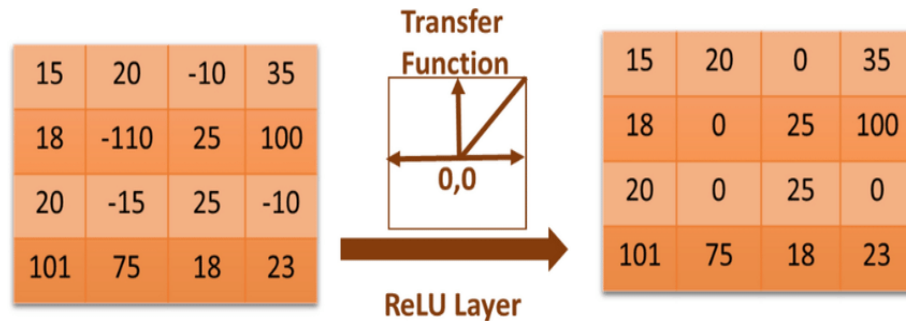


**Gambar 2. 7** Ilustrasi Proses Convolutional Layer

### 2.8.3 Activation Rectified Linear Units (ReLU)

*Convolutional Neural Network* (CNN) terdapat elemen penting yang fungsinya untuk mengubah nilai piksel negatif pada *feature maps* menjadi bernilai

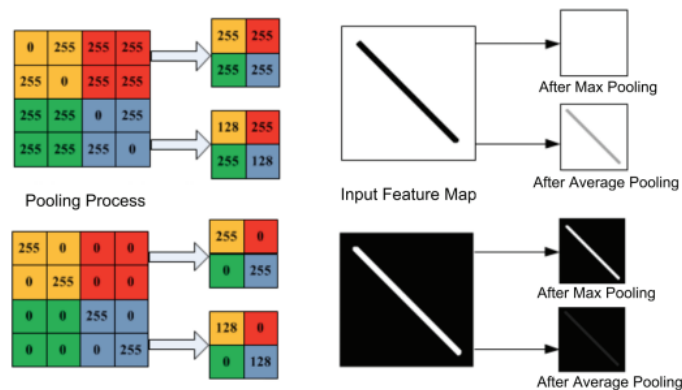
nol yang ditunjukkan pada **Gambar 2.8**, ReLU memiliki fungsi lain untuk melakukan *Thresholding* [24].



**Gambar 2. 8** Aktivasi ReLU

#### 2.8.4 Pooling Layer

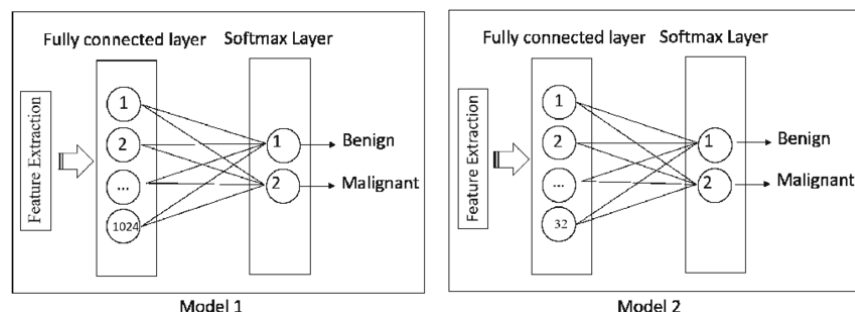
Definisi dari *pooling layer* adalah ketika menerima output dari *Convolution Layer*, pada *layer* ini ukuran data pada gambar akan dikurangi. Pada dasarnya, *pooling layer* terdiri dari filter dengan ukuran dan langkah tertentu yang bergeser diseluruh area *feature maps*. Dalam *pooling layer* terdapat 2 *layer* yang digunakan yaitu *average pooling* dan *max-pooling*. *Average Pooling* memanfaatkan rata-rata statistik pada data yang berfungsi untuk mengambil nilai dari area yang terpilih. *Max-Pooling* berfungsi untuk mendapatkan informasi nilai terbesar dari area yang terpilih [25][26]. Proses dari kedua *pooling layer* dapat diilustrasikan pada **Gambar 2.9**.



**Gambar 2. 9** Proses Dari *Pooling Layer*

### 2.8.5 Fully Connected Layer

Tidak seperti *Convolution Layer* yang di mana *neuron* terhubung hanya ke area tertentu pada *input*, namun pada *fully connected layer* menghubungkan seluruh *neuron* seperti pada **Gambar 2.10**. *Fully Connected Layer* mengubah multidimensi array *feature maps* menjadi array satu dimensi secara linier dan data dapat mudah diklasifikasikan, proses perubahan tersebut disebut dengan proses *flatten* [24][26]. *Fully Connected Layer* adalah bagian dari *Artificial Neural Network* (ANN) yang terdiri dari beberapa *neuron* dihubungkan dengan bobot penghubung.[27].



**Gambar 2. 10** Proses *Fully Connected Layer*

### 2.8.6 Softmax

Bentuk lain dari *Logistic Regression* yang dapat melakukan klasifikasi lebih dari dua kelas adalah definisi dari *Softmax*. Sesuai dengan fungsinya *softmax* berguna untuk mengubah output lapisan terakhir ke probabilitas dasar. Keuntungannya adalah ketika jumlah dari peluangnya sama dengan 1, rentang probabilitas keluaran dari 0 hingga 1 [26].

### 2.8.7 Loss Categorical Cross Entropy

*Cross-Entropy* merupakan *loss function* merupakan pengukuran perbedaan antara dua *probability* dalam rangkaian peristiwa tertentu. *Loss function* ini menyesuaikan bobot model selama proses *training*. Tujuan *cross-entropy loss function* adalah meminimalkan jumlah *loss* selama proses *training*, apabila *loss* semakin kecil maka semakin baik modelnya. *Cross-entropy* biasa digunakan dalam *multi class* klasifikasi. Apabila klasifikasi dilakukan dengan menggunakan *multi class* maka membutuhkan label untuk dikodekan sebagai kategori, maka

dipergunakanlah *categorical cross entropy* [28]. *Cross Entropy* dapat dihitung dengan menggunakan probabilitas pada persamaan 2.10.

$$LCE = - \sum_{i=1}^n t_i \log (p_i) \quad (2.10)$$

### 2.8.8 Confusion Matrix

Hasil metode pengklasifikasian dapat mengevaluasi kinerja dengan menggunakan *Confusion Matrix*. *Confusion Matrix* dapat memberikan informasi *class* yang diprediksi benar dan tidak benar oleh model yang telah dirancang [29].

**Tabel 2. 2** Tabel *Confusion Matrix Binary Classification*

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

Isi dari tabel *confusion matrix*, sebagai berikut:

1. *True Positive* (TP) adalah kondisi *class* yang sebenarnya adalah benar dan hasil prediksi juga benar.
2. *False Negative* (FN) adalah kondisi *class* yang sebenarnya adalah benar namun hasil prediksi adalah salah.
3. *True Negative* (TN) adalah kondisi *class* yang sebenarnya adalah salah dan hasil prediksi juga salah.
4. *False Positive* (FP) adalah kondisi *class* yang sebenarnya adalah salah, namun hasil prediksi adalah benar.



### 2.8.9 Accuracy

Rasio antara kumulatif data yang dapat diprediksi benar dengan total data yang diprediksi disebut *accuracy* [30]. *Accuracy* dapat dihitung dengan persamaan 2.11.

$$Acuraccy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.11)$$

### 2.8.10 Precision

*Precision* merupakan perbandingan jumlah data positif yang diprediksi benar dengan banyaknya data yang diprediksi positif oleh sistem [30]. Persamaan *precision* dapat dilihat pada persamaan 2.12.

$$Precision = \frac{TP}{TP + FP} \quad (2.12)$$

### 2.8.11 Recall

*Recall* merupakan rasio data positif yang diprediksi benar dengan semua data *class* yang sebenarnya adalah positif [30]. *Recall* dapat dilihat dari persamaan 2.13.

$$Recall = \frac{TP}{TP + FN} \quad (2.13)$$

### 2.8.12 F-1 Score

F1-Score merupakan sebagai nilai kombinasi dari rata-rata *precision* dan *recall* [30]. Persamaan F1-score dapat ditunjukkan pada persamaan 2.14.

$$F - 1 \text{ Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.14)$$

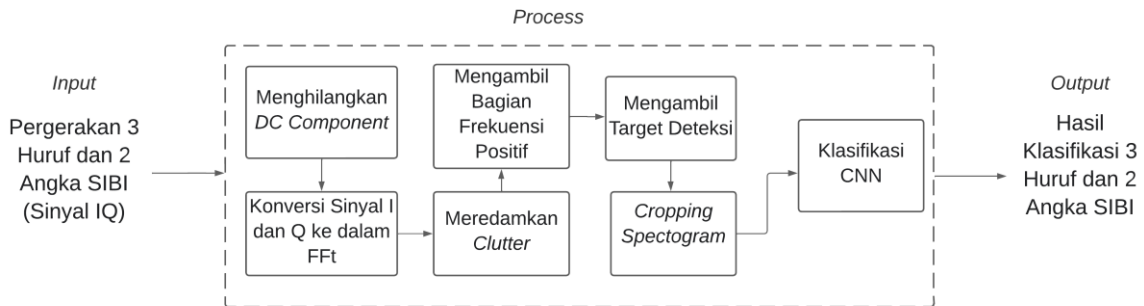
## BAB III

### PERANCANGAN SISTEM

#### 3.1. Desain Sistem

Penggunaan radar FMCW digunakan dalam mendeteksi gerakan statik tangan dalam menerjemahkan bahasa isyarat untuk interaksi penyandang disabilitas. Pendeteksi ini dilakukan dengan menggunakan uRAD radar sebagai alat pendeteksi. Radar memancarkan gelombang elektromagnetik melalui antenna *transmitter* kemudian gelombang yang terpantul akan diterima oleh antenna *receiver*. Sinyal yang terpantul dan diterima oleh *receiver* kemudian diproses pada uRAD untuk mendapatkan *raw data*. Dari *raw data* yang didapatkan akan melalui tahap *preprocessing* sebelum memasuki *deep learning*. *Raw data* yang didapatkan akan dikumpulkan menjadi dua dimensi dan dijadikan *dataset*. *Dataset* kemudian akan dipelajari menggunakan *deep learning* dengan algoritma CNN sehingga dapat diklasifikasikan jenis huruf dan angka SIBI.

##### 3.1.1. Diagram Blok Sistem



**Gambar 3. 1** Diagram Blok Sistem

Sistem pengenalan gestur statik tangan memiliki diagram blok yang menggambarkan proses kerja sistem yang ditunjukkan pada **Gambar 3.1**. Antena *transmitter* yang berada pada radar FMCW akan memancarkan sinyal yang akan diterima oleh antenna *receiver* terdeteksi pergerakan statik tangan. Dalam radar sinyal pantul tersebut diakuisisi, yang di mana referensi sinyal yang dipancarkan dan dipantulkan akan dikalikan secara terus menerus dalam radar. Keluaran dari radar menghasilkan *raw data* berupa sinyal I (*in-phase*) Q (*Quadrature*) yang berbentuk 1 dimensi dan disatukan sehingga menjadi 2 dimensi lalu sinyal I dan Q

akan diubah kedalam *Voltage*. Sinyal yang sudah dikonversi ke dalam *Voltage* tersebut akan melalui proses penghilangan *DC Component*. Sinyal I dan Q yang sudah di konversikan akan masuk ke proses konversi kembali ke dalam domain frekuensi menggunakan *Fast Fourier Transform* dengan index FFT sebesar 4096 sehingga dalam domain frekuensi memiliki range 0 sampai 4096. Sinyal yang masih dalam range 0 sampai 4096 akan dilakukan pemangkasan sinyal yang di mana hanya mengambil bagian 2048 sampai 4096 atau hanya bagian frekuensi positif saja. *Dataset* akan dilakukan pemangkasan kembali dengan hanya mengambil target gerakan saja dan akan dilakukan *cropping spectrogram* sepanjang *time series* untuk memperbanyak dataset dan menghilangkan fitur yang tidak digunakan dalam *deep learning*. Selanjutnya dataset akan di *input* ke dalam metode *deep learning* menggunakan algoritma *Convolution Neural Network* akan mengolah *dataset* dua dimensi tersebut yang akan menghasilkan klasifikasi 3 huruf alfabet dan 2 angka SIBI.

### 3.1.2. Fungsi dan Fitur

Pada tugas akhir ini, akan merancang sistem klasifikasi huruf alfabet dan angka SIBI menggunakan radar FMCW.

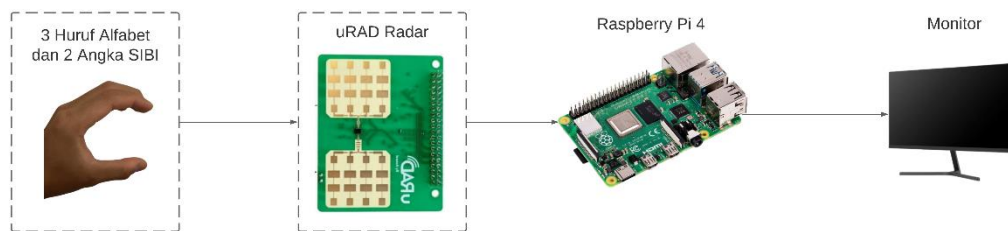
1. Radar FMCW yang berfungsi untuk mendeteksi pergerakan statik tangan huruf alfabet dan angka SIBI dengan memantulkan sinyal elektromagnetik.
2. Raspberry Pi 4 berfungsi sebagai mikrokomputer untuk menjalankan pemograman radar.
3. Metode *deep learning* dengan alogoritma CNN digunakan untuk mengklasifikasikan huruf alfabet dan angka bahasa isyarat SIBI dengan gerak statik tangan.

Perancangan tugas akhir ini memiliki fitur sebagai berikut :

1. Sistem ini dapat mendeteksi bahasa isyarat SIBI dengan gerakan statik tangan dan dapat diklasifikasikan ke dalam huruf alfabet dan angka.
2. Sistem ini menggunakan radar karena tidak memerlukan intensitas pencahayaan, dapat menjaga privasi dan resolusi yang tinggi yang sesuai dengan ukuran *bandwidth*.

### 3.2. Desain Perangkat Keras

**Gambar 3.2** merupakan desain perangkat keras yang digunakan dalam penelitian ini. Sistem pengenalan bahasa isyarat ini akan menggunakan radar FMCW uRAD Raspberry Pi v1.2 sebagai pendeteksi tangan isyarat. Radar menegeluarkan sinyal diskrit yang akan diolah dalam pengolahan data. Radar dihubungkan dengan Raspberry Pi 4 sebagai mikrokomputer dalam menggunakan uRAD radar.



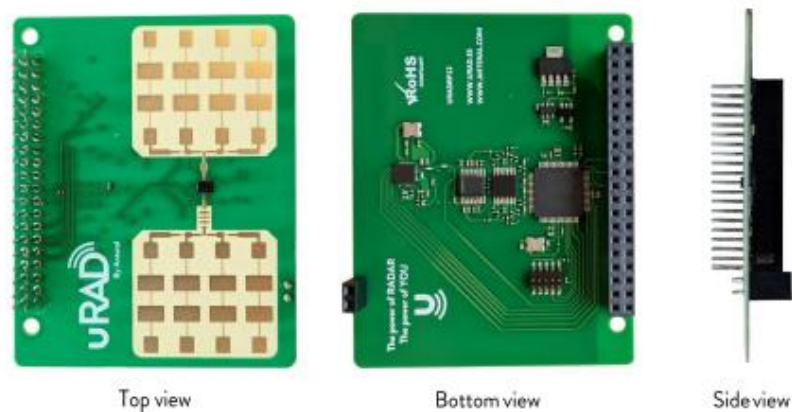
**Gambar 3. 2** Desain Perangkat Keras

Dalam merancang sistem pendeteksian bahasa isyarat SIBI dengan gerak statik tangan dibutuhkan beberapa komponen dengan spesifikasi sebagai berikut:

a) Radar FMCW uRAD Raspberry Pi v1.2

Pada **Gambar 3.3** merupakan komponen radar uRAD Raspberry Pi v1.2 yang berfungsi sebagai komponen pendeteksi bahasa isyarat SIBI dengan gestur gerak statik tangan. Komponen ini sudah terintegrasi dengan Raspberry Pi sehingga bisa langsung digunakan secara bersamaan. Kompon uRAD radar ini mempunyai jarak maksimal 15,625 meter dengan dapat dihitung dengan persamaan 3.1 [31].

$$Distance\ Max = 75 \times \frac{Ns}{BW} \quad (31)$$



**Gambar 3. 3** uRAD Raspberry Pi v1.2

**Tabel 3. 1** Tabel Spesifikasi uRAD Raspberh Pi v1.2

Kriteria	Spesifikasi
<i>Supply Voltage</i>	Min. 3.5 V, Max. 10 V
<i>Supply Current</i>	170 mA
<i>Operatin Temperature</i>	Min. -20°C Max. +65°C
<i>Frequency Bandwith</i>	Min. 24.005 GHz Max. 24.245 GHz
<i>Field Of View</i>	30 × 30 deg

b) Raspberry Pi 4

Raspberry Pi 4 pada **Gambar 3.4** adalah sebuah mikrokomputer versi terbaru yang berfungsi sebagai alat komputasi. Dalam tugas akhir ini Raspberry Pi 4 digunakan sebagai mikrokomputer dalam wadah mengakses uRAD radar yang sudah terintegrasi dengan Raspberry Pi 4. Pada Tabel 3.2 ditunjukkan spesifikasi komponen Raspberry Pi 4.



**Gambar 3. 4** Raspberry Pi 4

**Tabel 3. 2** Tabel Spesifikasi Raspberry Pi 4

Kriteria	Spesifikasi
<i>Power Source</i>	+5V, +3.3V
<i>Communication Interface</i>	UART Interface (RXD, TXD)
<i>Input Output Pins</i>	26 GPIO
RAM	8GB

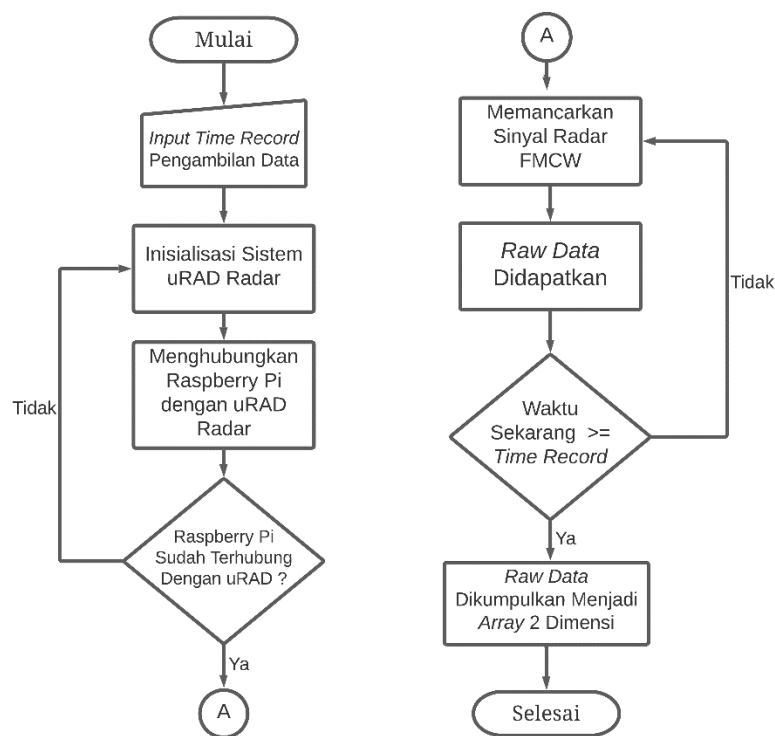
### 3.3. Desain Perangkat Lunak

#### 3.3.1. Dataset

Dalam metode deep learning terdapat *dataset* utama yang dievaluasi yaitu *dataset* dari rekaman Radar FMCW uRAD Raspberry Pi v1.2 yang memiliki dimensi (3600, 50) data *raw* sinyal I/Q. Pada *training*, penulis membagi *dataset* menjadi data *training* dan *testing* dengan rasio 25%. Tugas akhir ini menggunakan *dataset* objek pergerakan huruf alfabet dan angka SIBI. Penelitian ini akan membagi klasifikasi *dataset* menjadi beberapa sampel 3 huruf alfabet dan 2 angka bahasa SIBI.

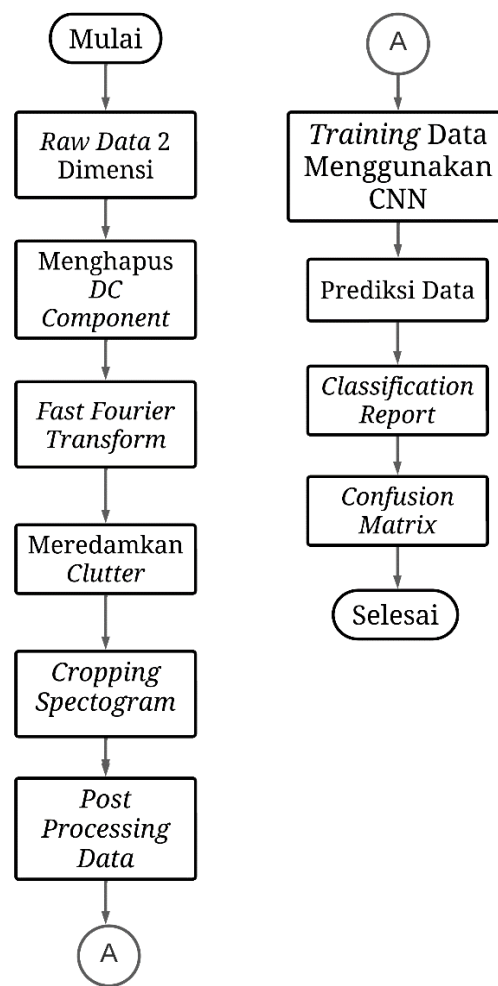
### 3.3.2. Diagram Alir Sistem dan Training

Dalam sistem klasifikasi yang dilakukan tidak hanya diproses dari radar saja namun terdapat algoritma pemograman yang dirancang dalam diagram alir sistem dan *training*.



**Gambar 3. 5** Diagram Alir Sistem

Proses diagram alir yang digambarkan pada **Gambar 3.5** berjalan ketika sistem diawali dengan *input time record* pengambilan data yang diperlukan. Ketika *time record* telah diinput selanjutnya Raspberry Pi akan menghubungkan dengan sistem uRAD radar, apabila uRAD radar terjadi *error* maka Raspberry Pi akan mencoba menghubungkan kembali ke sistem uRAD radar. Jika uRAD sudah terhubung dengan Raspberry Pi selanjutnya uRAD radar akan mulai melakukan pengambilan data dengan memancarkan sinyal radar FMCW sehingga didapatkan *raw data*. Jika *time record* sudah mencapai yang ditentukan, *raw data* yang telah didapatkan akan digabungkan kedalam satu *array 2 dimensi*.



**Gambar 3. 6** Diagram Alir *Training*

*Dataset* yang sudah selesai dalam metode *preprocessing* data akan diinput kedalam algoritma CNN seperti yang digambarkan pada **Gambar 3.6** yang di mana *input* dari *dataset* adalah dua dimensi yaitu (400, 45), dan jumlah data yang *ditraining* adalah sebesar 13335 data. Sistem akan melakukan prediksi pada data *testing*, proses ini bertujuan untuk melakukan probabilitas apakah radar sudah membaca objek secara tepat atau belum. Setelah berhasil melakukan proses prediksi selanjutnya sistem akan mengeluarkan *output classification report* dan *confusion matrix*.



## BAB IV

### HASIL DAN ANALISIS

#### 4.1 Pengujian Sistem

Pengujian sistem dilakukan pada bab hasil dan analisis ini, membahas hasil klasifikasi sistem pengenalan gerakan statik bahasa SIBI menggunakan Radar FMCW dengan metode *deep learning convolutional neural network* yang telah dirancang oleh penulis. Berikut ini merupakan beberapa langkah-langkah dalam perancangan sistem yang dilakukan:

1. Pengambilan *dataset*.
2. Metode *preprocessing* pada pengolahan *dataset*.
3. Perancangan arsitektur (model) *deep learning convolutional neural network*.
4. Pengujian hasil *training* dan hasil *testing* arsitektur (model).

#### 4.2 Pengambilan Dataset

Proses pengambilan *dataset* untuk jenis Gerakan (*class*) yang akan diuji dilakukan dengan pengambilan gerakan huruf dan angka SIBI sebanyak 5 data *class* dengan rentang waktu 180 detik per satu *class* dengan dimensi data (3600,50) atau dua dimensi. Sebelum dilakukan pengambilan data beberapa konfigurasi pengaturan pada uRAD radar harus diatur, berikut pengaturan konfigurasi pada uRAD radar:

**Tabel 4. 1** Tabel Konfigurasi uRAD

Inisiasi <i>Setting</i> uRAD Radar	
Ns	50
Mode	2 ( <i>Sawtooth</i> )
f(0)	5 MHz
<i>Bandwith</i>	240 MHz

Inisiasi di atas merupakan pengaturan yang disediakan oleh uRAD radar dalam melakukan pengambilan data pergerakan.  $N_s$  adalah jumlah sampel dalam 1 data *frame* yang di mana penulis menggunakan 50 sampel, artinya satu *frame* membutuhkan 50 milidetik. Mode 2 digunakan oleh penulis karena uRAD radar akan mendeteksi jarak dengan bentuk sinyal *sawtooth*, frekuensi awal yang digunakan sebesar 24,005 GHz, *bandwidth* yang digunakan dalam uRAD radar ketika mengambil data adalah sebesar 240 MHz karena maksimal *bandwidth* pada uRAD radar adalah 240 MHz dan ingin menggunakannya secara maksimal. *Datasheet* uRAD Radar menunjukkan bahwa uRAD radar memiliki FOV (*Field of View*) 30° vertikal dan 30° horizontal [31]. Dalam pengambilan *dataset* uRAD radar di letakkan dalam sebuah *case* yang sudah terintegrasi dengan RaspberryPi 4. Radar diletakkan *tripod* agar dapat direpresentasikan sebagai kamera, dan tinggi radar yang dipakai adalah setinggi 100 cm (1 meter), seperti yang digambarkan pada **Gambar 4.1**.



**Gambar 4. 1** Tinggi Tripod dan Radar

Dalam penelitian ini merupakan pengambilan sampel *dataset* jarak antara uRAD radar dan objek adalah sejauh 120 cm atau 1,2 meter seperti pada **Gambar 4.2**. Jarak pengambilan data tidak dapat diubah, karena dalam sistem tidak terdapat

sistem secara *real time* dan fitur *auto detection* target dalam mendeteksi target dari berbagai jarak yang melebihi atau kurang dari 120 cm. Pengambilan *dataset* terbagi menjadi 5 gestur (*class*) yaitu 3 Huruf Alfabet dan 2 Angka yaitu “Huruf C”, “Huruf F”, “Huruf T”, “Angka 3” dan “Angka 6”, lima *class* tersebut akan digunakan dalam proses *training* dan *testing* dalam sistem ini.



**Gambar 4. 2** Jarak Radar ke Objek Target

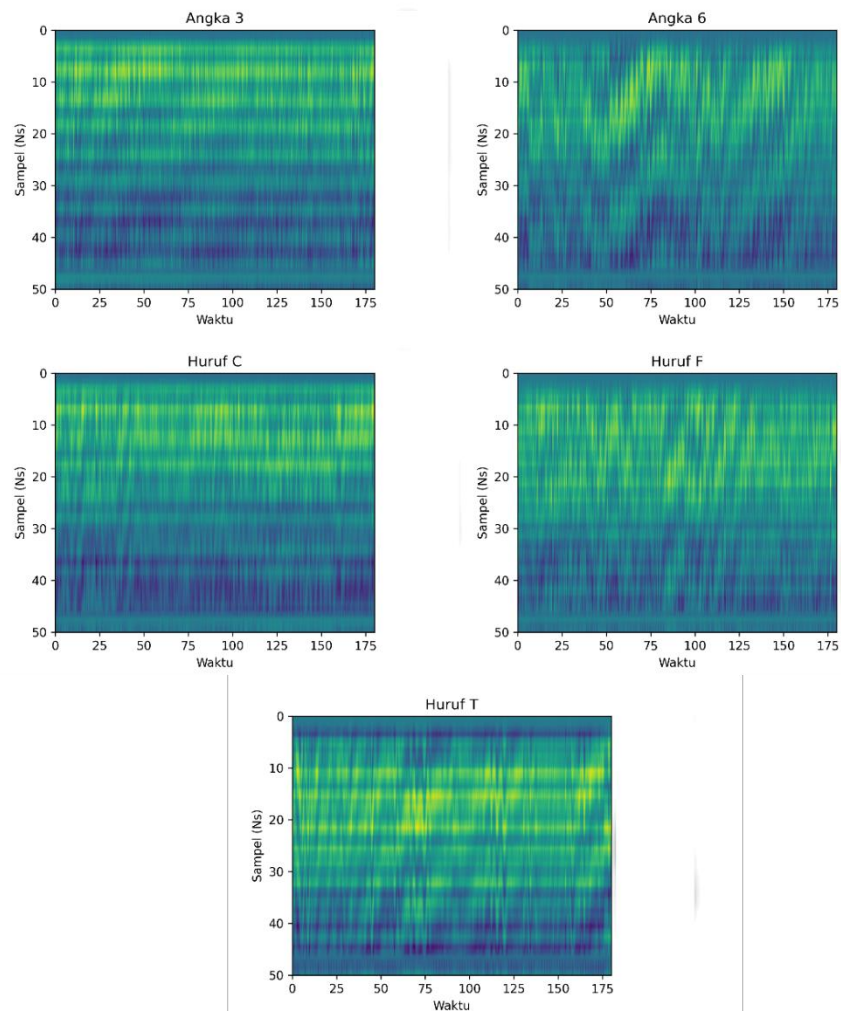
#### 4.3 Metode *Pre-processing* Data

Proses pengolahan data setelah *dataset* dikumpulkan terbagi menjadi beberapa metode *preprocessing* data sebelum masuk kedalam *training* dan *testing*, berikut metode *preprocessing* data yang dilakukan:

##### 1. Menghilangkan *DC Component*

*Raw dataset 5 class* yang dihasilkan oleh radar masih berbentuk dalam sinyal I (*in-phase*) Q (*Quadrature*). *Dataset 5 class* yang telah terkumpul

memiliki dimensi 1 data *class* (3600, 50) yang di mana artinya 1 *class dataset* memiliki 50 sampel dalam 1 data *frame* dan panjang waktu 180 detik dalam 1 data *class*. Penulis melakukan konversi bentuk sinyal I/Q yang terdapat dalam *dataset* agar data dapat diubah menjadi kedalaman tegangan (*Voltage*) dikarenakan *raw data* ini memiliki satuan *arbitrary units* dari 0 sampai 4095 [31]. Konversi sinyal tersebut menggunakan metode penghitungan *maximal Voltage* GPIO pada RaspberryPi 4 3,3 *Volt* dibagi dengan *ADC Intervals* dari uRAD. Komponen I dan Q memiliki *DC Component*, maka dari itu penghilangan *DC Component* sangat diperlukan, penghilangan *DC Component* dengan mengaplikasikan I dan Q dengan rumus 2.7.



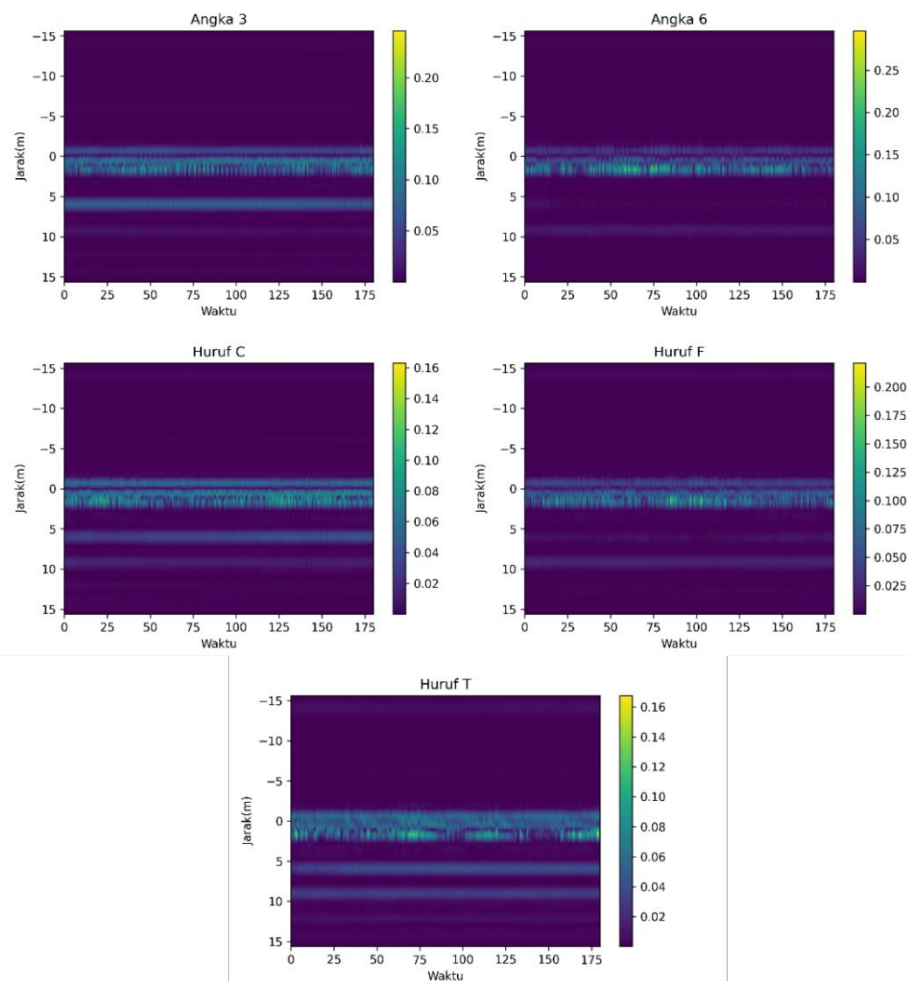
**Gambar 4.3** Bentuk Sinyal I dan Q dalam 2 dimensi

**Gambar 4.3** merupakan bentuk dari hasil *raw data* yang dapat digabung sehingga menjadi bentuk matriks, di mana Ns menandakan baris dan kolom

menandakan *frame* per *capture*. Dalam satu *frame* membutuhkan waktu 50 ms dalam 1 tembakan radar, penulis mengkonversikan menjadi satuan detik. Sinyal I dan Q dapat dibuat menjadi sinyal kompleks  $I + jQ$ , maka penulis dapat mengekstrak *magnitude* dari sinyal pantul [31].

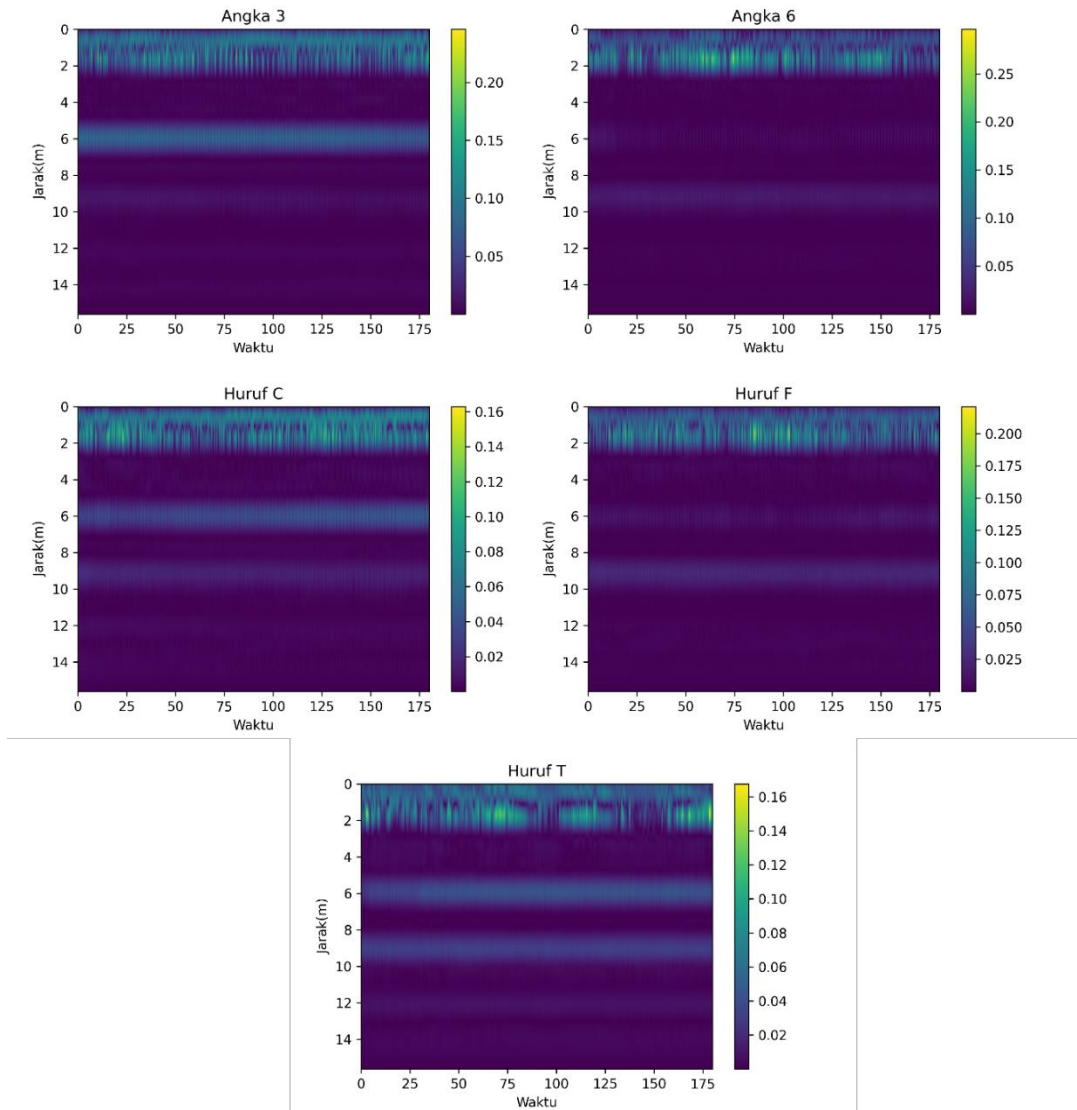
## 2. Fast Fourier Transform (FFT)

Jika melihat gambar keluaran sinyal I dan Q pada **Gambar 4.3** data tersebut masih memiliki bentuk pola yang belum jelas karena tidak terlihat objek pergerakan huruf alfabet dan angka SIBI serta tidak terlihat objek statik lainnya. Untuk dapat mengetahui informasi mengenai jarak dari radar ke objek pergerakan huruf alfabet dan angka SIBI, diperlukan metode *Fast Fourier Transform* [32]. Data sinyal I dan Q yang sudah dikonversikan ke dalam *Voltage* selanjutnya akan masuk ke dalam proses FFT dengan jumlah indeks frekuensi 4096.



**Gambar 4. 4** *Plot Spectrogram FFT*

Dalam **Gambar 4.4** tersebut terdapat dua frekuensi yang berbeda yaitu frekuensi positif dan negatif sehingga apabila menginginkan informasi mengenai jarak antara radar dengan target menggunakan indeks frekuensi dari rentang indeks frekuensi 2048 sampai 4096, karena rentang indeks frekuensi tersebut merupakan bagian frekuensi positif dan memiliki informasi mengenai jarak antara radar dengan target.



**Gambar 4. 5** *Plot Spectrogram Range Indeks Frekuensi 2048 Sampai 4096*

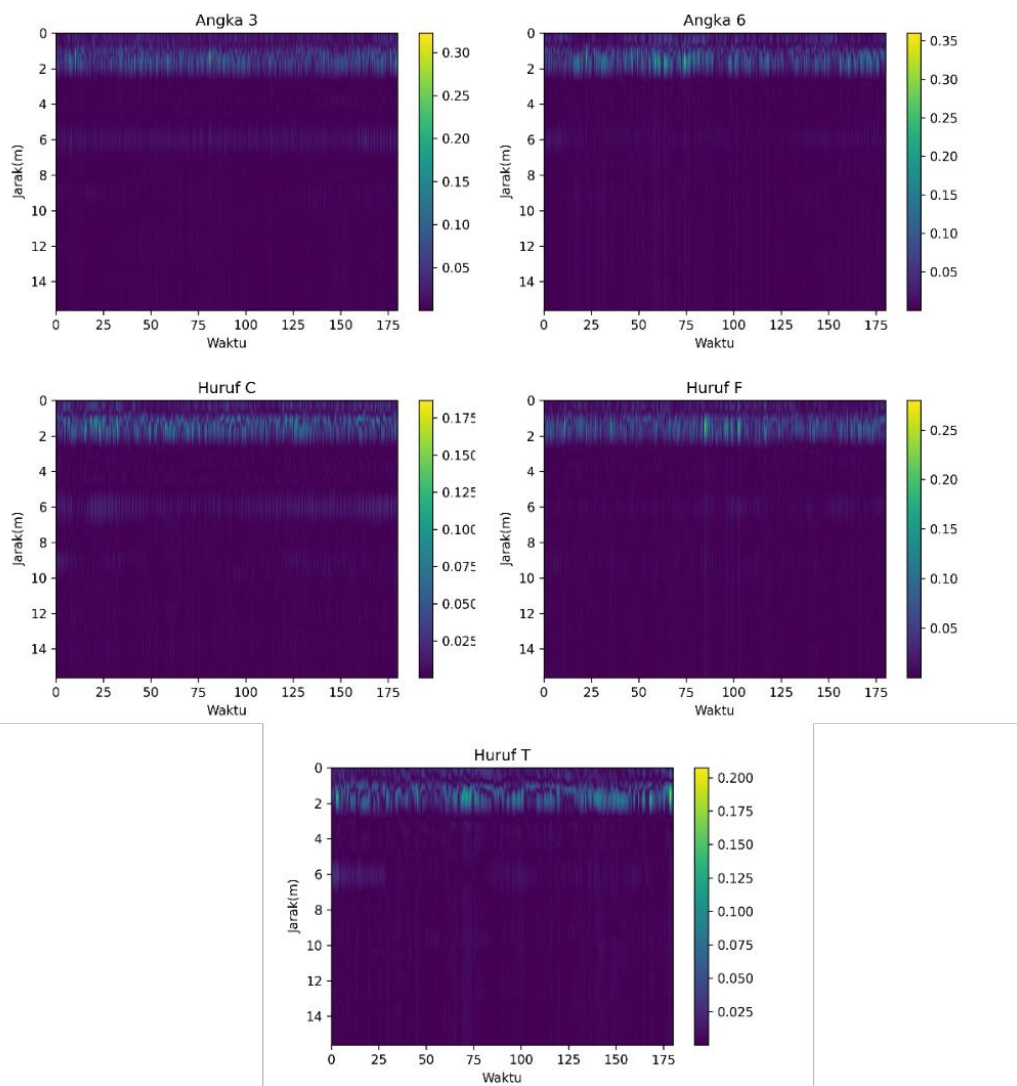
Bagian kolom (y) pada **Gambar 4.5** di atas merupakan indeks frekuensi, namun indeks frekuensi tersebut sudah dikonversikan kedalam jarak dengan jarak



maksimum yang dicapai uRAD 15,625 meter dengan perhitungan sesuai rumus 3.1. Bagian baris (x) pada gambar *spectrogram* diatas adalah waktu dalam detik.

### 3. Peredaman *Clutter* (*Clutter Suppression*)

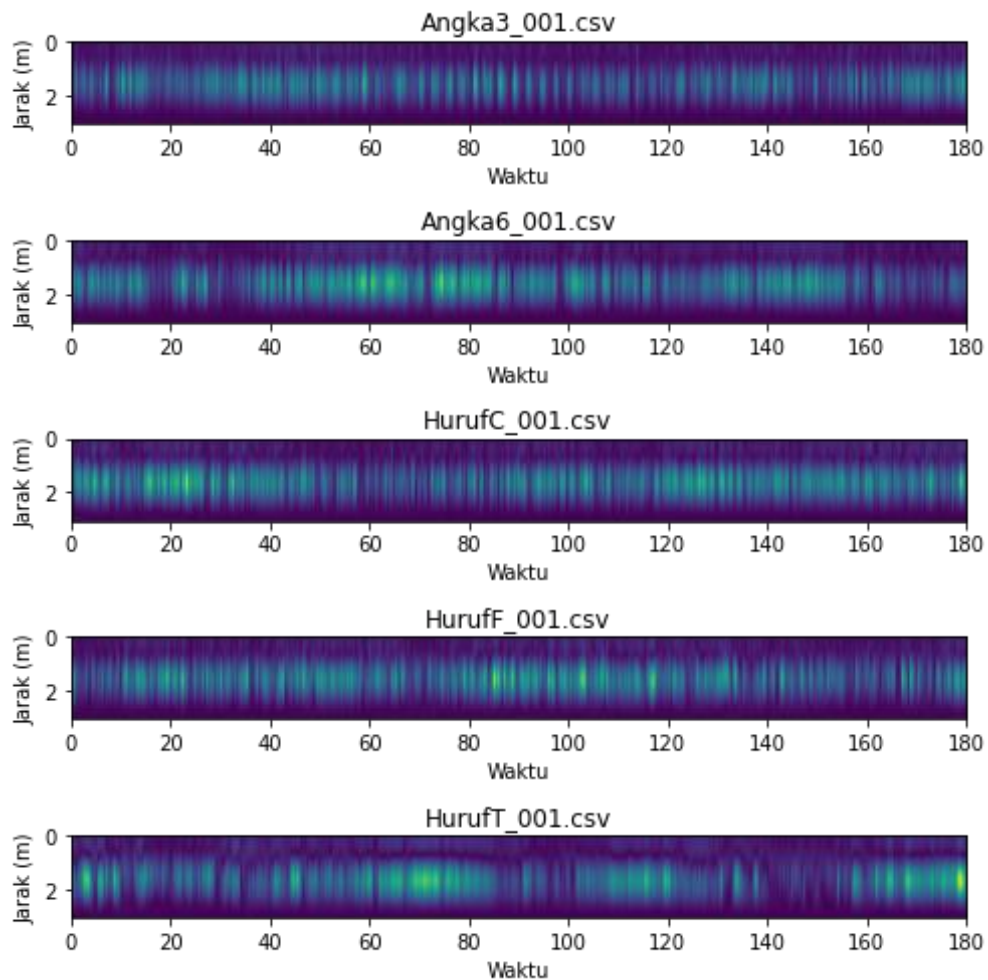
Data sinyal I dan Q yang telah melalui proses konversi ke dalam *Voltage* dan domain frekuensi masih terdapat objek statik (*clutter*) [20]. Sehingga diperlukan suatu metode untuk meredamkan objek statik tersebut agar pergerakan lebih jelas terlihat. Dalam peredaman gangguan objek statik tersebut dapat dihilangkan secara matematis yang dapat dituliskan pada persamaan 2.8. Sehingga hasil dari proses peredaman *clutter* dapat ditunjukkan pada **Gambar 4.6**.



**Gambar 4. 6** *Plot Spectrogram Hasil Peredaman Clutter*

#### 4. Metode *Cropping Spectrogram*

Sebelum *dataset* dimasukkan kedalam model *training*, penulis terlebih dahulu melakukan pemangkasan kembali yang di mana hanya memangkas target yang dideteksi uRAD radar seperti yang ditunjukkan pada **Gambar 4.7**. Pemangkasan ini bertujuan untuk membuang fitur dari data yang tidak terpakai untuk kedalam metode *Deep Learning Convolutional Neural Network*.

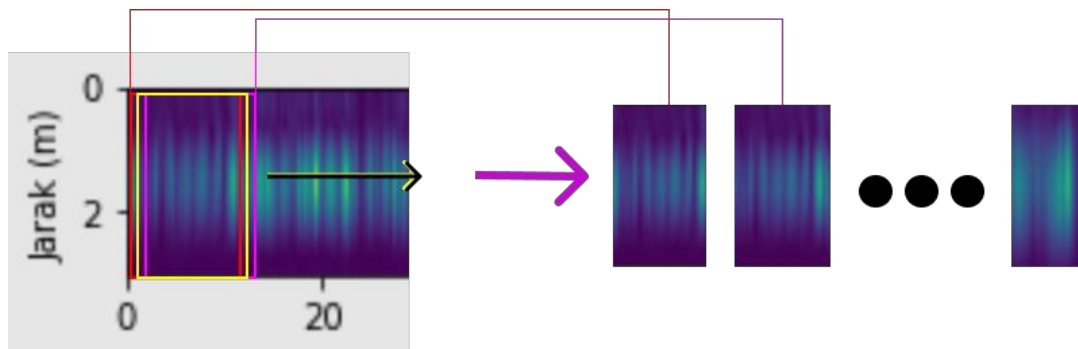


**Gambar 4. 7** *Plot Spectrogram* Hanya Target Terdeteksi

Setelah memangkas target yang dideteksi uRAD, selanjutnya data akan di *cropping* (pangkas) sebanyak 45 *frame* dengan membentuk matriks persegi panjang (400x45) mengikuti *time series* (waktu) atau dengan waktu 2.25 detik, *cropping* tersebut bertujuan untuk memperbanyak jumlah *dataset* sehingga tidak perlu mengambil data manual secara berkala. Matriks persegi panjang tersebut diekstrak setiap *frame* atau 50 milidetik, dari *cropping*



tersebut menghasilkan banyak matriks persegi yang panjang. Matriks persegi panjang akan dijadikan *dataset* untuk *deep learning*. Pada **Gambar 4.8** dapat ditunjukkan dari ilustrasi proses *cropping spectrogram*, *dataset* akhir yang telah melewati seluruh proses *preprocessing* maka didapatkan jumlah *dataset* yaitu sebanyak 3556 data setiap jenis *class* dan total data menjadi 17780 data.



**Gambar 4. 8** Ilustrasi Proses *Cropping* Per 2.25 Detik

#### 4.4 Perancangan arsitektur (model) *Deep Learning* CNN

Pengujian arsitektur dalam mencari model terbaik diperlukan perancangan arsitektur CNN yang ditentukan dari penggunaan beberapa parameter. Parameter yang diperlukan dalam pengujian adalah penentuan jumlah *convolutional layer*, *pooling layer* dan jumlah *epoch*. Perancangan arsitektur berdasarkan parameter tersebut dapat ditentukan model terbaik untuk klasifikasi.

*Convolutional layer* merupakan terdiri dari *neuron* yang diatur untuk membentuk suatu filter yang memiliki panjang dan tinggi (dalam *pixels*). *Epoch* merupakan waktu yang dibutuhkan seluruh *dataset* untuk melalui proses *training* pada *neural network* untuk satu kali putaran. Perancangan arsitektur terdiri dari 3 skenario pengujian yang di mana terdiri dari 4 *layer*, 5 *layer*, dan 6 *layer* konvolusi dan *pooling layer* pada setiap jumlah *layer* yang dipakai.

**Tabel 4. 2** Pengaruh Jumlah *Layer* Konvolusi, *Max Pooling Layer*, dan Jumlah *Epoch*

<b>Jumlah Layer Konvolusi</b>	<b><i>Pooling Layer</i></b>	<b><i>Epoch</i></b>	<b><i>Validation Accuracy</i></b>	<b><i>Validation Categorical Crossentropy Loss</i></b>
4	<i>Max Pooling</i>	30	97,59%	6,24%
		50	99,35%	2,3%
5	<i>Max Pooling</i>	30	96,15%	9,88%
		50	96,20%	11,13%
6	<i>Max Pooling</i>	30	99,19%	2,42%
		50	99,91%	0,36%

**Tabel 4. 3** Pengaruh Jumlah *Layer* Konvolusi, *Average  
Pooling Layer*, dan Jumlah *Epoch*

<b>Jumlah Layer Konvolusi</b>	<b><i>Pooling Layer</i></b>	<b><i>Epoch</i></b>	<b><i>Validation Accuracy</i></b>	<b><i>Validation Categorical Crossentropy Loss</i></b>
4	<i>Average Pooling</i>	30	95,88%	11,58%
		50	99,73%	1,53%
5	<i>Average Pooling</i>	30	96,45%	10%
		50	99,26%	2,16%
6	<i>Average Pooling</i>	30	96,31%	9,37%
		50	99,98%	0,37%

#### 4.5 Pengujian hasil *training* dan hasil *testing* arsitektur (model)

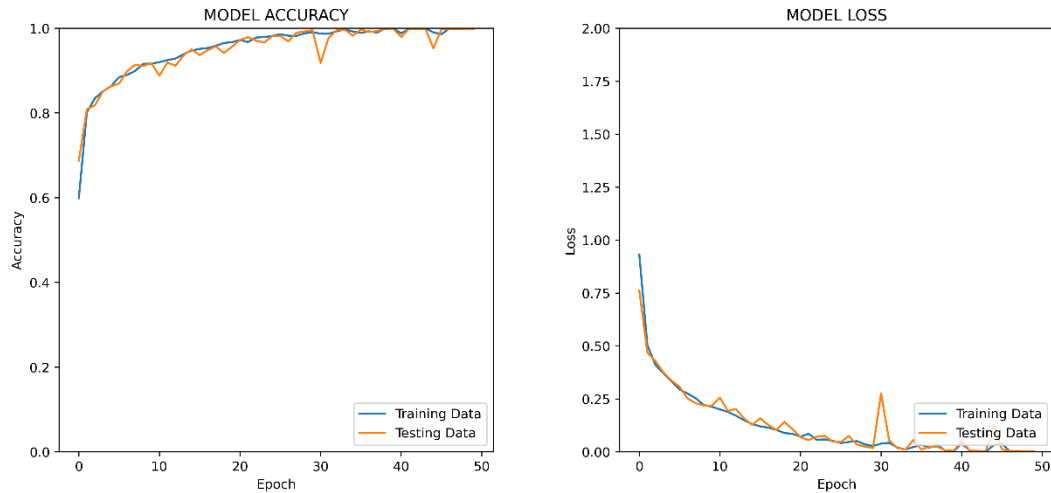
Apabila melihat **Tabel 4.3**, proses *training* dilakukan dengan menggunakan 4 sampai 6 *layer* konvolusi, *max pooling layer*, dan 30 & 50 *epoch* sebagai perbandingan parameter pengujian. Dalam pengujian tersebut digunakan data *training* sebanyak 13335 gerakan, dan data *test* sebanyak 4445 gerakan. Pada setiap jumlah *layer* yang masuk kedalam proses *training* masing-masing menggunakan *batchsize* yang sama yaitu sebesar 32 *batchsize*. Artinya, penyabaran data pada proses *training* akan semakin banyak. Berdasarkan pengujian jumlah *epoch* pada setiap *layer* konvolusi dan *max pooling layer* yang masuk kedalam proses *training*, masing-masing memiliki *validation accuracy* yang berbeda. Parameter pengujian jumlah *epoch*, hasil yang memiliki tingkat akurasi terbaik pada penggunaan 6 *layer* konvolusi yang di mana menghasilkan akurasi sebesar 99,91% pada 50 *epoch*. Jumlah *layer* dan *epoch* dapat memengaruhi akurasi pada proses *training*, semakin banyak *layer* yang digunakan, akan mempengaruhi proses *training* dan parameter model. Semakin banyak jumlah *epoch* yang digunakan maka semakin baik akurasinya namun tidak menutup kemungkinan bahwa proses *training* akan menjadi *overfitting* atau akurasi menurun.

Pengujian berikutnya hampir sama seperti pengujian sebelumnya yang di mana tetap menggunakan jumlah 4 sampai 6 *layer* konvolusi, dan 30 & 50 jumlah *epoch*. Pengujian berikutnya penulis menambahkan parameter pengujian yaitu dengan menggunakan *average pooling layer*. Pengujian menggunakan *layer average pooling* berdasarkan hasil **Tabel 4.3** jumlah konvolusi *layer* dan *epoch* juga dapat mempengaruhi hasil proses *training*. Pengujian menghasilkan bahwa akurasi dan *loss* terbaik dihasilkan oleh jumlah 6 *layer* konvolusi. *Layer* tersebut menghasilkan akurasi sebesar 99,98% dan akurasi *loss* 0,37% dengan *epoch* 50. Berdasarkan hasil penggunaan *pooling layer* yang ditunjukkan pada **Tabel 4.2** dan **Tabel 4.3** dapat disimpulkan bahwa memiliki hasil akurasi yang terbaik dari model yang terbaik. Namun analisis harus dilakukan pada pengujian 2 *pooling layer* yang berbeda untuk mendapatkan model terbaik.

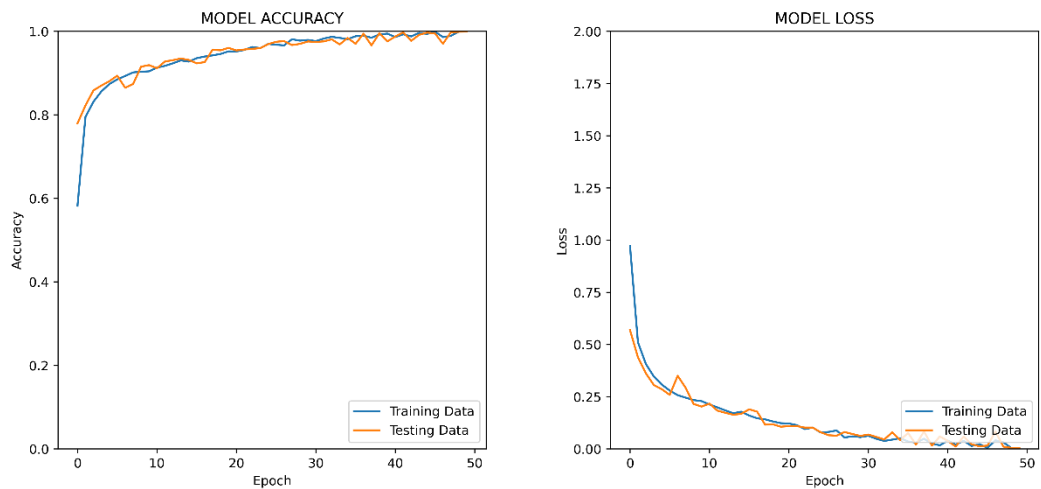
Pengujian pertama menggunakan *max pooling layer* dapat disimpulkan bahwa akurasi terbaik dihasilkan oleh penggunaan 6 *layer* konvolusi dengan menghasilkan akurasi sebesar 99,91% dan akurasi *loss* 0,36% pada 50 *epoch*. Pengujian kedua menggunakan parameter *average pooling layer* dapat dilihat akurasi terbaik dihasilkan oleh jumlah 6 konvolusi *layer*, *layer* tersebut menghasilkan akurasi sebesar 99,98% dan akurasi *loss* 0,37% pada *epoch* 50. Terdapat tiga model terbaik pada pengujian pertama dengan menggunakan *max pooling layer* yaitu terdapat pada jumlah 4, dan 6 konvolusi *layer*, serta jumlah *epoch* 30 pada *layer* 6 dan 50 pada jumlah *layer* 4 dan 6. Kedua *layer* tersebut mendapatkan akurasi yang sangat baik yaitu sebesar 99% namun terjadi penurunan akurasi sebesar 0,16% pada *layer* ke 4 pada jumlah *epoch* 50 dan 6 *layer* pada jumlah *epoch* 30, dan terjadi peningkatan akurasi *loss* sebesar 0,12%.

Jumlah kedua *layer* tersebut ketika menggunakan 6 *layer* konvolusi dengan jumlah *epoch* 50 terjadi peningkatan akurasi sebesar 0,72% dan penurunan akurasi *loss* sebesar 2,06%. Model yang dihasilkan dengan jumlah 6 *layer* konvolusi dengan menggunakan jumlah 50 *epoch* dapat disimpulkan lebih unggul dibandingkan dengan menggunakan jumlah 4 *layer* dan 6 *layer* jumlah *epoch* 30. Pengujian menggunakan *average pooling layer* terdapat tiga model terbaik. Ketiga model terbaik tersebut terdapat pada jumlah 4,5, dan 6 *layer* konvolusi, masing-masing menggunakan jumlah *epoch* yang sama yaitu 50. Terdapat penurunan akurasi sebesar 0,47% dan peningkatan akurasi *loss* sebesar 0,63% pada jumlah *layer* 4 dan 5. Namun, ketika menggunakan 6 *layer* konvolusi terjadi peningkatan akurasi sebesar 0,72% dan penurunan akurasi *loss* sebesar 1,79%. Artinya, bahwa penggunaan 6 *layer* konvolusi dalam pengujian *average pooling layer* dengan jumlah *epoch* sebesar 50 lebih unggul daripada kedua *layer* konvolusi yaitu 4 dan 5 *layer* konvolusi jumlah *epoch* yang sama yaitu 50. Hasil analisis kedua pengujian

*pooling layer* tersebut dihasilkan *plot* dari nilai *accuracy* dan *loss* menggunakan dataset *training* dan *testing* pada jumlah 6 *layer* konvolusi dengan jumlah *epoch* 50.

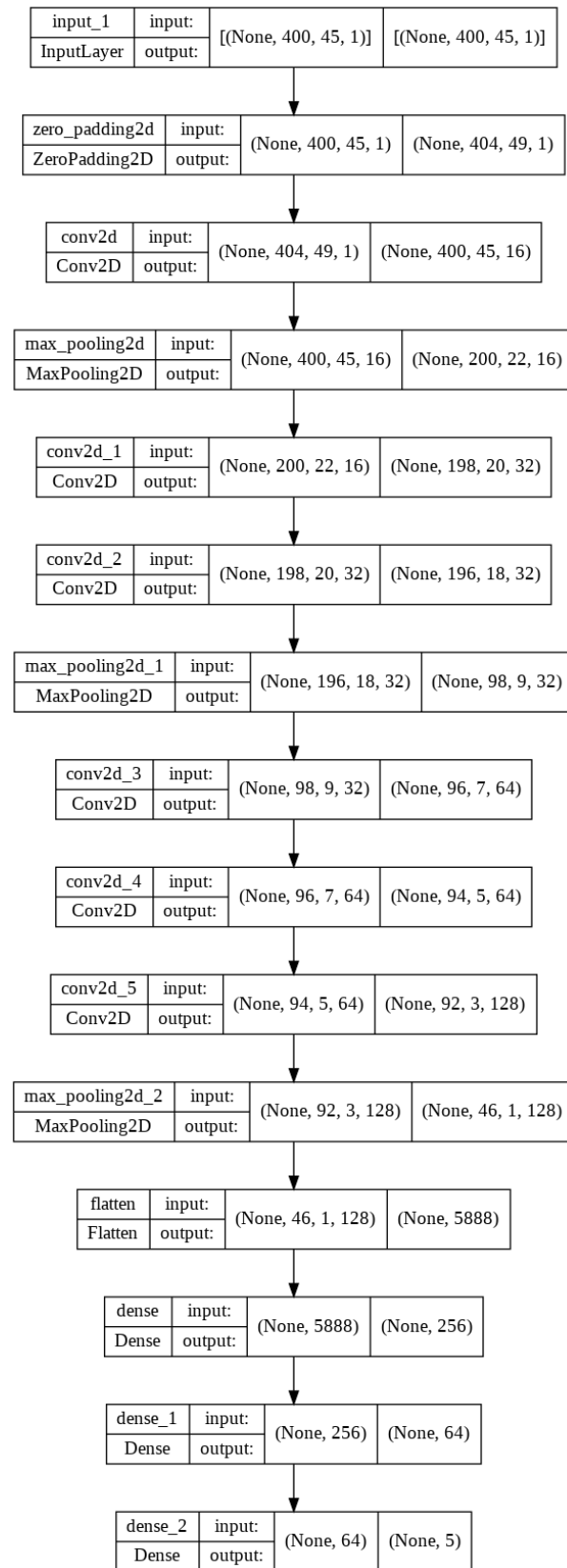


**Gambar 4. 9** *Plot Accuracy dan Loss Max Pooling Layer*

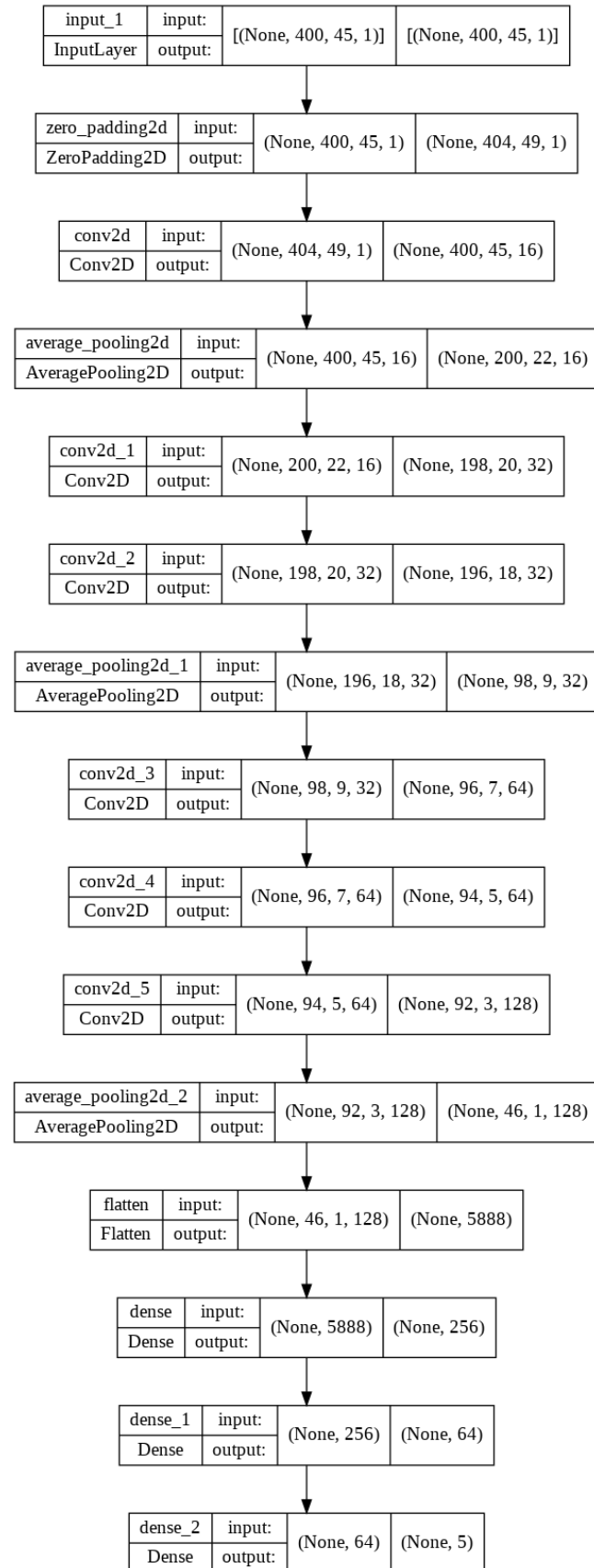


**Gambar 4. 10** *Plot Accuracy dan Loss Average Pooling Layer*

**Gambar 4.9** dan **Gambar 4.10** merupakan *plot* grafik dari hasil *epoch* menggunakan dataset *training* dan *testing*. Akurasi menggunakan data *training* akan digunakan dalam melatih algoritma ketika mencari model yang sesuai. Data *testing* dipakai dalam menguji performansi model yang telah didapatkan dalam tahap *testing*. Selain *plot* nilai *accuracy* dan *loss* dari parameter pengujian tersebut, perancangan arsitektur CNN juga ditampilkan pada **Gambar 4.11** dan **Gambar 4.12**.

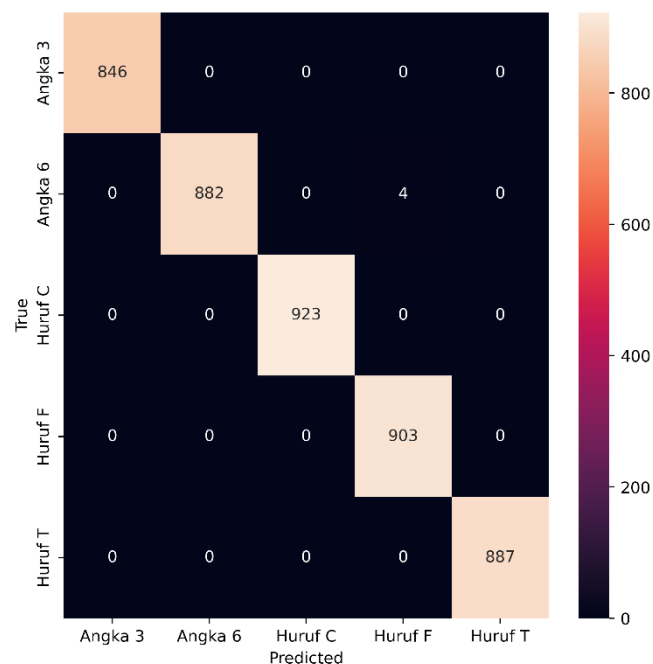


**Gambar 4. 11** Arsitektur CNN 6 Layer Konvolusi Max Pooling Layer



**Gambar 4. 12** Arsitektur CNN 6 Layer Konvolusi *Average Pooling Layer*

*Feature Learning* dan *Classification* merupakan dua tahapan dalam pembentukan arsitektur CNN. *Feature Learning* adalah seperangkat teknik yang memungkinkan sistem untuk secara otomatis menemukan representasi yang diperlukan dalam klasifikasi. Setelah data masuk dalam *feature learning* selanjutnya akan masuk dalam tahapan klasifikasi berdasarkan setiap kelas yang sudah ditentukan. Pengujian menggunakan parameter jumlah *layer* konvolusi, *pooling layer*, dan jumlah *epoch* dihasilkan analisis model yang terbaik pada **Tabel 4.2** dan **Tabel 4.3**. Dalam evaluasi sebuah performansi dari sebuah model untuk melakukan klasifikasi, dapat dilihat melalui *confusion matrix* dan *classification report*. Hasil dari *confusion matrix* dan *classification report* yang digunakan dalam pengujian berdasarkan *max pooling layer* dan *average pooling layer* ditunjukkan sesuai dengan hasil dari analisis dalam mendapatkan model terbaik. Berikut gambar dari *confusion matrix* dalam menghitung faktor-faktor dalam performansi klasifikasi.

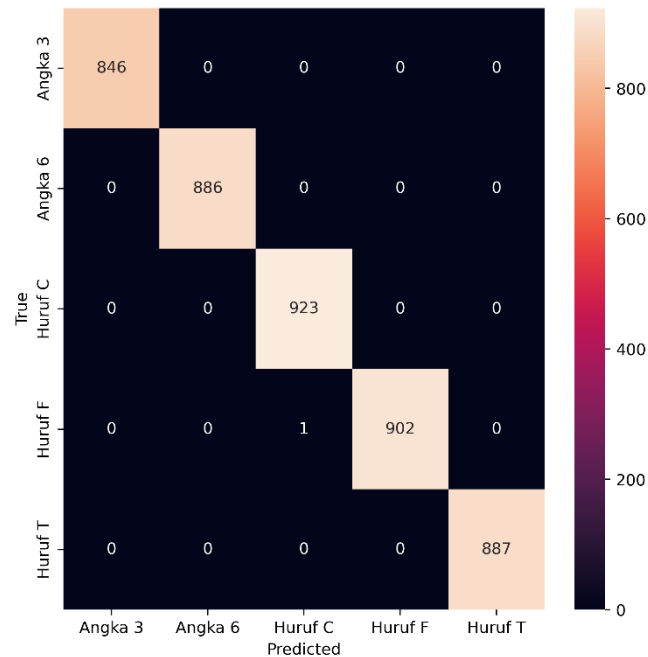


**Gambar 4. 13** *Confusion Matrix Max Pooling Layer*

Penggunaan metode CNN pada sistem ini menghasilkan *confusion matrix max pooling* dapat ditunjukkan pada **Gambar 4.13**. Berdasarkan hasil *confusion matrix*



tersebut didapatkan hasil prediksi yang baik dengan menggunakan data *test*. Pada *class* “Angka 3” dapat diprediksi secara benar diseluruh data *test* sebanyak 846 data, *class* “Angka 6” dapat diprediksi benar sebanyak 882 data namun terdapat 4 kali *miss* klasifikasi. *Class* “Huruf C” diprediksi benar sebagai huruf c sebanyak 923 data dan tanpa terjadi *miss* klasifikasi. *Class* “Huruf F” terprediksi benar sebanyak 903 data dan *class* “Huruf T” dapat diprediksi benar sebanyak 887 data.



**Gambar 4. 14** *Confusion Matrix Average Pooling Layer*

Selain hasil dari *confusion matrix* pada penggunaan *max pooling layer*, terdapat hasil dari *confusion matrix* pada penggunaan *average pooling layer* dapat ditunjukkan pada **Gambar 4.14**. *Class* “Angka 3” dapat diprediksi secara benar diseluruh data *test* sebanyak 846 data, *class* “Angka 6” dapat diprediksi benar sebanyak 886 data tanpa terjadi *miss* klasifikasi. *Class* “Huruf C” diprediksi benar sebagai huruf c sebanyak 923 data dan tanpa terjadi *miss* klasifikasi. *Class* “Huruf F” diprediksi benar sebanyak 902 data namun terjadi *miss* klasifikasi sebanyak 1 kali, dan *class* “Huruf T” dapat diprediksi benar sebanyak 887 data.

**Tabel 4. 4** Tabel *Classification Report Max Pooling Layer*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Angka 3	1.00	1.00	1.00	846
Angka 6	1.00	1.00	1.00	886
Huruf C	1.00	1.00	1.00	923
Huruf F	1.00	1.00	1.00	903
Huruf T	1.00	1.00	1.00	887
<i>Accuracy</i>			1.00	4445

**Tabel 4. 5** Tabel *Classification Report Average Pooling Layer*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Angka 3	1.00	1.00	1.00	846
Angka 6	1.00	1.00	1.00	886
Huruf C	1.00	1.00	1.00	923
Huruf F	1.00	1.00	1.00	903
Huruf T	1.00	1.00	1.00	887
<i>Accuracy</i>			1.00	4445

*Classification Report* merupakan sebuah laporan yang digunakan ketika mengukur kualitas prediksi dari algoritma klasifikasi, seperti berapa banyak prediksi yang benar dan berapa banyak yang salah. *Precision* disini adalah hasil perbandingan antara nilai *True Positive* (TP) dengan banyak data yang telah diprediksi positif. *Recall* merupakan suatu perbandingan antara *True Positive* (TP) dengan banyak data yang sebenarnya positif. *F1-score* merupakan nilai *harmonic* rata-rata dari

*precision* dan *recall*. Tidak hanya digambarkan secara *confusion matrix*, perhitungan keseluruhan akurasi dari matriks penggunaan *max pooling layer* tersebut dapat dihitung dengan menggunakan persamaan 4.1.

$$\begin{aligned} \text{Overall Accuracy} &= \frac{TP \text{ All}}{\text{Total Number Of Value Testing Entries}} \times 100\% \\ &= \frac{4441}{4445} \times 100\% = 99,91\% \end{aligned} \quad (4.1)$$

Perhitungan keseluruhan akurasi dari matriks penggunaan *average pooling layer* dapat dihitung dengan persamaan 4.2.

$$\begin{aligned} \text{Overall Accuracy} &= \frac{TP \text{ All}}{\text{Total Number Of Value Testing Entries}} \times 100\% \\ &= \frac{4444}{4445} \times 100\% = 99,97\% \end{aligned} \quad (4.2)$$

Didapatkan hasil akurasi dari pemodelan dengan metode CNN menggunakan *input* data (400,40), dan jumlah data *test* sebanyak 4445 data didapatkan nilai akurasi pada penggunaan *max pooling layer* sebesar 99,91% dan hasil akurasi dari penggunaan *average pooling layer* sebesar 99,97%. Hasil akurasi yang didapatkan pemodelan terbaik dihasilkan pada pengujian menggunakan parameter 6 *layer* konvolusi, jumlah *epoch* 50 dan menggunakan *average pooling layer* mendapatkan keseluruhan akurasi sebesar 99,97%.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengamatan yang mengacu pada perancangan sistem, metode *preprocessing*, perancangan model (arsitektur) dan pengujian model dengan menggunakan metode *Convolutional Neural Network*, dapat disimpulkan sebagai berikut:

1. *Dataset* yang telah dikumpulkan akan memasuki tahap metode *preprocessing* antara lain menggunakan metode penghilangan *dc component*, *fast fourier transform* dalam mengubah kedalam domain frekuensi, metode peredaman *clutter* dalam meredam objek statik yang terdeteksi, dan metode *cropping spectrogram* dalam memperbanyak dataset serta membuang fitur yang tidak dipakai dalam CNN, sehingga total keseluruhan data sebesar 17780 data *class* dengan matriks 400 x 45.
2. Berdasarkan pengujian arsitektur CNN dengan parameter jumlah *layer* konvolusi, jumlah *epoch*, dan *pooling layer*. Akurasi yang dicapai pada pengujian dua *pooling layer* dengan menggunakan *max pooling* dan *average pooling*, mendapatkan model terbaik dengan jumlah 6 *layer* konvolusi, penggunaan *average pooling layer*, dan *epoch* 50. Model tersebut mendapatkan akurasi validasi sebesar 99,98% serta keseluruhan akurasi sebesar 99,97%. Model tersebut dapat memberikan hasil klasifikasi dan prediksi yang baik sehingga mampu mengklasifikasikan huruf albet dan angka seperti Huruf C, Huruf F, Huruf T, Angka 3, dan Angka 6.

#### **5.2 Saran**

Berdasarkan hasil dari perancangan sistem dan model (arsitektur), pengujian dan analisis dari tugas akhir ini, penulis ingin memberikan saran yaitu sebagai berikut:

1. Pengembangan lebih lanjut diperlukan pada penelitian selanjutnya mengenai implementasi sistem secara *real time* maupun sejenisnya dengan dapat mendeteksi dari berbagai jarak agar sistem pengenalan gerakan bahasa isyarat

SIBI khususnya huruf alfabet dan angka dapat diaplikasikan langsung kepada penyandang disabilitas (tunarungu dan tunawicara).

2. Pada penelitian selanjutnya diharapkan dapat merekam lebih banyak *class* gerakan huruf alfabet dan angka SIBI agar semua huruf dan angka tersebut dapat diimplementasikan untuk pengolahan data, klasifikasi, dan maupun untuk sistem *real time*.

## DAFTAR PUSTAKA

- [1] D. R. Kurnia and T. Slamet, "Menormalkan yang Dianggap 'Tidak Normal' (Studi Kasus Penertiban Bahasa Isyarat Tunarungu di Sekolah Luar Biasa [SLB] dan Perlawanannya di Kota Malang)," *Ijds*, vol. 3, no. 1, pp. 34–43, 2016.
- [2] Y. Bukesra Ulee Kareng and B. Aceh, "Jurnal Ilmiah Mahasiswa FISIP Penggunaan Sistem Isyarat Bahasa Indonesia (SIBI) sebagai media komunikasi (studi pada siswa tunarungu di SLB "PENGUNAAN SISTEM ISYARAT BAHASA INDONESIA (SIBI) SEBAGAI MEDIA KOMUNIKASI (studi pada siswa tunar)," *J. Ilm. Mhs. FISIP Unsyiah*, vol. 4, no. 3, pp. 1–12, 2019, [Online]. Available: [www.jim.unsyiah.ac.id/FISIP](http://www.jim.unsyiah.ac.id/FISIP)
- [3] A. Basuki, M. Zikky, J. Akhmad, N. Hasim, and N. I. Ramadhan, "Sensor Gerak dengan Leap Motion untuk Membantu Komunikasi Tunarungu/Wicara," *Pros. Sentia*, vol. 8, no. October, pp. 317–321, 2016.
- [4] L. Yusnita, R. Roestam, and R. B. Wahyu, "Implementation of Real-Time Static Hand," *CommIT (Communication Inf. Technol.*, vol. 11, no. 2, pp. 85–91, 2017, doi: Implementation of Real-Time Static Hand Yusnita LRoestam RWahyu R.
- [5] Q. Zhu and Y. Wang, "FMCW radar implemented with GNU Radio Companion," pp. 1–20, 2016.
- [6] H. R. Lee, J. Park, and Y. J. Suh, "Improving classification accuracy of hand gesture recognition based on 60 GHz FMCW radar with deep learning domain adaptation," *Electron.*, vol. 9, no. 12, pp. 1–24, 2020, doi: 10.3390/electronics9122140.
- [7] H. Kulhandjian, P. Sharma, M. Kulhandjian, and C. D'Amours, "Sign language gesture recognition using doppler radar and deep learning," *2019 IEEE Globecom Work. GC Wkshps 2019 - Proc.*, no. December, 2019, doi: 10.1109/GCWkshps45667.2019.9024607.

- [8] F. Gustomo, “Analisa Penggunaan Sinyal Radar Bentuk Pulsa dan Gelombang Kontinyu untuk Target Bergerak dengan Model Clutter Terdistribusi Rayleigh,” *J. Tek. POMITS Vol. 2, No. 2, ISSN 2337-3539 (2301-9271)*, vol. 2, no. 2, pp. 3–8, 2013.
- [9] M. A. Richards, *Fundamentals of Radar Signal Processing, Second Edition (2nd ed.)*, vol. 58, no. 12. 2014. [Online]. Available: [https://www.cambridge.org/core/product/identifier/CBO9781107415324A009/type/book\\_part%0Ahttp://www.ncbi.nlm.nih.gov/pubmed/25246403%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4249520](https://www.cambridge.org/core/product/identifier/CBO9781107415324A009/type/book_part%0Ahttp://www.ncbi.nlm.nih.gov/pubmed/25246403%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4249520)
- [10] P. Martelletti, *Editor-in-Chief's 2006 report*, vol. 7, no. 1. 2006. doi: 10.1007/s10194-006-0268-4.
- [11] S. Pal, *FMCW – Radar Design*, vol. 65, no. 4. 2019. doi: 10.1080/03772063.2019.1615392.
- [12] O. N. Samijayani, S. Rahmatia, V. N. Septiyani, and I. Ibrahim, “Perancangan Software Defined Radar Untuk Radar Pulsa dan Radar FMCW,” *J. Al-AZHAR Indones. SERI SAINS DAN Teknol.*, vol. 3, no. 3, p. 144, 2017, doi: 10.36722/sst.v3i3.220.
- [13] S. M. Lawalata, E. Ali, S. Prodi, T. Telekomunikasi, F. T. Elektro, and U. Telkom, “ANALISIS PENGGUNAAN RADAR FMCW DALAM MENDETEKSI GESTURE TANGAN MENGGUNAKAN SDR ANALYSIS OF HAND GESTURE DETECTION IN FMCW RADAR USING SDR,” vol. 7, no. 2, pp. 3323–3330, 2020.
- [14] H. Pratiwi, M. R. Hidayat, A. A. Pramudita, and F. Y. Suratman, “Improved FMCW Radar System for Multi-Target Detection of Human Respiration Vital Sign,” *J. Elektron. dan Telekomun.*, vol. 19, no. 2, p. 38, 2019, doi: 10.14203/jet.v19.38-44.
- [15] F. Yasmin *et al.*, “SIMULASI SISTEM RADAR FMCW PENERIMA JAMAK UNTUK PENGHINDAR TABRAKAN PADA MOBIL TANPA PENGEMUDI SIMULATION OF MULTI RECEIVER FMCW RADAR

- FOR COLLISION AVOIDANCE ON AUTONOMOUS CAR,” vol. 7, no. 1, pp. 1050–1057, 2020.
- [16] Y. Sun, T. Fei, X. Li, A. Warnecke, E. Warsitz, and N. Pohl, “Real-Time Radar-Based Gesture Detection and Recognition Built in an Edge-Computing Platform,” *IEEE Sens. J.*, vol. 20, no. 18, pp. 10706–10716, 2020, doi: 10.1109/JSEN.2020.2994292.
  - [17] D. T. Kusuma, “Fast Fourier Transform (FFT) Dalam Transformasi Sinyal Frekuensi Suara Sebagai Upaya Perolehan Average Energy (AE) Musik,” *Petir*, vol. 14, no. 1, pp. 28–35, 2020, doi: 10.33322/petir.v14i1.1022.
  - [18] A. S. Irtawaty, “Implementasi Metode Fast Fourier Transform (Fft) Dalam Mengklasifikasikan Suara Pria Dan Wanita Di Laboratorium Jurusan Teknik Elektro Politeknik Negeri Balikpapan,” *JTT (Jurnal Teknol. Terpadu)*, vol. 7, no. 2, pp. 70–75, 2019, doi: 10.32487/jtt.v7i2.661.
  - [19] M. Jahn, C. Wagner, and A. Stelzer, “DC-Offset Compensation Concept for Monostatic FMCW Radar Transceivers,” *undefined*, vol. 20, no. 9, pp. 525–527, Sep. 2010, doi: 10.1109/LMWC.2010.2056359.
  - [20] S. W. Kang, M. H. Jang, and S. Lee, “Identification of human motion using radar sensor in an indoor environment,” *Sensors*, vol. 21, no. 7. 2021. doi: 10.3390/s21072305.
  - [21] E. Pittella, B. Zanaj, S. Pisa, and M. Cavagnaro, “Measurement of Breath Frequency by Body-Worn UWB Radars: A Comparison among Different Signal Processing Techniques,” *IEEE Sens. J.*, vol. 17, no. 6, pp. 1772–1780, 2017, doi: 10.1109/JSEN.2017.2654538.
  - [22] X. Li, Y. He, and X. Jing, “A survey of deep learning-based human activity recognition in radar,” *Remote Sens.*, vol. 11, no. 9, 2019, doi: 10.3390/rs11091068.
  - [23] Y. N. Fu’adah, N. C. Pratiwi, M. A. Pramudito, and N. Ibrahim, “Convolutional Neural Network (CNN) for Automatic Skin Cancer



- Classification System,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 982, no. 1, 2020, doi: 10.1088/1757-899X/982/1/012005.
- [24] N. Ibrahim, Y. N. Fu’adah, N. C. Pratiwi, S. Rizal, and K. Usman, “Computer Aided System for Gambung Tea Identification using Convolutional Neural Network,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 982, no. 1, 2020, doi: 10.1088/1757-899X/982/1/012001.
- [25] H. J. Jie and P. Wanda, “Runpool: A dynamic pooling layer for convolution neural network,” *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 66–76, 2020, doi: 10.2991/ijcis.d.200120.002.
- [26] Y. N. Fu’Adah, I. Wijayanto, N. K. C. Pratiwi, F. F. Taliningsih, S. Rizal, and M. A. Pramudito, “Automated Classification of Alzheimer’s Disease Based on MRI Image Processing using Convolutional Neural Network (CNN) with AlexNet Architecture,” *J. Phys. Conf. Ser.*, vol. 1844, no. 1, 2021, doi: 10.1088/1742-6596/1844/1/012020.
- [27] H. S. Ibrahim, M. Si, U. N. Wisesty, F. Informatika, and U. Telkom, “Analisis Deep Learning Untuk Mengenali Qrs Kompleks Pada Sinyal Ecg Dengan Metode Cnn,” *e-Proceeding Eng.*, vol. 5, no. 2, pp. 3718–3725, 2018.
- [28] “Mengenal 6 Jenis Loss Function pada Machine Learning - Trivusi.” <https://www.trivusi.web.id/2022/08/loss-function.html> (accessed Aug. 30, 2022).
- [29] “Confusion Matrix untuk Evaluasi Model pada Supervised Learning | by Kuncahyo Setyo Nugroho | Medium.” <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f> (accessed Aug. 30, 2022).
- [30] “Membicarakan Precision, Recall, dan F1-Score | by Stevanus Setiawan | Medium.” <https://stevkarta.medium.com/membicarakan-precision-recall-dan-f1-score-e96d81910354> (accessed Aug. 30, 2022).

- [31] A. Analysis and U. Manual, “User Manual User Manual,” *Data Base*, vol. 3304, no. January, pp. 1–148, 2012.
- [32] S. Ahmed, J. Park, and S. H. Cho, “FMCW Radar Sensor Based Human Activity Recognition using Deep Learning,” *2022 Int. Conf. Electron. Information, Commun. ICEIC 2022*, 2022, doi: 10.1109/ICEIC54506.2022.9748776.

## LAMPIRAN

### 1. *Source Code Sistem*

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from numpy.fft import fft, fftshift
import numpy as np
import tensorflow as tf
from keras.models import Model
from keras.layers import Input, Activation, Dense, Conv2D, MaxPooling2D,
AveragePooling2D, ZeroPadding2D, Flatten
from tensorflow.keras.optimizers import Adam
from keras.utils.np_utils import to_categorical
from keras.callbacks import TensorBoard
from sklearn.model_selection import train_test_split
from sklearn.metrics._plot.confusion_matrix import plot_confusion_matrix
import sklearn.metrics as metrics
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay
import seaborn as sn
from google.colab import drive
drive.mount('/content/drive')

path = r"/content/drive/MyDrive/TA_ABI_180detik_MTIOFF"
fileNames = os.listdir(path)
fileNames.sort()
new_arr=[]
for i in fileNames:
    df = pd.read_csv(path + '/' + i)
    conv = df.to_numpy(complex)
    new_arr.append(conv)
    #return new_arr

def FFT (data_complex):
    max_voltage = 3.3
    ADC_intervals = 4096
    Ns = np.size(data_complex, 1)

    result = []
    for i in range(np.size(data_complex,0)):
        I = np.array(data_complex[i].real)
        Q = np.array(data_complex[i].imag)

#     I, Q = uRAD_processing.deleteBadSamples(I, Q)
```

```

    I = np.subtract(np.multiply(I, max_voltage/ADC_intervals),
np.mean(np.multiply(I, max_voltage/ADC_intervals)))
    Q = np.subtract(np.multiply(Q, max_voltage/ADC_intervals),
np.mean(np.multiply(Q, max_voltage/ADC_intervals)))

    ComplexVector = I + 1j*Q

    ComplexVector = ComplexVector * np.hanning(Ns) * 2 / 3.3

    N_FFT = 4096

    FrequencyDomainComplex = fftshift(fft(ComplexVector / Ns, N_FFT))
    result.append(FrequencyDomainComplex)

return result

def FFTlog(FDcomplex):
    ##      FrequencyDomain = 2*np.absolute(fftshift(fft(ComplexVector / Ns,
N_FFT)))
    N_FFT = 4096
    FDcomplex = np.array(FDcomplex)

    for i in range(np.size(FDcomplex, 0)):
        FDcomplex[i] = np.subtract(FDcomplex[i], np.mean(FDcomplex[i]))

    for i in range(np.size(FDcomplex, 1)):
        FDcomplex[:,i] = np.subtract(FDcomplex[:,i], np.mean(FDcomplex[:,i]))

    result = []
    for i in range(np.size(FDcomplex, 0)):
        FrequencyDomain = 2*np.absolute(FDcomplex[i])
        start = int(N_FFT/2)
        FrequencyDomain[start] = FrequencyDomain[start - 1]
    ##      FrequencyDomain = 20 * np.log10(FrequencyDomain)
    result.append(FrequencyDomain)

return result

def crop (membagi, n_crop):
    array_slicing=[]
    for i in range(len(membagi[0])-(n_crop-1)):
        array_slicing.append(membagi[:,i:i+n_crop])
    return array_slicing

def plot(data, rows, cols, sizefig, title):
    axes = []
    fig = plt.figure(figsize = sizefig)

```

```

if(rows*cols > np.size(data,0)):
    rowxcol = np.size(data,0)
else:
    rowxcol = rows*cols

for i in range(rowxcol):
    axes.append(fig.add_subplot(rows, cols, i+1))
    subplot_title = (title[i])
    axes[-1].set_title(subplot_title)
    plt.imshow(data[i],aspect = "auto")

fig.tight_layout()
plt.show()

new_data=[]
for i in range(len(new_arr)):
    data_fft = FFT(new_arr[i])
    data_fft_2 = FFTlog(data_fft)
    transpose_data = np.transpose(data_fft_2)
    new_data.append(transpose_data)

print(np.shape(new_data))

after_fft = np.array(new_data)
crop_empatratu = after_fft[:,2048:2048+400,:]
print(np.shape(crop_empatratu))

data_cropping = []
for i in range (len(crop_empatratu)):
    data_cropping.append(crop(crop_empatratu[i],45))

print(np.shape(data_cropping))
# print(data_crop)
# print(dataset)

X =
np.reshape(data_cropping,((np.size(data_cropping,0)*np.size(data_cropping,1
)),np.size(data_cropping,2),np.size(data_cropping,3)))
print(np.shape(X))

Y = []
for i in range(5):
    Y.extend(i for j in range(int(np.size(X, 0)/5)))
print(Y)
print(np.shape(Y))

# Y = to_categorical(Y)
# print(np.shape(Y))

```

```
# print(Y)

train_x, test_x, train_y, test_y = train_test_split(X, Y, test_size=0.25,
random_state=0)
print(np.shape(train_x))
print(np.shape(test_x))
print(np.shape(train_y))
print(np.shape(test_y))

train_x = np.reshape(train_x, (np.size(train_x, 0), np.size(train_x, 1),
np.size(train_x, 2),1))
test_x = np.reshape(test_x, (np.size(test_x, 0), np.size(test_x, 1),
np.size(test_x, 2),1))

print(np.shape(train_x))
print(np.shape(test_x))
```

## 2. Gambar Alat

Berikut ini merupakan gambar uRAD radar yang dipakai dalam penelitian tugas akhir ini:



## 3. Dokumentasi Pengambilan Dataset

Berikut ini merupakan dokumentasi dalam pengambilan dataset gerakan huruf alfabet dan angka SIBI:





(a) Huruf C



(b) Huruf F



(c) Huruf T



(d) Angka 3



(e) Angka 6