



**SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA  
INDONESIA) MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

**TUGAS AKHIR**



**Program Studi  
S1 TEKNIK KOMPUTER**

**Oleh:**

**ILHAM RIZALDY WIDY PUTRA**

**17410200032**

---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2021**

**SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA  
INDONESIA) MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK***

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Teknik**



UNIVERSITAS  
**Dinamika**

**Disusun Oleh:**

**Nama : Ilham Rizaldy Widy Putra**  
**NIM : 17410200032**  
**Program Studi : S1 Teknik Komputer**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA  
2021**

## TUGAS AKHIR

### SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA INDONESIA) MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

Dipersiapkan dan disusun oleh:

**Ilham Rizaldy Widy Putra**

**NIM : 17410200032**

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 15 Juli 2021

#### Susunan Dewan Pembahas

##### Pembimbing:

I. Heri Pratikno, M.T., MTCNA., MTCRE.

NIDN: 0716117302

II. Musayyanah, S.ST., M.T.

NIDN: 0730069102

##### Pembahas:

Dr. Jusak

NIDN: 0708017101

  
Dr. Heri Pratikno, M.T.,  
Universitas Dinamika,  
Departemen Teknik Komputer,  
email: heripr@dinamika.ac.id,  
CUIID  
Date: 2021.07.29 15:01:01  
+07'00'

  
Digitally signed by  
Musayyanah, S.ST., M.T.,  
Universitas Dinamika,  
email: musayyanah@dinamika.ac.id,  
Date: 2021.07.29 15:01:01  
+07'00'

  
Digitally signed by  
Universitas  
Dinamika  
Date: 2021.07.30  
14:38:44 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana

Digitally signed by  
Universitas Dinamika  
Date: 2021.08.02  
07:23:16 +07'00'

**Tri Sagirani, S.Kom., M.MT.**

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



*“Ketika kamu bermalas-malasan ingatlah bahwa banyak pesaingmu diluar sana yang sedang berjuang”*

UNIVERSITAS  
Dinamika



*Dipersembahkan untuk Ayah dan Mama yang telah memberi dukungan dan doa yang selalu diberikan kepada saya. Dan teruntuk semua orang yang juga selalu membantu secara langsung maupun tidak langsung agar saya tidak mudah menyerah dan menjadi orang yang lebih baik.*

UNIVERSITAS  
Dinamika

**SURAT PERNYATAAN  
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Ilham Rizaldy Widy Putra  
NIM : 17410200032  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Laporan Tugas Akhir  
Judul Karya : **SISTEM DETEKSI PADA SIMBOL SIBI (SISTEM  
ISYARAT BAHASA INDONESIA) MENGGUNAKAN  
CONVOLUTIONAL NEURAL NETWORK**



Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 16 Juni 2021

Yang menyatakan

  
  
**Ilham Rizaldy Widy Putra**  
NIM: 17410200032

## ABSTRAK

Penyandang disabilitas khususnya tuna rungu sudah menjadi kesatuan dalam rakyat Indonesia. Menurut Media Indonesia dalam *websitenya* dituliskan jumlah pemnyandang tuna rungu yang terdaftar pada DPT (daftar pemilih tetap) pada pemilu 2019 mencapai angka 472.852 orang. Penyandang tuna rungu memiliki bahasa isyarat sendiri yang dimana digunakan sebagai alat komunikasi mereka, namun permasalahannya banyak masyarakat masih awam dengan bahasa isyarat tersebut sehingga orang yang tidak mempelajari bahasa isyarat aka mengalami kesulitan memahami penyandang tuna rungu. SIBI (Sistem isyarat bahasa Indonesia) menjadi satu dari dua bahasa isyarat yang berada di Indonesia, namun SIBI adalah bahasa isyarat yang diresmikan oleh pemerintah pada tahun 1994 melalui Mendikbud No. 0161/U/1994 berisi tentang pembakuan SIBI (Sistem Isyarat Bahasa Indonesia). Dalam penelitian ini, akan dilakukan deteksi pada SIBI menggunakan *Convolutional Neural Network*, dengan harapan penelitian ini dapat menjadi acuan untuk bisa dikembangkan lagi sebagai salah satu teknologi untuk deteksi pada SIBI (Sistem Isyarat Bahasa Indonesia). Berbeda dengan sebelumnya dimana deteksi dilakukan pada angka dan alfabet, penelitian ini akan menggunakan *Dataset* baru namun dibatasi hanya digunakan 6 simbol pada SIBI yaitu (Saya, Kamu, Dia, Maaf, Sedih, Saya), *Dataset* juga diambil dengan 5 subjek yang berbeda dengan masing-masing akan diambil 6 simbol. Hasil dari pengujian *Training* pada bahasa isyarat mencapai akurasi di angka 90%. *Validation* data juga dilakukan dan mencapai hasil akurasi sebesar 91%. pada hasil prediksi juga dilakukan pada 5 subjek dengan menggunakan *data testing*. Pada prediksi didapatkan hasil akurasi pada simbol “Cinta” sebesar 80%, simbol “Dia” dan “Saya” sebesar 70%, simbol “Kamu” sebesar 90%, simbol “Sedih” sebesar 100% dan simbol “Maaf” mendapatkan akurasi 0%.

Kata Kunci: Tunarungu, SIBI (Sistem Isyarat Bahasa Indonesia), *Convolutional Neural Network*.

## KATA PENGANTAR

Segala puji bagi Tuhan, karena dengan rahmatNya penulis dapat menyelesaikan penyusunan Laporan Tugas Akhir yang berjudul “Sistem Deteksi Simbol Pada SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan *Convolutional Neural Network*”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Teknik pada program studi S1 Teknik Komputer Universitas Dinamika. Pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada pihak-pihak yang memberi dukungan dan masukan dalam menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan kepada:

1. Orang Tua dan Saudara-saudara saya tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir ini.
2. Bu Tri Sagirani, S.Kom., M.MT. selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika,
4. Bapak Dr. Jusak selaku Dosen Pembahas atas saran dan masukannya sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini.
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen pembimbing I yang selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
6. Ibu Musayyanah, S.ST., M.T. selaku Dosen Pembimbing II yang juga selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
7. Seluruh dosen pengajar Program Studi S1 Teknik Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Universitas Dinamika.
8. Teman-teman seperjuangan Teknik Komputer angkatan 2017 dan semua pihak yang terlibat namun tidak dapat penulis sebutkan satu persatu atas bantuan dan dukungannya.



9. Serta semua pihak lain yang tidak dapat disebutkan secara satu per satu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kata sempurna, masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih banyak kesalahan baik dalam penulisan maupun Bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, 15 Juli 2021



UNIVERSITAS  
Dinamika

Penulis

## DAFTAR ISI

	Halaman
<b>ABSTRAK .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
<b>BAB II LANDASAN TEORI .....</b>	<b>4</b>
2.1 Tunarungu.....	4
2.2 SIBI (Sistem Isyarat Bahasa Indonesia).....	4
2.3 <i>Deep Learning</i> .....	5
2.4 <i>Convolutional Neural Network</i> .....	6
2.4.1 Konvolusi .....	7
2.4.2 ReLU ( <i>Rectified Linier Unit</i> ).....	8
2.4.3 <i>Max Pooling</i> .....	8
2.4.4 <i>Classification</i> .....	9
2.5 Google Colab.....	9
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>11</b>
3.1 Pengumpulan <i>Dataset</i> .....	11

3.2 <i>Image Pre-processing Dataset</i> .....	15
3.3 <i>Flowchart dan Program</i> .....	18
3.2.1 <i>Image Pre-processing</i> .....	18
3.2.2 <i>Flowchart Deep Leaning</i> .....	19
3.2.3 <i>Data Testing</i> .....	20
3.4 <i>Arsitektur Convolutional Neural Network</i> .....	21
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>23</b>
4.1 Hasil dan Grafik Proses <i>Training</i> .....	23
4.2 Perbandingan Hasil <i>Training</i> dan <i>Validation</i> Pada Arsitektur CNN....	26
4.3 <i>Image Prediction</i> .....	28
<b>BAB V KESIMPULAN DAN SARAN</b> .....	<b>30</b>
5.1 Kesimpulan.....	30
5.2 Saran.....	30
<b>DAFTAR PUSTAKA</b> .....	<b>32</b>
<b>LAMPIRAN</b> .....	<b>35</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Alfabet dalam bahasa isyarat SIBI.....	5
Gambar 2.2 Jaringan dalam <i>Deep Learning</i> .....	6
Gambar 2.3 Arsitektur CNN .....	7
Gambar 2.4 Proses konvolusi pada CNN.....	7
Gambar 2.5 Proses <i>Max Pooling</i> pada CNN.....	8
Gambar 2.6 Tampilan Google Colab .....	10
Gambar 3.1 Bahasa isyarat “Saya”.....	11
Gambar 3.2 Bahasa isyarat “Kamu” .....	11
Gambar 3.3 Bahasa isyarat “Dia” .....	12
Gambar 3.4 Bahasa isyarat “Cinta” .....	12
Gambar 3.5 Bahasa isyarat “Maaf” .....	12
Gambar 3.6 Bahasa isyarat “Sedih” .....	12
Gambar 3.7 <i>Folder Data Training</i> .....	14
Gambar 3.8 <i>Folder Data Testing</i> .....	14
Gambar 3.9 Citra <i>Dataset</i> asli.....	15
Gambar 3.10 Hasil citra proses <i>Remove Background</i> .....	15
Gambar 3.11 Hasil proses ubah menjadi hitam putih .....	16
Gambar 3.12 Hasil proses Canny.....	16
Gambar 3.13 Hasil akhir <i>Image Pre-Processing</i> .....	17
Gambar 3.14 Folder hasil <i>Image Pre-Processing</i> .....	17
Gambar 3.15 <i>Flowchart Image Pre-Processing</i> .....	18
Gambar 3.16 <i>Flowchart Deep Learning</i> .....	19
Gambar 3.17 <i>Flowchart Data Testing</i> .....	21
Gambar 3.18 Arsitektur CNN yang digunakan.....	22
Gambar 3.19 <i>Summary</i> model arsitektur CNN .....	22
Gambar 4.1 Grafik akurasi dengan 5 <i>Layer</i> .....	24
Gambar 4.2 Grafik <i>Loss</i> dengan 5 <i>Layer</i> .....	24
Gambar 4.3 Grafik akurasi dengan 6 <i>Layer</i> .....	25
Gambar 4.4 Grafik <i>Loss</i> dengan 6 <i>Layer</i> .....	25

Gambar L5.1 Hasil turnitin halaman 1 .....	52
Gambar L5.2 Hasil turnitin halaman 2 .....	53
Gambar L5.3 Hasil turnitin halaman 3 .....	54
Gambar L5.4 Hasil turnitin halaman 4 .....	55



UNIVERSITAS  
Dinamika

## DAFTAR TABEL

	Halaman
Tabel 3.1 Tabel <i>Summary Dataset</i> .....	13
Tabel 4.1 Akurasi Hasil <i>Training</i> .....	23
Tabel 4.2 Tabel performa hasil model <i>5 Layer</i> .....	26
Tabel 4.3 Tabel Perbandingan Hasil <i>Training</i> .....	26
Tabel 4.4 Tabel Hasil Prediksi .....	29
Tabel L4.1 Tabel Hasil <i>Training</i> dan <i>Validation</i> .....	45



UNIVERSITAS  
**Dinamika**

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 <i>Source Code Program Image Pre-Processing</i> .....	35
Lampiran 2 <i>Source Code Program Deep Learning</i> .....	38
Lampiran 3 <i>Source Code Program Data Testing</i> .....	43
Lampiran 4 Hasil <i>Training dan Validation</i> .....	45
Lampiran 5 Hasil Turnitin.....	52



UNIVERSITAS  
**Dinamika**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Penyandang disabilitas sudah menjadi kesatuan dari sekian masyarakat di dunia, salah satu dari penyandang disabilitas adalah penyandang tuna rungu. Penyandang tunarungu ialah orang yang mengalami keterbatasan pada pendengarannya sehingga penyandang tunarungu memiliki Bahasa isyarat tertentu dalam komunikasinya. Indonesia termasuk salah satu yang memiliki masyarakat penyandang tunarungu, menurut (Jelita, 2018) yang ditulis pada *website* Media Indonesia, jumlah penyandang tunarungu yang menjadi DPT (daftar pemilih tetap) pada pemilu 2019 berjumlah 472.852 orang.

Di Indonesia, penyandang tuna rungu memiliki 2 jenis bahasa isyarat yang digunakan, yaitu BISINDO adalah Bahasa Isyarat Indonesia yang dimana bisa dibilang merupakan bahasa ibu yaitu bahasa yang hidup dari kelompok tunarungu itu sendiri, sehingga di setiap tempat dapat memiliki perbedaan dalam komunikasi isyarat. Maka dari itu pemerintah di Indonesia menciptakan sebuah sistem yang *Universal* yang diambil dari referensi ASL (*American Sign Language*) sehingga tercipta SIBI yang merupakan Sistem Isyarat Bahasa Indonesia.

Komunikasi bahasa isyarat diperuntukkan untuk komunikasi penyandang tunarungu sedangkan orang normal yang tidak pernah belajar mengenai sistem isyarat tentu mengalami kesulitan, namun di era sekarang manusia dipermudahkan teknologi untuk berkomunikasi, dengan bahasa yang berbeda *gesture* yang berbeda, teknologi dapat mengenali dan mengkonversi kedalam *output* tertentu. Salah satu teknologinya adalah *Deep Learning*. *Deep learning* secara singkat adalah sebuah model yang bisa mempelajari dengan sendirinya sebuah metode komputasi. Pada penelitian sebelumnya ada 2 yang meneliti mengenai bahasa isyarat dengan 2 metode yang berbeda, yang pertama oleh (Darmatasia, 2020), membahas mengenai pengenalan alfabet SIBI (Sistem Isyarat Bahasa Indonesia) dengan menggunakan *Gradient-Convolutional Neural Network*, mendapatkan akurasi tertinggi pengenalan yang didapatkan hingga mencapai 98%. Namun pada penelitian tersebut, dibatasi oleh jumlah *Dataset* sejumlah 5 alfabet (A, B, C, D, E), yang



kedua oleh (Asriani & Susilawati, 2010) membahas mengenai pengenalan isyarat tangan statis pada Sistem Isyarat Bahasa Indonesia berbasis jaringan syaraf tiruan perambatan balik, namun kekurangan dari penelitian tersebut adalah maksimal akurasi deteksi yang didapatkan hanya pada angka 69%.

Sehingga dari latar belakang diatas, pada Tugas Akhir ini diajukan rancangan sistem deteksi simbol SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan *Convolutional Neural Network*. Langkah pertama yaitu pengumpulan *Dataset Training* dan *Dataset uji* berupa citra, kemudian pengolahan citra menggunakan *Convolutional Neural Network*. Lalu pengujian akurasi deteksi simbol SIBI (Sistem Isyarat Bahasa Indonesia) untuk menentukan akurasi dari masing-masing simbol.

Pada Tugas Akhir ini membuat sendiri keenam simbol SIBI (Sistem Isyarat Bahasa Indonesia) melalui 5 partisipan, dimana setiap partisipan akan diambil datanya dari tiap bentuk simbol SIBI (Sistem Isyarat Bahasa Indonesia), sehingga data yang didapatkan secara total dari kelima partisipan tersebut sebanyak 660 buah citra. Dari 660 *Dataset* tersebut, sebanyak 60 citra digunakan sebagai validasi dan *testing* dan 540 buah citra *Dataset* digunakan sebagai *Training* pada model arsitektur CNN.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut:

1. Bagaimana cara menerapkan sistem pendeteksi simbol pada Sistem Isyarat Bahasa Indonesia menggunakan *Convolutional Neural Network* ?
2. Berapa besar tingkat akurasi sistem deteksi simbol pada Sistem Isyarat Bahasa Indonesia menggunakan CNN ?

## 1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi beberapa hal berikut :

1. Pembatasan hanya 6 simbol bahasa isyarat yang menjadi *Dataset*.
2. Pengujian yang dilakukan adalah pengujian akurasi deteksi.
3. Pengujian tidak dilakukan secara *Real-time* melainkan citra yang di *Load*.

#### 1.4 Tujuan

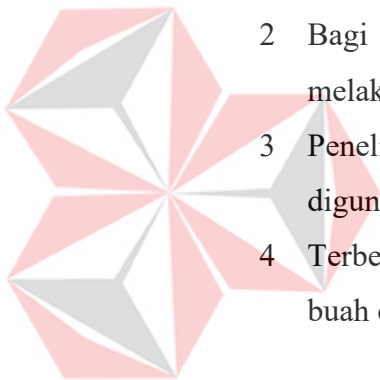
Berdasarkan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada tugas akhir ini sebagai berikut :

- 1 Mampu menerapkan sistem pendeteksi simbol pada Sistem Isyarat Bahasa Indonesia menggunakan *Convolutional Neural Network*.
- 2 Mengetahui tingkat akurasi yang didapatkan pada sistem deteksi simbol Sistem Isyarat Bahasa Indonesia menggunakan CNN.

#### 1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut :

- 1 Bagi penulis yaitu untuk menambah pengetahuan dan penerapan mengenai deteksi simbol pada SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan *Convolutional Neural Network*.
- 2 Bagi mahasiswa yaitu menjadi referensi bagi mahasiswa yang akan melakukan penelitian menggunakan *Convolutional Neural Network*.
- 3 Penelitian ini memberikan kontribusi berupa sistem deteksi yang bisa digunakan pada Sistem Isyarat Bahasa Indonesia.
- 4 Terbentuknya *Dataset* Sistem Isyarat Bahasa Indonesia sebanyak 660 buah citra.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tunarungu**

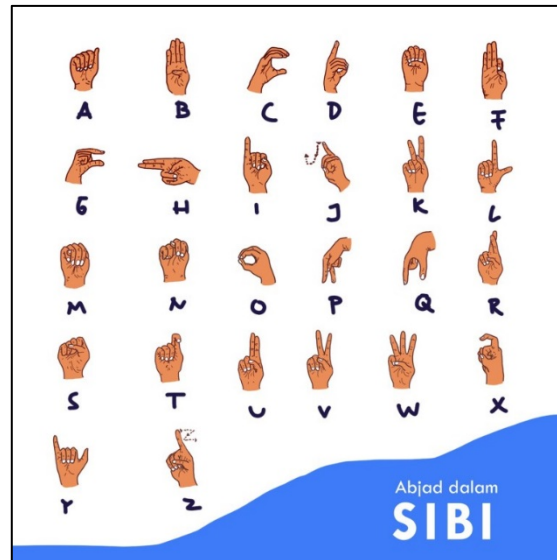
Dikutip dari (Mursita, 2015) tuna rungu ialah orang yang mengalami kekurangan atau kehilangan kemampuan mendengar yang disebabkan kerusakan atau tidak berfungsinya sebagian atau seluruh alat pendengaran sehingga ia mengalami hambatan dalam perkembangan bahasanya. Menurut (Jelita, 2018), jumlah penyandang tunarungu yang terdaftar menjadi DPT (daftar pemilih tetap) untuk pemilu 2019 berjumlah 472.852 orang dari jumlah total 6.008.640 orang penyandang disabilitas.

Sehingga penyandang tuna rungu menggunakan bahasa isyarat sebagai alat untuk berkomunikasi. Di Indonesia terdapat 2 bahasa isyarat yang digunakan, yang pertama adalah BISINDO (Bahasa Isyarat Bahasa Indonesia) dan yang kedua adalah SIBI (Sistem Isyarat Bahasa Indonesia). Perbedaan yang mendasar pada kedua bahasa isyarat tersebut adalah BISINDO lahir pada golongan penyandang tunarungu itu sendiri, sehingga pada beberapa daerah memiliki perbedaan bahasa isyarat, lalu untuk SIBI adalah Sistem yang di resmi kan oleh pemerintah sebagai bahasa pemersatu meskipun tergolong bahasa isyarat yang formal.

#### **2.2 SIBI (Sistem Isyarat Bahasa Indonesia)**

Pada tanggal 30 Juni 1994, dengan keputusan Mendikbud No. 0161/U/2994 berisi tentang pembakuan SIBI (Sistem Isyarat Bahasa Indonesia) pemerintah Indonesia telah meresmikan Kamus SIBI (Sistem Isyarat Bahasa Indonesia), dengan itu pendidikan secara formal untuk penyandang tunarungu dapat menggunakan acuan dari kamus SIBI (Sistem Isyarat Bahasa Indonesia)

Sistem Isyarat Bahasa Indonesia (SIBI) itu sendiri di adopsi dari bahasa isyarat Amerika atau lebih dikenal dengan ASL (*American Sign Language*). Sistem Isyarat Bahasa Indonesia (SIBI) lebih sering digunakan secara formal seperti pada Pendidikan Sekolah dan Sistem Isyarat Bahasa Indonesia (SIBI) pada penerapannya salah satunya memiliki individual bahasa isyarat untuk alfabet seperti yang dapat dilihat pada Gambar 2.1 dibawah ini.



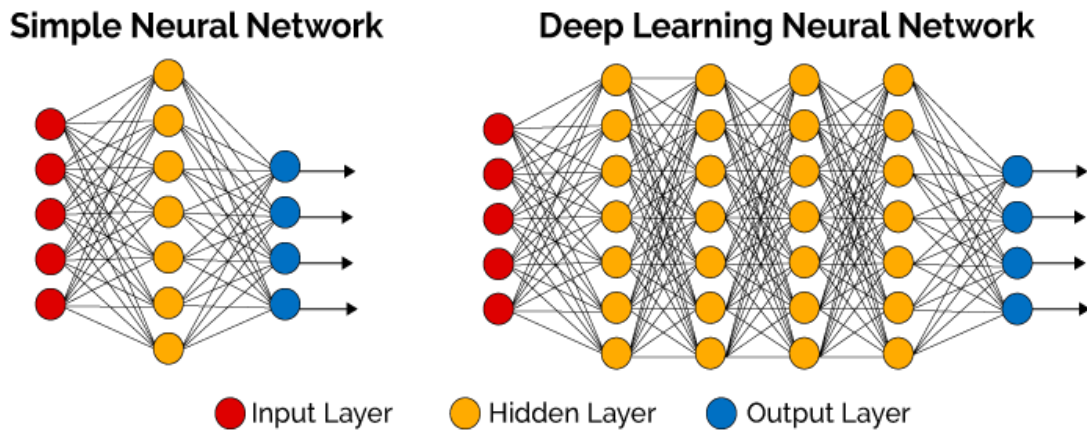
Gambar 2.1 Alfabet dalam bahasa isyarat SIBI  
(Sumber : (Maharsi, 2018))

### 2.3 Deep Learning

Pada era sekarang, *Deep Learning* sudah banyak diaplikasikan pada teknologi yang digunakan oleh manusia. Pengenalan wajah pada *Smartphone*, mobil Tesla yang dapat dikendalikan secara *Auto Pilot*, klasifikasi pada *object* tertentu, itu menjadi bagian dari penerapan *Deep Learning*. Secara definisi, *Deep learning* menurut (Nugroho, et al., 2020) dijelaskan bahwa:

merupakan metode *learning* yang memanfaatkan *artificial neural network* yang berlapis-lapis (*multi-Layer*). *Artificial Neural Network* ini dibuat mirip otak manusia, dimana neuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit. Pada *Deep learning* metode pembelajarannya juga dapat berupa *unsupervised*, *supervised* dan *semi-supervised*.

Pada jaringan yang terbagi menjadi beberapa neuron dan terbagi pada beberapa *Layer* yang mempunyai fungsi berbeda - beda, dengan jaringan tersebut akan digunakan *Deep Learning* untuk dapat menganalisa macam – macam dimensi dari data sehingga menghasilkan kesimpulan yang tepat. *Layer* biasa terdiri dari 3 bagian yaitu *Input Layer*, *Hidden Layer* dan *Output Layer*, seperti yang dapat dilihat pada Gambar 2.2 dibawah ini.



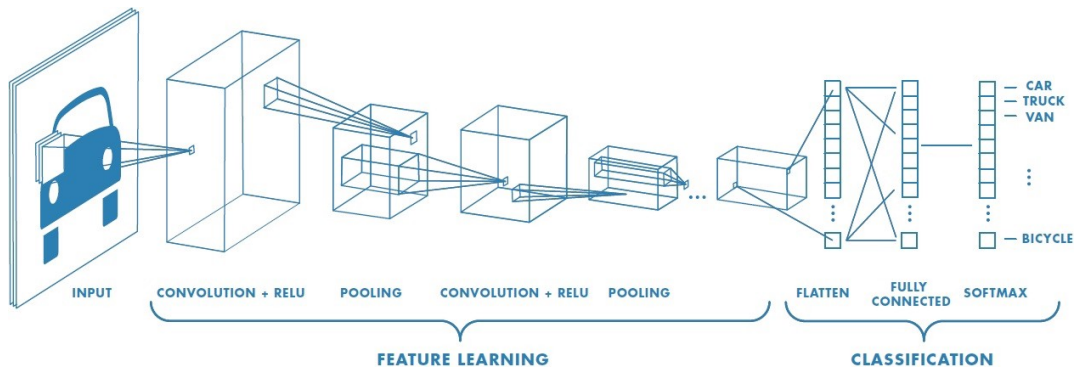
Gambar 2.2 Jaringan dalam *Deep Learning*  
(Sumber : (INIXINDO, 2021))

## 2.4 *Convolutional Neural Network*

Pengertian *Convolutional Neural Network* Menurut (Nugroho, et al., 2020) *Convolutional Neural Network* adalah pengembangan dari *MultiLayer Perceptron* (MLP) yang termasuk dalam *neural network* bertipe *feed forward* (bukan berulang).

*Convolutional Neural Network* adalah desain *Neural Network* yang dapat menganalisa maupun mengolah data yang mempunyai dua dimensi. Karena jaringan tinggi yang sangat dalam serta banyak diaplikasikan untuk klasifikasi pada data citra sehingga *Convolutional Neural Network* termasuk dalam kategori *Deep Neural Network*. *Convolutional Neural Network* ini dapat digunakan untuk analisa pada visual citra lalu dilakukan *Training* untuk dapat mengenali dan mendeteksi objek-objek tertentu pada sebuah citra, dengan acuan berupa parameter pada citra dari vector yang berdimensi tinggi agar menjadi ciri yang khas dari citra tersebut. *Convolutional Neural Network* secara garis besar tidak berbeda jauh dengan *Neural Network* pada umumnya.

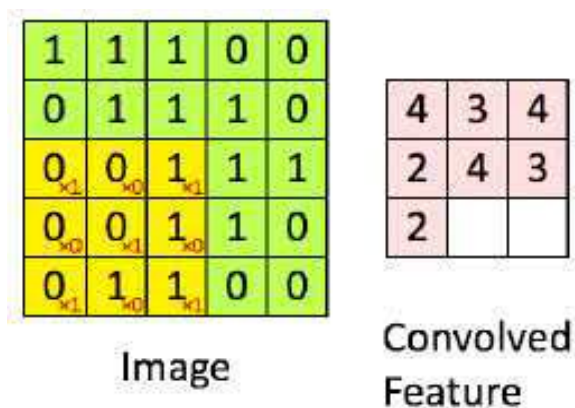
Arsitektur pada *Convolutional Neural Network* yang digunakan ada 2, seperti Gambar 2.3 adalah *Feature Learning* dan *Classification*, pada *Feature Learning* dalam satu *Layer* berisi *Convolution*, *Relu* (sebagai *Activation Function*), setelah itu *Max Pooling*. Lalu pada *Classification* berisi *Flatten*, *Fully Connected*, lalu dengan fungsi aktivasi *Softmax* untuk melakukan pengklasifikasian.



Gambar 2.3 Arsitektur CNN  
(Sumber : (Sena, 2017))

### 2.4.1 Konvolusi

Konvolusi berfungsi sebagai fitur ekstraksi dari citra yang diinputkan nantinya. Konvolusi akan memerlukan *kernel size* sebagai *filter* untuk mengekstraksi fitur pada citra, maka dari itu *kernel size* yang digunakan berukuran 3x3, lalu untuk pergesaran *kernel* akan digunakan *Strides* 1, untuk contoh fitur ekstraksi dapat dilihat pada Gambar 2.4. Setelah fitur diekstraksi akan dilakukan fungsi aktivasi menggunakan ReLU (*Rectified Linier Unit*), selain itu ditambahkan dengan yang namanya *Padding*, disini *Padding* berfungsi untuk mengakali fitur agar bisa di ekstrak fitur nya lebih banyak lagi dengan cara memberikan dimensi tambahan pada sisi-sisi dari citra bernilai 0. Pada *Layer* konvolusi ini akan digunakan beberapa *Layer* untuk percobaan, akan digunakan percobaan 2 *Layer*, 3 *Layer* hingga 4 *Layer*.



Gambar 2.4 Proses konvolusi pada CNN  
(Sumber : (Suartika, et al., 2016))

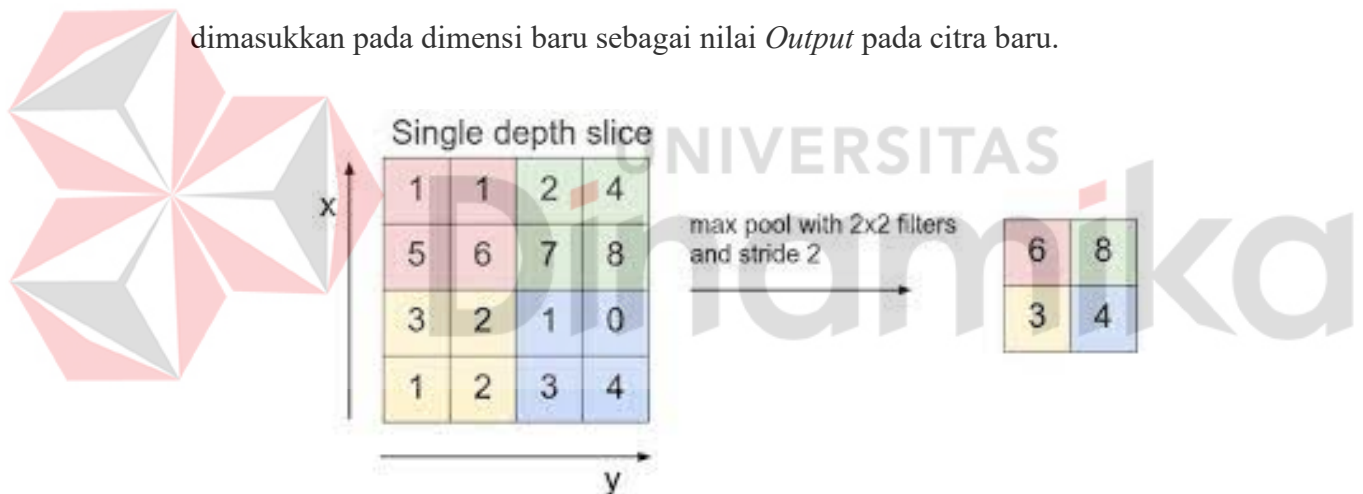


### 2.4.2 ReLU (*Rectified Linier Unit*)

Fungsi aktivasi ini akan berfungsi untuk mengubah nilai atau menormalisasi pada hasil fitur ekstraksi sebelumnya. Cara kerja dari fungsi aktivasi ini jika pada fitur ekstraksi terdapat nilai negatif pada fiturnya maka akan dinormalisasikan menjadi nilai 0. Fungsi aktivasi ini dirumuskan sebagai berikut  $f(x) = \max(0, x)$ .

### 2.4.3 Max Pooling

Fungsi dari *Max Pooling* adalah untuk mereduksi atau mengurangi dimensi dari citra untuk mempermudah komputasi. Contoh fitur ekstraksi dari *Max Pooling* dapat terlihat pada Gambar 2.5. dalam contoh digunakan filter dengan ukuran 2x2 dan *Strides* bernilai 1, namun juga bisa digunakan filter berukuran 3x3 dan *Strides* bernilai 2. Cara kerjanya adalah membagi *Grid* pada fitur citra sesuai dengan dimensi pada *kernel filter*, lalu nilai fitur pada dicari nilai maksimalnya dan dimasukkan pada dimensi baru sebagai nilai *Output* pada citra baru.



Gambar 2.5 Proses *Max Pooling* pada CNN  
(Sumber : (Suartika, et al., 2016))

#### 2.4.4 Classification

Sebelum masuk pada *Fully Connected Layer* hasil output dari *Feature Extraction* masih dalam bentuk 2 dimensi, maka dari itu akan dilakukan proses *Flatten* yang berfungsi mengubah dimensi citra dari 2 dimensi menjadi 1 dimensi, hal ini dilakukan agar klasifikasi pada *Layer* setelahnya dapat dilakukan secara linear. Setelah itu juga akan dilakukan dengan yang namanya *Dropout*, berfungsi untuk membuang nilai yang dirasa tidak perlu untuk di *Training* nantinya agar memudahkan komputasi pada *Deep Learning*. Lalu citra 1 dimensi tersebut akan masuk pada *Layer Dense* yang merupakan *Hidden Layer* dalam bentuk *neuron* atau jaringan, Pada *Layer* dense juga diberi fungsi aktivasi ReLU (*Rectified Linier Unit*). di *Layer* ini yang nantinya akan berfungsi untuk klasifikasi dan umumnya untuk proses klasifikasi dengan *class* lebih dari dua atau *categorical* digunakan fungsi aktivasi *Softmax*. Dalam *Softmax* ini yang akan menentukan nilai-nilai untuk masing-masing kategori atau *class*, dimana untuk *data testing* yang nanti nya di prediksi akan di tentukan nilai nya itu lebih mendekati *class* yang mana.

#### 2.5 Google Colab

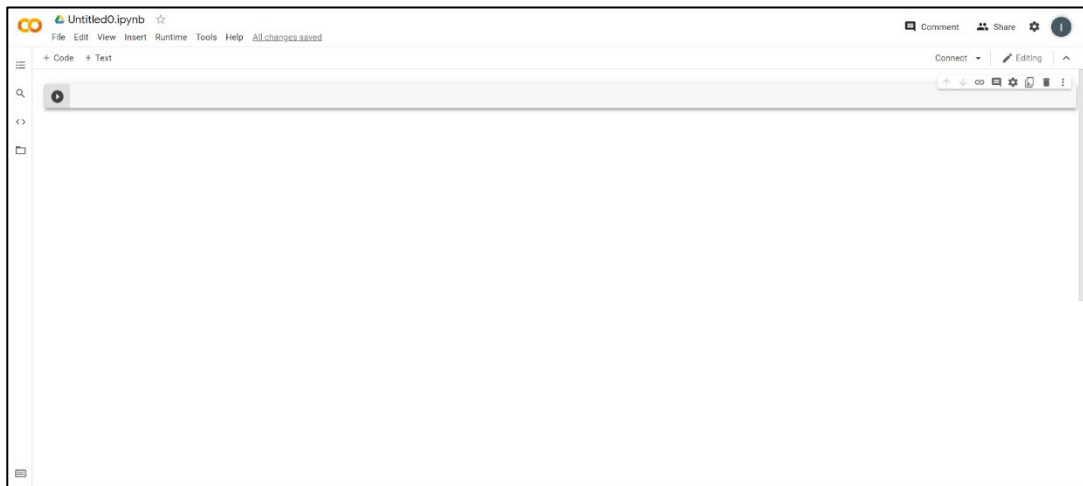
Dikutip dari (Ahmadian, 2020) dalam artikelnya yang diunggah di *website Wordpress* mengenai Google colab dijelaskan bahwa:

“*Google Colaboratory* atau disingkat Google Colab adalah satu *tools* yang dikembangkan oleh Google untuk data pecinta *Science*, mahasiswa atau penikmat *Artificial Intelligence* untuk menjalankan dan mengeksekusi *code* Python di *Browser*. Google Colab persis seperti *Jupyter Notebook*. Beberapa keunggulan penggunaan Google Colab ini antara lain tidak perlu mengkonfigurasi apapun karena menggunakan teknologi *cloud computing*, akses gratis untuk mesin berkecepatan tinggi (GPU) dan sangat mudah dikoneksikan dengan Google Drive dan github”.

Google colab menyediakan *open source* untuk *user* yang ingin belajar pemrograman bahasa Python, sehingga Google Colab sangat direkomendasikan untuk pemula pada bidang data. Keuntungan menggunakan yaitu Google colab sangatlah fleksibel dikarenakan mudah sekali untuk menghubungkan dengan *Jupyter Notebook*. *User* juga dapat terhubung dengan Google Drive bagi yang menyimpan banyak data pada Google Drive. Selain itu bagi yang biasa untuk mengakses data-data pada Github juga dapat dengan mudah untuk terhubung.



Penggunaan pada Google Colab tentunya akan memerlukan akses berupa akun dari Gmail. Tampilan Google Colab yang sudah dapat akses dari akun Gmail dapat dilihat pada Gambar 2.6.



Gambar 2.6 Tampilan Google Colab



UNIVERSITAS  
**Dinamika**

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Pengumpulan *Dataset*

*Dataset* yang digunakan yaitu dengan batasan beberapa simbol bahasa isyarat yaitu 3 kata ganti orang (Saya, Kamu dan Dia) seperti pada Gambar 3.1, Gambar 3.2 dan Gambar 3.3, kata perasaan (Cinta, Maaf, Sedih) seperti pada Gambar 3.4, Gambar 3.5 dan Gambar 3.6.

*Dataset* diambil dengan menggunakan kamera *Logitech C920*. Masing-masing simbol memiliki 90 citra sebagai *Data Training* dengan total *Dataset* sebanyak 660 citra terbagi menjadi 540 citra sebagai *Data Training*, 60 citra sebagai *Data Validasi* dan 60 citra sebagai *Data Testing*. Bahasa isyarat terdapat 2 jenis yaitu statik seperti bahasa isyarat Saya, Kamu, Dia, Cinta. Lalu dinamik seperti bahasa isyarat Sedih dan Maaf, sehingga citra diambil dengan urutan bergerak isyarat.



Gambar 3.1 Bahasa isyarat “Saya”



Gambar 3.2 Bahasa isyarat “Kamu”



Gambar 3.3 Bahasa isyarat “Dia”



Gambar 3.4 Bahasa isyarat “Cinta”



Gambar 3.5 Bahasa isyarat “Maaf”



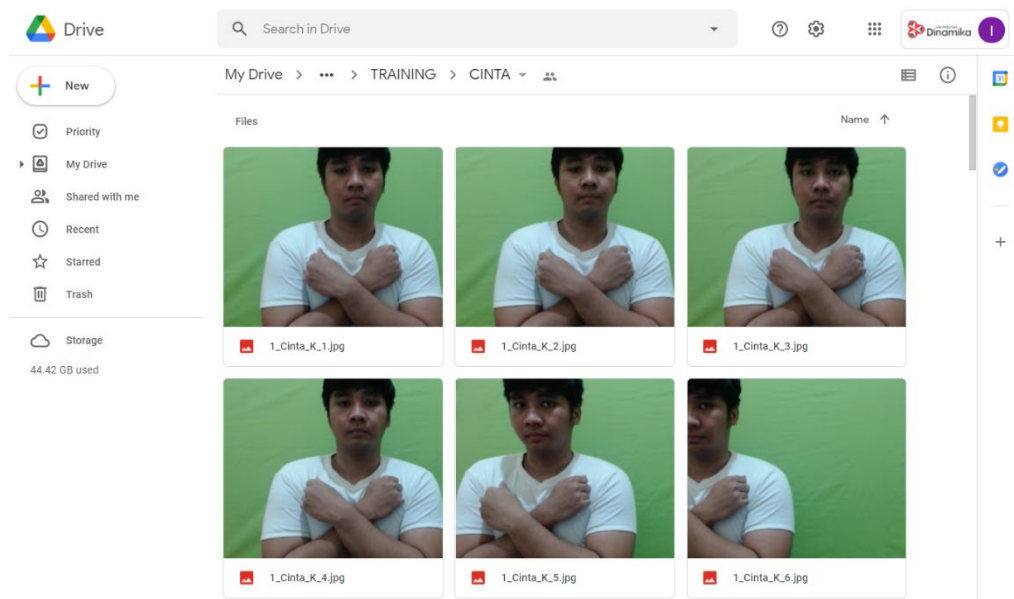
Gambar 3.6 Bahasa isyarat “Sedih”



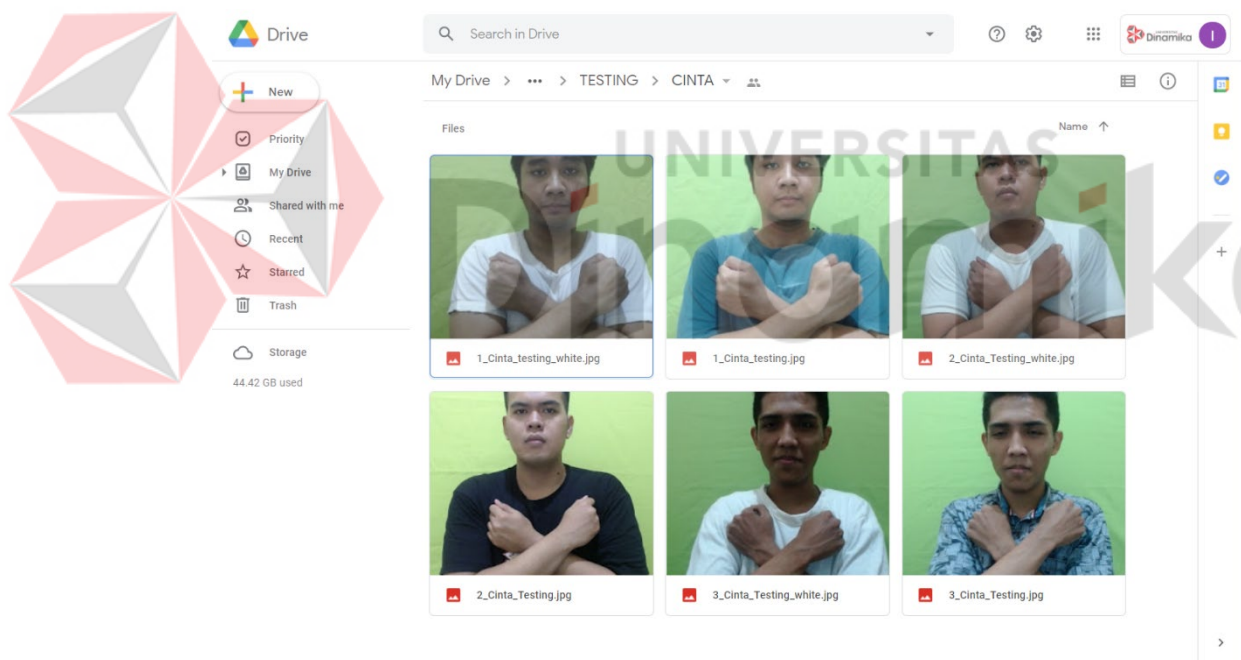
Tabel 3.1 Tabel *Summary Dataset*

No	Subjek	Usia	Gender	Tanggal Ambil Data	Jumlah Data (citra)
1	Subjek 1	23 Tahun	Laki-laki	“Saya” (12/04/2021) “Kamu” (19/04/2021) “Dia” (19/04/2021) “Maaf” (19/04/2021) “Cinta” (19/04/2021) “Sedih” (20/04/2021)	<i>Training</i> :108 Validasi: 12 Testing:12
2	Subjek 2	21 Tahun	Laki-laki	“Saya” (20/04/2021) “Kamu” (20/04/2021) “Dia” (20/04/2021) “Maaf” (20/04/2021) “Cinta” (20/04/2021) “Sedih” (20/04/2021)	<i>Training</i> :108 Validasi: 12 Testing:12
3	Subjek 3	23 Tahun	Laki-laki	“Saya” (20/04/2021) “Kamu” (20/04/2021) “Dia” (20/04/2021) “Maaf” (20/04/2021) “Cinta” (20/04/2021) “Sedih” (20/04/2021)	<i>Training</i> :108 Validasi: 12 Testing:12
4	Subjek 4	51 Tahun	Wanita	“Saya” (23/04/2021) “Kamu” (23/04/2021) “Dia” (23/04/2021) “Maaf” (23/04/2021) “Cinta” (23/04/2021) “Sedih” (23/04/2021)	<i>Training</i> :108 Validasi: 12 Testing:12
5	Subjek 5	22 Tahun	Wanita	“Saya” (27/04/2021) “Kamu” (27/04/2021) “Dia” (27/04/2021) “Maaf” (27/04/2021) “Cinta” (27/04/2021) “Sedih” (27/04/2021)	<i>Training</i> :108 Validasi: 12 Testing:12

Tabel 3.1 menunjukkan *Summary* dari *Dataset* yang diambil, variasi subjek diambil sebanyak 5 subjek berbeda. *Dataset* diambil pada lokasi yang sama, serta menggunakan *Green Screen* sebagai standar *background*. Pakaian yang digunakan juga berwarna putih sebagai standar *Dataset*, namun untuk *Data Testing* diberi tambahan variasi dengan pakaian yang tidak berwarna putih. Lalu *file Dataset* di unggah ke Google Colab sebagai direktori utama. Dibagi menjadi 2 *folder* yaitu *folder* “*Training*” yang dapat dilihat pada Gambar 3.7 dan *folder* “*Testing*” seperti pada Gambar 3.8.



Gambar 3.7 Folder Data Training



Gambar 3.8 Folder Data Testing

### 3.2 *Image Pre-processing Dataset*

Sebelum masuk untuk proses *Training* pada *Convolutional Neural Network*, dilakukan *Image Pre-processing*, hal ini dilakukan dengan alasan *Dataset* hanya akan di deteksi sebagai orang, dalam artian tidak terfokus dengan simbol bahasa isyarat, maka dari itu diperlukan *Image Pre-processing*. Yang pertama dilakukan adalah *removing background* dari citra *Dataset* asli seperti pada Gambar 3.9, dengan menggunakan fungsi `cv2.inRange`, citra diberi 2 *threshold* warna hijau, nilai pada *threshold* diatur dengan rentang nilai 4-85, proses nya ketika nilai pada citra lebih dari 85 atau kurang dari 4 maka nilai akan menjadi 0, ketika nilai diantara 4-85 maka nilai menjadi 255, hasil dari proses ini terlihat pada Gambar 3.10



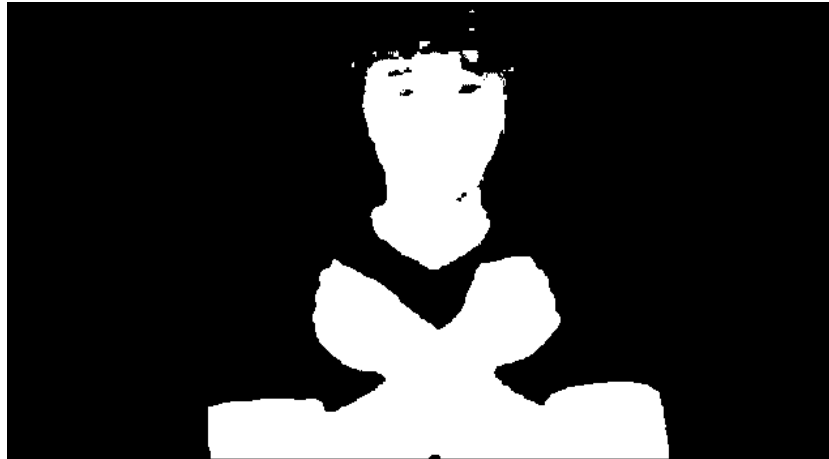
Gambar 3.9 Citra *Dataset* asli



Gambar 3.10 Hasil citra proses *Remove Background*



Hasil citra asli yang telah di *remove background* nya, di-*resize* menjadi ukuran dimensi 100x100. Lalu dilakukan proses mengubah citra objek menjadi berwarna putih, nilai citra pada yang bernilai 255 diubah menjadi putih, sedangkan yang bernilai 0 tetap bernilai 0. Hasil nya seperti pada Gambar 3.11.



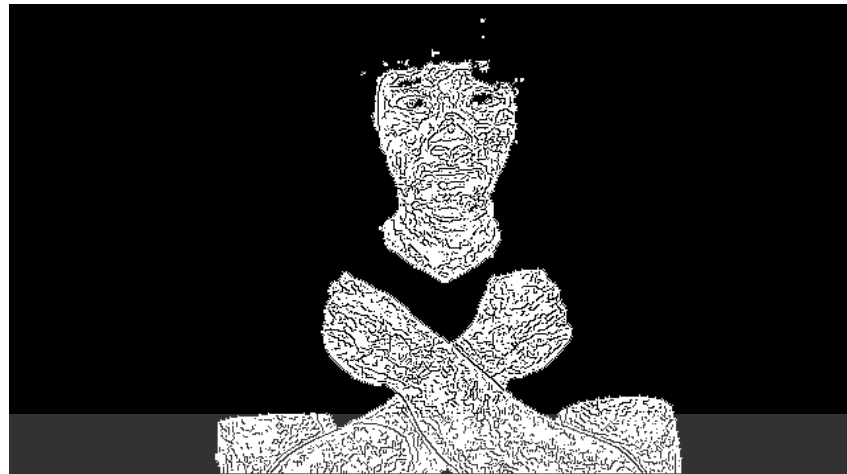
Gambar 3.11 Hasil proses ubah menjadi hitam putih

Dari hasil citra asli yang telah di-*remove background* nya, juga dilakukan proses *Edge Detection* menggunakan Canny. Pemilihan filter Canny adalah dikutip dari (Katiya & Arun, 2014) dalam penelitiannya mendapatkan kesimpulan bahwa *Edge Detection* yang terbaik adalah Canny. Untuk proses Canny adalah proses untuk mencari garis tepi pada citra, sehingga garis tangan dapat terlihat jelas dan hasilnya dapat dilihat seperti Gambar 3.12



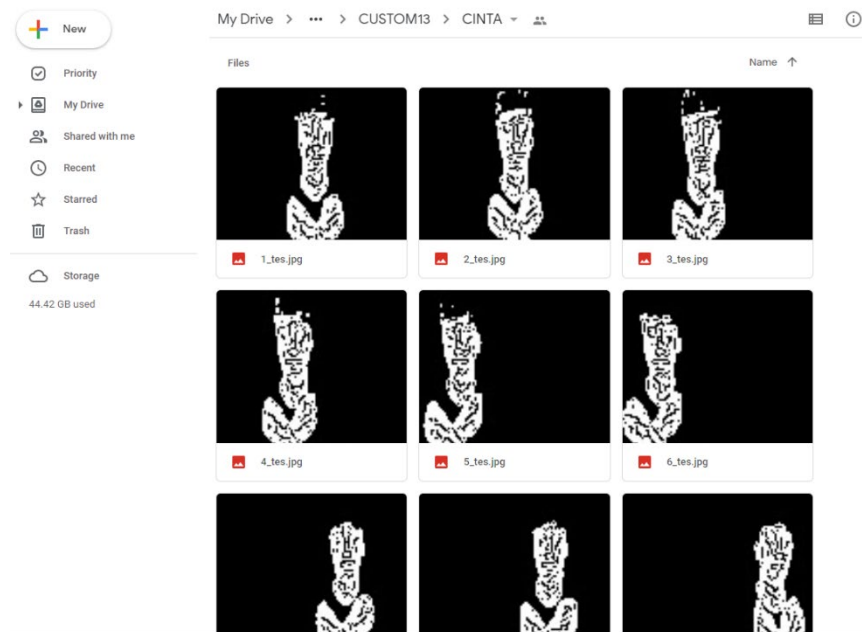
Gambar 3.12 Hasil proses Canny

Setelah itu memberikan fungsi `cv2.InRange` pada citra hasil proses canny (Gambar 3.12) dengan citra yang telah diubah menjadi warna hitam putih (Gambar 3.11), lalu diberi fungsi `cv2.InRange`, disini berfungsi untuk mengubah warna putih menjadi hitam dan begitu pula sebaliknya sehingga hasilnya seperti pada Gambar 3.13, hasil *image* ini yang menjadi *Data Training*.



Gambar 3.13 Hasil akhir *Image Pre-processing*

Hasil dari proses *Image Pre-processing* disimpan pada *folder* baru (Gambar 3.14) di Google Drive untuk memisahkan *folder* citra asli dengan citra *Image Pre-processing*. Hasil inilah yang menjadi *Data Training*, *Data Validation* dan *Data Testing*.



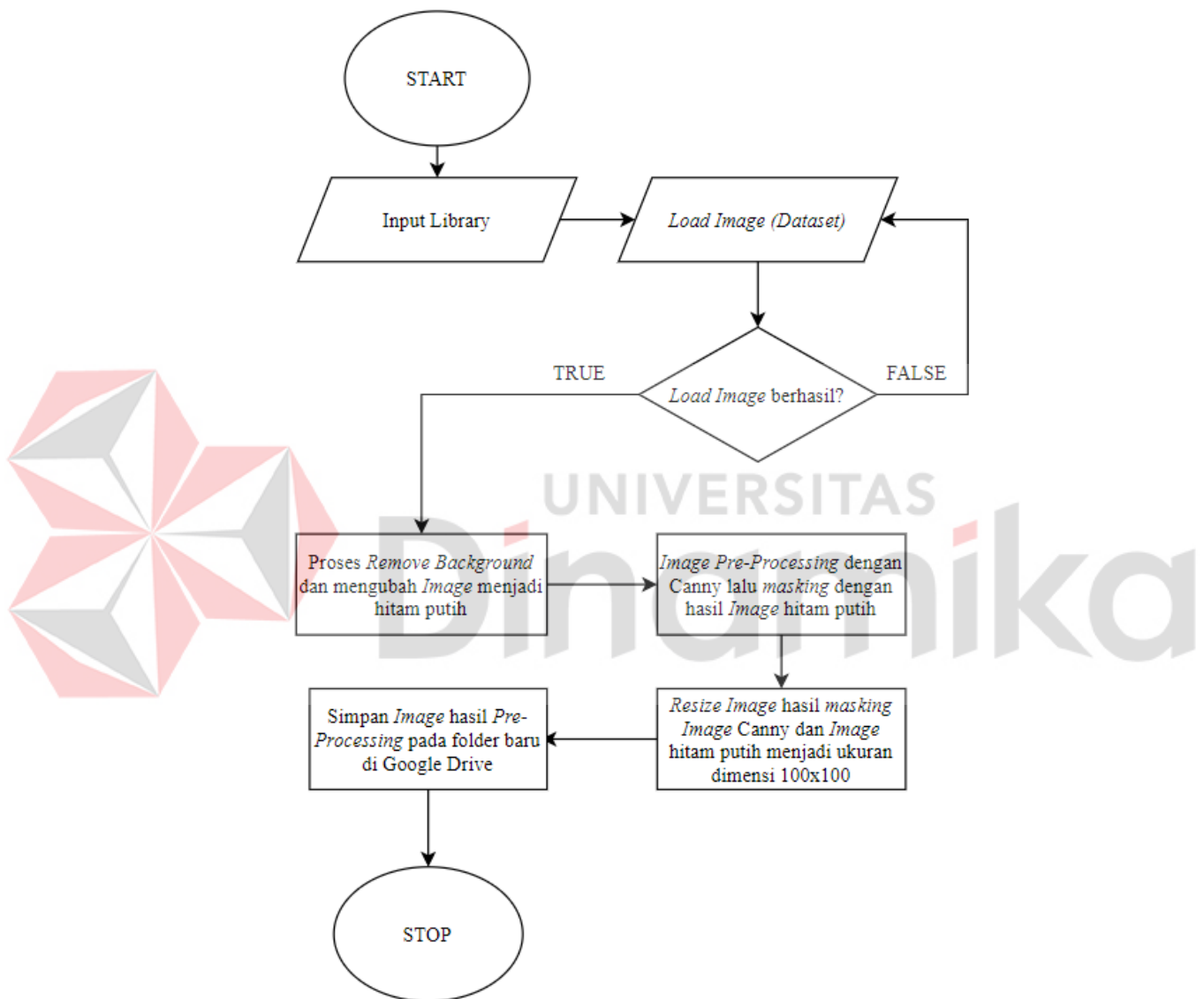
Gambar 3.14 *Folder* hasil *Image Pre-processing*



### 3.3 Flowchart dan Program

#### 3.2.1 Image Pre-processing

Sebelum memasuki *Training Image* pada *Convolutional Neural Network*, *Dataset* terlebih dahulu dilakukan *Image Pre-processing* dengan *Flowchart* seperti pada Gambar 3.15 dibawah ini.



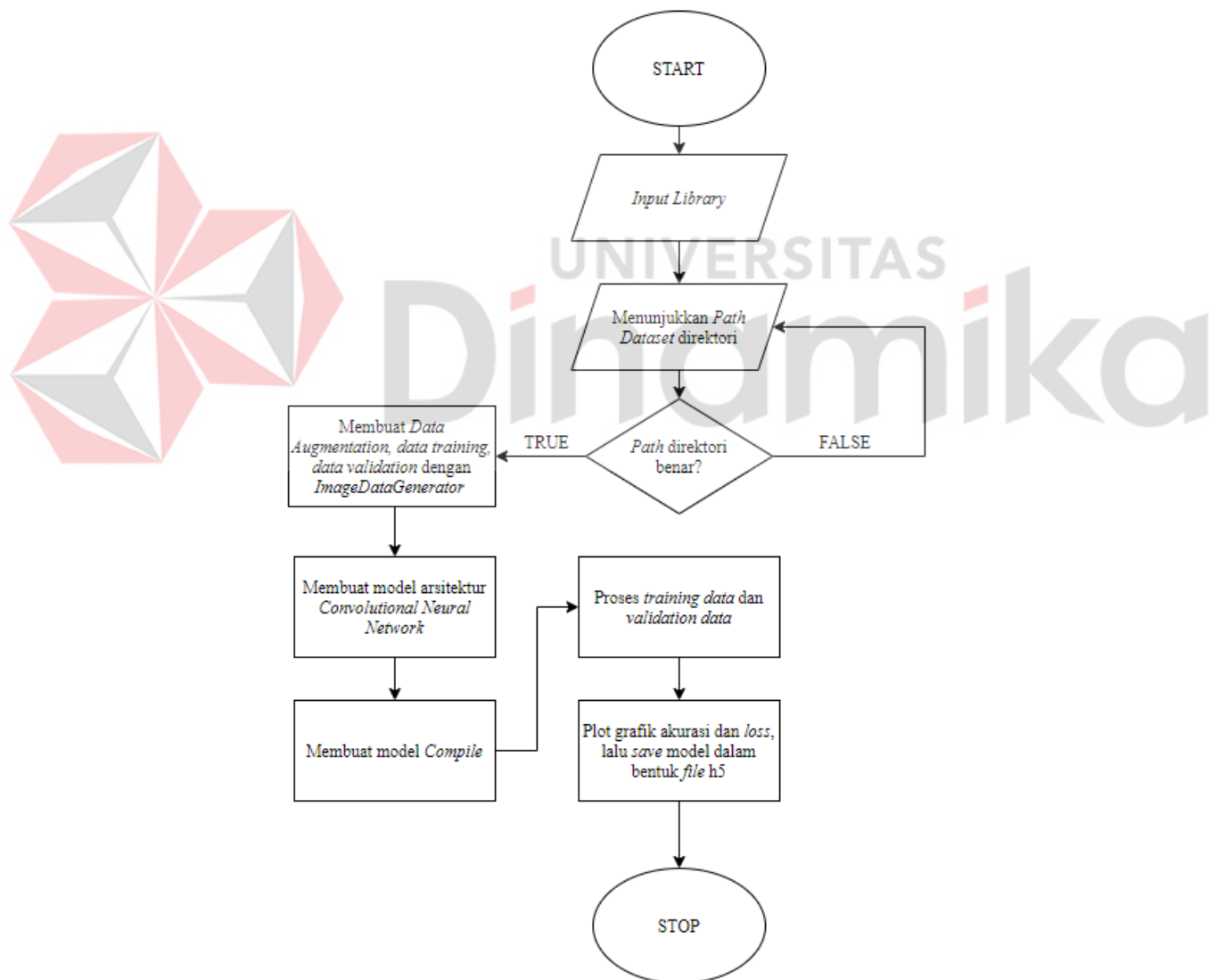
Gambar 3.15 Flowchart Image Pre-processing

Setelah program start, input library lalu Dataset di-Load untuk dilakukan Pre-processing, pertama Image Dataset asli diproses untuk remove background (Green Screen). Lalu Image juga diubah menjadi warna hitam putih, sehingga Image sekarang memiliki 2 bentuk, yaitu Image tanpa background dengan mode

warna RGB dan *Image* dengan warna hitam putih. Selanjutnya. *Image* asli tanpa *background* difilter dengan Canny untuk mendapatkan garis tepi *Image*. Lalu dimasking dengan *Image* berwarna hitam putih. Hasil proses *Masking Image* di-*resize* menjadi ukuran berdimensi 100x100. Hasil *Image Pre-processing* disimpan pada folder yang baru di Google drive sesuai dengan simbol SIBI (Sistem Isyarat Bahasa Indonesia) masing-masing, maka dengan itu program berhenti/selesai.

### 3.2.2 Flowchart Deep Learning

Proses dari *Deep Learning* dapat dilihat dari Gambar 3.16 yaitu *Flowchart Deep Learning*.

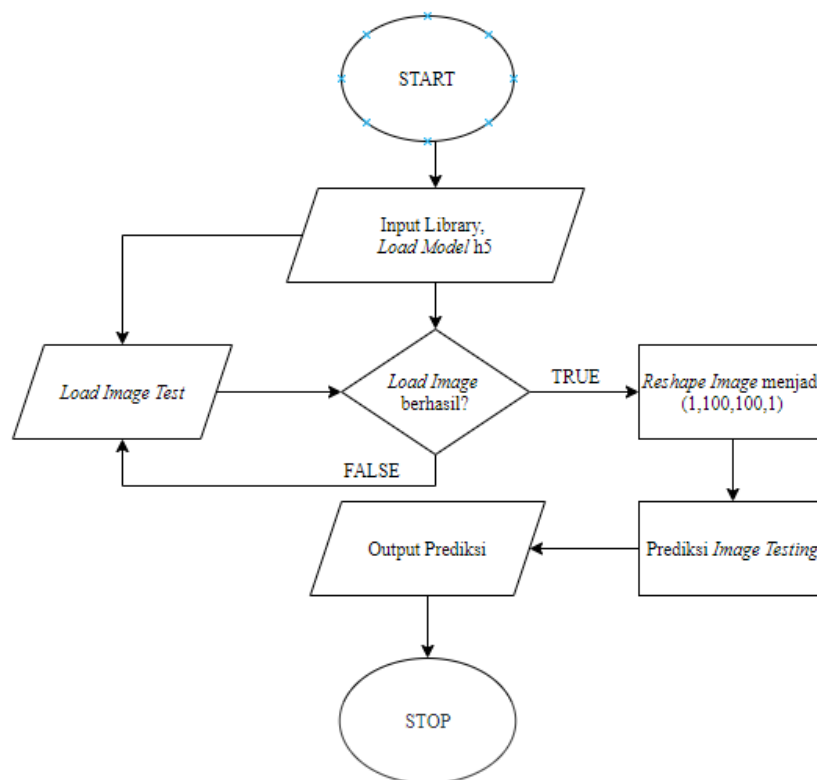


Gambar 3.16 *Flowchart Deep Learning*

Pada Gambar 3.16 dimuat *Flowchart* dari *Deep Learning*, setelah program *Start*, input *Library* yang digunakan, setelah itu menunjukkan path dimana letak *Dataset* itu pada Google Drive. Dengan *ImageDataGenerator* dari *Library Keras*, digunakan untuk membuat *Data* augmentasi, bertujuan untuk menambah variasi serta jumlah dari *Dataset* dikarenakan *Dataset* yang dimiliki masih relatif sedikit sehingga agar tidak terjadi *Underfitting* (kondisi dimana model kurang terpelajari dengan baik karena kurangnya *Data Training*). Parameter pada *Data* augmentasi yang digunakan adalah *rescale*, *rotation range*, *zoom range*, *shear range*, *width shift range*, *height shift range*, *horizontal flip* dan *vertical flip*. Setelah itu buat model arsitektur *Convolutional Neural Network* (CNN), untuk menentukan berapa *Layer* fitur ekstraksi dan pada *Layer Fully Connected* ditentukan berapa *dense* dan fungsi aktivasi apa yang digunakan untuk *classification*. Setelah itu buat model *compile*, dalam model *compile* terdapat *optimizer* yang digunakan agar hasil *Training* bisa lebih bagus. Selanjutnya dilakukan proses *Training* dan *validasi*, disinilah *Dataset* kita dilatih oleh *Deep Learning* untuk dapat diprediksi *Class* nya. Lalu untuk melihat grafik akurasi dan validasi dilakukan *plotting* dari hasil model *Training*. Agar tidak perlu melakukan *Training* berulang kali maka model disimpan dengan format *h5*, dimana *h5* ini adalah hasil *model Training*.

### 3.2.3 Data Testing

Proses *Data Testing* terlihat pada Gambar 3.17. *Data Testing* tidak termasuk bagian dari *Data Training* atau *Data Validation*. Setelah Program *Start*, *Input Library* yang digunakan serta *Load model Training* yang telah disimpan format *h5*. *Load Image test* yang berada pada *folder* dalam Google Drive, setelah itu dilakukan *reshape Image* menjadi 4 array (1,100,100,1), dikarenakan untuk *function predict* mengharapkan *Image* dalam bentuk 4 array dan dengan *size image* yang sesuai pada *softmax*. Ketika *Image* sudah selesai di-*reshape*, proses selanjutnya adalah prediksi *Image* dilanjutkan dengan *Output* prediksi dari *function predict* agar mengetahui apakah *Class* yang diprediksi itu benar/akurat, dengan ini program berhenti/selesai.



Gambar 3.17 Flowchart Data Testing

### 3.4 Arsitektur *Convolutional Neural Network*

Arsitektur pada *Feature Extracting* menggunakan 5 *Layer* seperti pada Gambar 3.18, pemilihan berapa banyak *Layer* yang digunakan adalah hasil dari yang terbaik dari perbandingan yang dilakukan. *Layer* konvolusi seluruhnya menggunakan *kernel filter* berukuran 3x3, dengan pergeseran (*strides*) bernilai 1, fungsi aktivasi ReLU digunakan untuk mernormalisasi nilai negatif dari *output* konvolusi. *Padding* disini digunakan untuk menambah dimensi pada citra dengan cara menambah nilai 0 di lapisan luar dimensi citra, tujuannya untuk memanipulasi agar fitur ekstraksi pada citra tidak kehilangan banyak fitur nya. Lalu pada bagian *Fully Connected Layer* disini menggunakan 3 *Layer Dense*, *Layer* pertama sebanyak 512, *Layer* kedua menggunakan 256 dan yang terakhir yaitu 6 (jumlah *class*). lalu menggunakan fungsi aktivasi *softmax* dikarenakan *class* nya bersifat *multiclass*.

Model *Summary* dapat dilihat pada Gambar 3.19. Total parameter yang di-*Training* pada berjumlah 6.825.830, nilai yang nantinya akan dilakukan *Training*.

```

#buat arsitektur CNN
model = Sequential()
#Ekstraksi Fitur
model.add(Conv2D(32, kernel_size=3, strides=1, activation='relu', padding='same', input_shape=(100,100,1)))
model.add(MaxPooling2D(pool_size=(3,3),strides=2))

model.add(Conv2D(32, kernel_size=3, strides=1, activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(3,3),strides=2))

model.add(Conv2D(64, kernel_size=3, strides=1, activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(2,2),strides=2))

model.add(Conv2D(64, kernel_size=3, strides=1, activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(2,2),strides=1))

model.add(Conv2D(128, kernel_size=3, strides=1, activation='relu',padding='same'))
model.add(MaxPooling2D(pool_size=(2,2),strides=1))

#Flatten
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(classes, activation='softmax'))

```

Gambar 3.18 Arsitektur CNN yang digunakan

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 100, 100, 32)	320
max_pooling2d (MaxPooling2D)	(None, 49, 49, 32)	0
conv2d_1 (Conv2D)	(None, 49, 49, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(None, 11, 11, 64)	0
conv2d_4 (Conv2D)	(None, 11, 11, 128)	73856
max_pooling2d_4 (MaxPooling2D)	(None, 10, 10, 128)	0
flatten (Flatten)	(None, 12800)	0
dropout (Dropout)	(None, 12800)	0
dense (Dense)	(None, 512)	6554112
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 6)	1542
Total params: 6,825,830		
Trainable params: 6,825,830		
Non-trainable params: 0		

Gambar 3.19 Summary model arsitektur CNN

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil dan Grafik Proses *Training*

*Training* pada *Dataset* dilakukan dengan 250 epochs dimana setiap epochs memiliki ukuran batch sebanyak 32 dan memiliki iterasi sebanyak 17. Proses *Training* menggunakan optimizer “Adam” dengan *Learning Rate* bernilai 0.001. Hasil dari *Training* dapat terlihat pada Tabel 4.2 yang diambil dari 10 epochs terakhir, menunjukkan akurasi pada epochs 250 sebesar 91% dan akurasi validasi sebesar 90%. Model *Training* disimpan dalam bentuk file .h5 sehingga tidak perlu melakukan *Training* lagi dikarenakan model sudah disimpan dalam bentuk h5 dan disimpan di Google Drive.

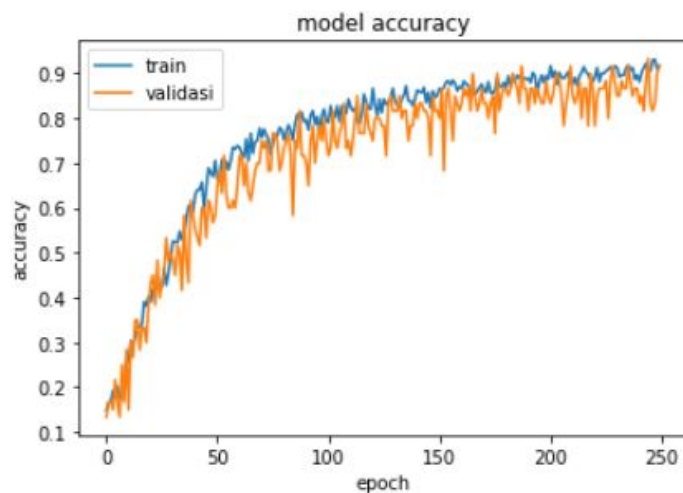
Tabel 4.1 Akurasi Hasil *Training*

Epochs	Akurasi <i>Training</i> (%)	Akurasi <i>Validation</i> (%)	<i>Training Loss</i> (%)	<i>Validation Loss</i> (%)
241	89	83	28	39
242	89	85	31	36
243	92	82	22	58
244	93	88	26	38
245	91	93	23	22
246	92	83	25	48
247	93	82	25	51
248	94	83	17	37
249	92	90	22	29
250	91	90	25	23

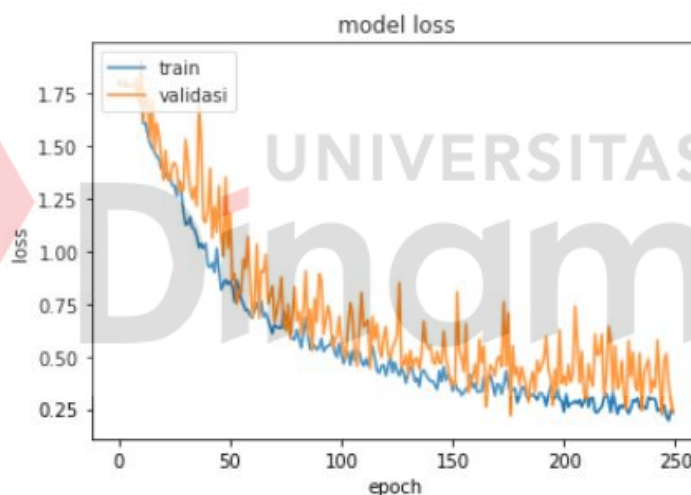
Hasil *Training* dapat dipengaruhi dari beberapa aspek, yang pertama adalah *Layer* pada arsitektur *Convolutional Neural Network*, dimana jika total parameter akhir nya bernilai belasan juta atau bahkan ratusan juta program *Training* nya pasti lebih lama, tetapi bisa jadi karena parameter nya banyak yang di *Training* maka hasil *Training* nya lebih bagus. Juga bisa dari fitur ekstraksi pada citra input, pada penelitian ini hasil *Training* yang paling bagus yaitu dengan menggunakan 5 *Layer* seperti pada Gambar 3.18. Yang kedua adalah percobaan pada *Learning rate* dari *optimizer*, nilai pada setiap kasus berbeda-beda, dalam penelitian ini *Learning Rate*



bernilai 0.001 adalah yang paling bagus. Yang ketiga adalah dari *Dataset* itu sendiri. Kuantitas dan kualitas *Dataset* dapat mempengaruhi hasil *Training*.



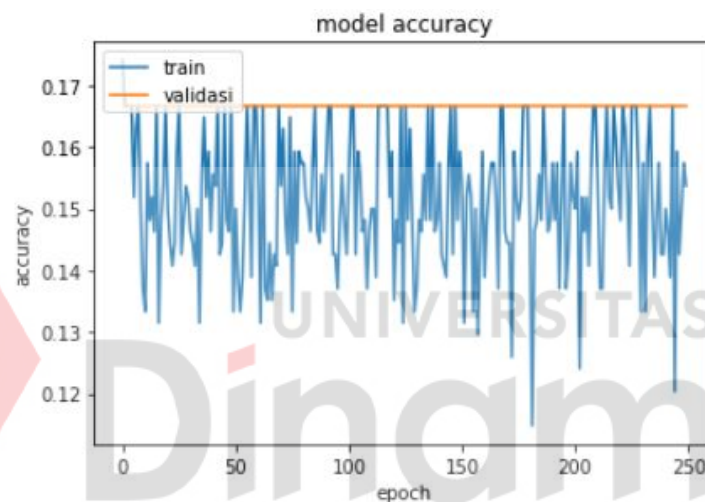
Gambar 4.1 Grafik akurasi dengan 5 Layer



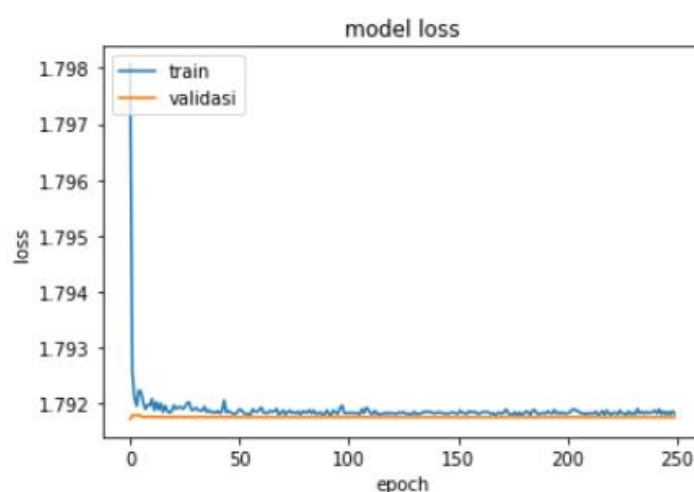
Gambar 4.2 Grafik Loss dengan 5 Layer

Hasil *Training* juga di plot dalam bentuk grafik yang terlihat pada Gambar 4.1 dan Gambar 4.2. Dalam grafik akurasi dan *loss* terlihat fluktuatif, namun tidak terjadi *Overfitting* (kondisi yang biasa memiliki *loss* rendah dan akurasi rendah) maupun *Underfitting* (kondisi yang biasa memiliki *loss* yang tinggi dan akurasi rendah), ini dikarenakan penggunaan metode adalah dengan CNN yang dasar, tanpa menggunakan *Transfer Learning* sehingga hasil grafik masih kurang *Smooth*. Hasil *Training* cenderung naik sedangkan untuk *Loss* cenderung turun.

Sedangkan percobaan dengan 6 *Layer* mengalami perubahan yang sangat signifikan pada hasil grafik. Hasil grafik yang didapatkan tidak bagus, seperti pada Gambar 4.3 dan Gambar 4.4. Pada Grafik akurasi mengalami fluktuasi yang tinggi terhadap hasil *Training* dan tidak mengalami peningkatan akurasi pada hasil *Training* maupun *Validation*, sedangkan pada grafik *Loss* tidak mengalami kecenderungan turun pada nilai *Loss* nya. Dapat di analisa bahwa penambahan menjadi 6 *Layer* membuat program tidak dapat mempelajari fitur pada citra dengan baik, dengan kata lain arsitektur yang digunakan tidak bagus untuk mengekstrak fitur pada citra.



Gambar 4. 3 Grafik akurasi dengan 6 *Layer*



Gambar 4. 4 Grafik Loss dengan 6 *Layer*



Tabel 4. 2 Tabel performa hasil model 5 *Layer*

	Precision	Recall	F1-Score	Support
“Cinta”	1.00	1.00	1.00	6
“Dia”	0.80	1.00	0.89	4
“Kamu”	1.00	0.86	0.92	7
“Maaf”	0.50	1.00	0.67	3
“Saya”	1.00	0.57	0.73	7
“Sedih”	1.00	1.00	1.00	5
Accuracy			0.88	32
Macro Avg	0.88	0.90	0.87	32
Weighted Avg	0.93	0.88	0.88	32

Pada Tabel 4.2 adalah hasil performa dari model. Dari hasil tabel nilai sensitivitas (*Recall*) yang didapatkan adalah 90%, nilai ini didapatkan dari rata-rata seluruh *class*. Sedangkan untuk spesifitas (*Precision*) mendapatkan hasil 88%, sama halnya dengan sensitivitas, nilai spesifitas didapatkan dari rata-rata setiap *class*. Nilai *score* mendapatkan rata-rata 87%, nilai *score* itu sendiri merupakan acuan pada hasil model untuk prediksi.

#### 4.2 Perbandingan Hasil *Training* dan *Validation* Pada Arsitektur CNN

Untuk mengetahui arsitektur CNN yang terbaik untuk digunakan pada penelitian ini, maka dilakukan beberapa proses *Training* dengan 4 arsitektur yang berbeda, yang di ubah pada fitur ekstraksinya (Konvolusi) yaitu 4 *Layer*, 5 *Layer*, 6 *Layer* dan 7 *Layer*, hasil nya dapat dilihat pada Tabel 4.3 dibawah ini.

Tabel 4. 3 Tabel Perbandingan Hasil *Training*

<i>Layer</i> pada <i>Feature Learning</i>	Epochs	Akurasi <i>Training</i> (%)	Akurasi <i>Validation</i> (%)	<i>Training</i> <i>Loss</i> (%)	<i>Validation</i> <i>Loss</i> (%)
4 <i>Layer</i>	241	87	83	35	59
	242	89	83	33	47
	243	90	83	28	39
	244	86	83	42	40
	245	89	80	27	51
	246	88	80	36	44
	247	88	88	30	33
	248	87	83	38	42
	249	86	86	31	29
	250	86	83	42	41
5 <i>Layer</i>	241	89	83	28	39
	242	89	85	31	36
	243	92	82	22	58
	244	93	88	26	38
	245	91	93	23	22

<i>Layer pada Feature Learning</i>	Epochs	Akurasi <i>Training (%)</i>	Akurasi <i>Validation (%)</i>	<i>Training Loss (%)</i>	<i>Validation Loss (%)</i>
6 Layer	246	92	83	25	48
	247	93	82	25	51
	248	94	83	17	37
	249	92	90	22	29
	250	91	90	25	23
	241	16	17	179	179
	242	15	17	179	179
	243	16	17	179	179
	244	19	17	179	179
	245	12	17	179	179
7 Layer	246	17	17	179	179
	247	16	17	179	179
	248	15	17	179	179
	249	19	17	179	179
	250	17	17	179	179
	241	15	17	179	179
	242	13	17	179	179
	243	18	17	179	179
	244	16	17	179	179
	245	14	17	179	179
	246	16	17	179	179
	247	18	17	179	179
	248	14	17	179	179
	249	18	17	179	179
	250	17	17	179	179

Hasil dari perbandingan pada keempat *Layer* yang berbeda pada arsitektur mendapatkan hasil terbaik pada 5 *Layer* dengan hasil pada epoch terakhir akurasi *Training* sebesar 91% dan akurasi *Validation* sebesar 90% sedangkan pada hasil *Loss* mendapatkan 25% untuk *Training* dan 23% untuk *Validation*. Hasil dari 4 *Layer* cukup bagus namun pada hasil akhir, nilai akurasi *Training* yang didapatkan pada epoch terakhir 86% dan *Validation* sebesar 83%, hasil *Loss* pun masih cukup besar yaitu dengan *Loss* pada *Training* 42% dan *Validation* 41%. Sedangkan perbandingan dengan 6 *Layer* dan 7 *Layer*, hasil yang didapatkan sangatlah kurang atau bisa dikatakan hasil model sangat buruk. Akurasi < 20% dan *Loss* nya masih >100%.

Kesimpulan analisa yang didapatkan adalah arsitektur yang digunakan bukan berarti semakin banyak *Layer* yang dipakai semakin bagus hasilnya. Sehingga pada penelitian ini arsitektur terbaik yang digunakan pada *Feature Learning* adalah dengan 5 *Layer*, penambahan lebih dari 5 *Layer* mengakibatkan hasil model *Training* yang buruk.

### 4.3 Image Prediction

Model yang disimpan dalam bentuk h5 menjadi model yang di *testing*. *Data testing* yang digunakan tidak diambil dari *Data* yang digunakan untuk *Training*. Total *Data testing* yang digunakan sebanyak 60 citra, subjek memiliki 2 citra pada setiap simbol SIBI (Sistem Isyarat Bahasa Indonesia). 2 citra tersebut antara lain, Citra 1 memakai pakaian bebas, bisa dengan warna nya yang bervariasi maupun pakaian yang memiliki logo atau gambar, ini dilakukan sebagai variasi, untuk mengetahui apakah akan mempengaruhi pada prediksi. sedangkan Citra 2 menggunakan pakaian berwarna putih, sebagai variasi pertama yang memiliki standar dengan *Data Training* namun berbeda dengan *Data Training*. Citra juga dilakukan *Image Pre-processing*, agar memudahkan untuk prediksi dengan akurat, selain itu perlu dilakukan *resize* menjadi 100x100, sebelum dilakukan prediksi citra di-*reshape* menjadi 1x100x100x1 karena pada fungsi aktivasi *softmax*, mengharapkan citra dalam dimensi 100x100 dan memiliki 4 array.

Hasil dari prediksi dapat dilihat pada Tabel 4.4 simbol “Cinta” mendapatkan akurasi sebesar 80%, simbol “Dia” dan “Saya” mendapatkan akurasi 70%, simbol “Kamu” mendapatkan akurasi 90%, simbol “Sedih” mendapatkan akurasi sebesar 100% sedangkan pada simbol “Maaf” mendapatkan akurasi 0%. Analisa pada *miss prediction* adalah dimana pada bahasa isyarat memiliki kemiripan, contoh seperti pada simbol “Maaf” memiliki kemiripan terhadap simbol “sedih” sehingga simbol “Maaf” tidak dapat terprediksi dengan benar, sama dengan halnya simbol “Cinta” yang memiliki 2 *miss prediction* dan “Saya” yang memiliki 3 *miss prediction*. Hal ini juga dapat dipengaruhi dari hasil *Image Pre-Processing* dimana hasil *output* yang dimasukkan pada proses *Training* dan *Validation* memiliki fitur yang hampir sama untuk beberapa simbol bahasa isyarat tertentu. Sehingga pada model hasil *Training* tidak cukup bagus pada saat dilakukan *Image Prediction*.

Tabel 4. 4 Tabel Hasil Prediksi

No	Subjek	Simbol SIBI (Sistem Isyarat Bahasa Indonesia)					
		Cinta	Dia	Kamu	Maaf	Saya	Sedih
1	Subjek 1	Citra 1	Citra 1	Citra 1	Citra 1	Citra 1	Citra 1
		Prediksi: Cinta	Prediksi: Dia	Prediksi: Kamu	Prediksi: Sedih	Prediksi: Saya	Prediksi: Sedih
		Citra 2	Citra 2	Citra 2	Citra 2	Citra 2	Citra 2
2	Subjek 2	Prediksi: Cinta	Prediksi: Dia	Prediksi: Kamu	Prediksi: Sedih	Prediksi: Saya	Prediksi: Sedih
		Citra 1	Citra 1	Citra 1	Citra 1	Citra 1	Citra 1
		Prediksi: Sedih	Prediksi: Saya	Prediksi: Sedih	Prediksi: Sedih	Prediksi: Saya	Prediksi: Sedih
3	Subjek 3	Citra 2	Citra 2	Citra 2	Citra 2	Citra 2	Citra 2
		Prediksi: Cinta	Prediksi: Dia	Prediksi: Kamu	Prediksi: Sedih	Prediksi: Sedih	Prediksi: Sedih
		Citra 1	Citra 1	Citra 1	Citra 1	Citra 1	Citra 1
4	Subjek 4	Prediksi: Cinta	Prediksi: Kamu	Prediksi: Kamu	Prediksi: Cinta	Prediksi: Sedih	Prediksi: Sedih
		Citra 2	Citra 2	Citra 2	Citra 2	Citra 2	Citra 2
		Prediksi: Sedih	Prediksi: Dia	Prediksi: Kamu	Prediksi: Sedih	Prediksi: Saya	Prediksi: Sedih
5	Subjek 5	Citra 1	Citra 1	Citra 1	Citra 1	Citra 1	Citra 1
		Prediksi: Cinta	Prediksi: Dia	Prediksi: Kamu	Prediksi: Sedih	Prediksi: Saya	Prediksi: Sedih
		Citra 2	Citra 2	Citra 2	Citra 2	Citra 2	Citra 2
Presentase		Prediksi: Cinta	Prediksi: Dia	Prediksi: Kamu	Prediksi: Sedih	Prediksi: Saya	Prediksi: Sedih
		Benar: 8	Benar: 7	Benar: 9	Benar: 0	Benar: 7	Benar: 10
		Salah: 2	Salah: 3	Salah: 1	Salah: 10	Salah: 3	Salah: 0
		Presentase: 80%	Presentase: 70%	Presentase: 90%	Presentase: 0%	Presentase: 70%	Presentase: 100%

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan didapatkan beberapa kesimpulan yaitu:

1. Hasil dari *Training Dataset* menggunakan *Convolutional Neural Network* menghasilkan nilai akurasi *Training* sebesar 91% sedangkan untuk validasi menghasilkan nilai akurasi sebesar 90%.
2. *Dataset* untuk *Training*, *Validation* dan *Testing* perlu dilakukan proses *Image Pre-processing*, hal ini dilakukan agar proses *Training* dan *Testing* fokus pada bentuk isyarat tangan bukan pada orangnya.
3. Dari hasil *Data Testing* sebanyak 60 citra terbagi menjadi 6 simbol, masing-masing simbol memiliki 10 *Data Testing* dengan 5 subjek, menghasilkan prediksi dan presentase sebagai berikut:
  - a. Simbol “Cinta” mendapatkan akurasi sebesar 80%
  - b. Untuk simbol “Dia” mempunyai akurasi sebesar 70%
  - c. Sedangkan simbol “Kamu” memiliki akurasi sebesar 90%
  - d. Selanjutnya Simbol “Maaf” akurasi yang didapatkan 0%
  - e. Adapun Simbol “Saya” menghasilkan akurasi sebesar 70%
  - f. Yang terakhir Simbol “Sedih” mencapai tingkat akurasi sebesar 100%

#### 5.2 Saran

Adapun saran untuk pengembangan pada penelitian ini agar lebih baik, terdapat beberapa saran yaitu:

1. Variasi *Dataset* dapat ditambah, adapun penambahan *Dataset* terdapat beberapa aspek, yang pertama dari aspek kuantitas bahasa isyarat, bisa ditambahkan simbol bahasa isyarat yang lain. Lalu yang kedua adalah pengambilan *Dataset* bisa diambil dengan variasi *background* (tanpa *green screen*).
2. Meningkatkan kualitas pada *Dataset*, bisa dengan melakukan proses seperti *Labelling*, *Annotation*, *Caption* atau *Ground Truth* yang dimana *Dataset* tidak

perlu dilakukan *Image Pre-processing*, sehingga yang akan di *Training* oleh *Deep Learning* nya adalah citra asli.

3. Menggunakan proses *Deep Learning* selain *Convolutional Neural Network* seperti YOLO (*You Only Look Once*) atau SSD (*Single Shot Detector*)
4. Untuk mengurangi grafik yang fluktuatif bisa menggunakan *Convolutional Neural Network* dengan tambahan metode seperti *Transfer Learning*, bisa menggunakan VGG16, ResNet50, Xception dan lain-lain, agar hasil grafik bisa lebih *Smooth*.



UNIVERSITAS  
Dinamika

## DAFTAR PUSTAKA

- Ahmadian, H., 2020. *Apa itu Google Colaboratory*. [Online]  
Available at: <https://hendri83.wordpress.com/2020/05/23/apa-itu-google-colaboratory/>
- Asriani, F. & Susilawati, H., 2010. Pengenalan Isyarat Tangan Statis Pada Sistem Isyarat Bahasa Indonesia Berbasis Jaringan Syaraf Tiruan Perambatan Balik. *Makara, Teknologi*, 14(2), pp. 150-154.
- Darmatasia, 2020. Pengenalan Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Gradient Convolutional Neural Network. *Jurnal Informatika Sains dan Teknologi*.
- INIXINDO, 2021. *Apa Itu Machine Learning dan Perbedaannya Dengan Deep Learning*. [Online]  
Available at: <https://inixindo.co.id/index.php/it-forum/79-pemrogramman/680-apa-itu-machine-learning-dan-perbedaannya-dengan-deep-learning>  
[Accessed 4 June 2021].
- Jelita, I. N., 2018. *MEDIA INDONESIA*. [Online]  
Available at: <https://m.mediaindonesia.com/politik-dan-hukum/204499/tercatat-12-juta-penyandang-disabilitas-masuk-dpt-pemilu-2019>
- Katiya, S. K. & Arun, P. V., 2014. Comparative Analysis of Common Edge Detection Techniques of Context of Object Extraction. *IEEE TGRS*, 50(11b), pp. 68-79.
- Maharsi, J., 2018. *Mengenal Bahasa Isyarat*. [Online]  
Available at: <https://komunita.id/2018/11/30/mengenal-bahasa-isyarat/>
- Mursita, R. A., 2015. Respon Tunarungu Terhadap Penggunaan Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO).
- Nugroho, P. A., Fenriana, I. & Arijanto, R., 2020. Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Jurnal Algor*.

Sena, S., 2017. *Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN)*. [Online]

Available at: <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>

[Accessed 1 April 2021].

Suartika, I. W., Wijaya, A. Y. & Soelaiman, R., 2016. Klasifikasi Citra Menggunakan Convolutional Neural Network. *Jurnal Teknik ITS*, V(1).



UNIVERSITAS  
Dinamika